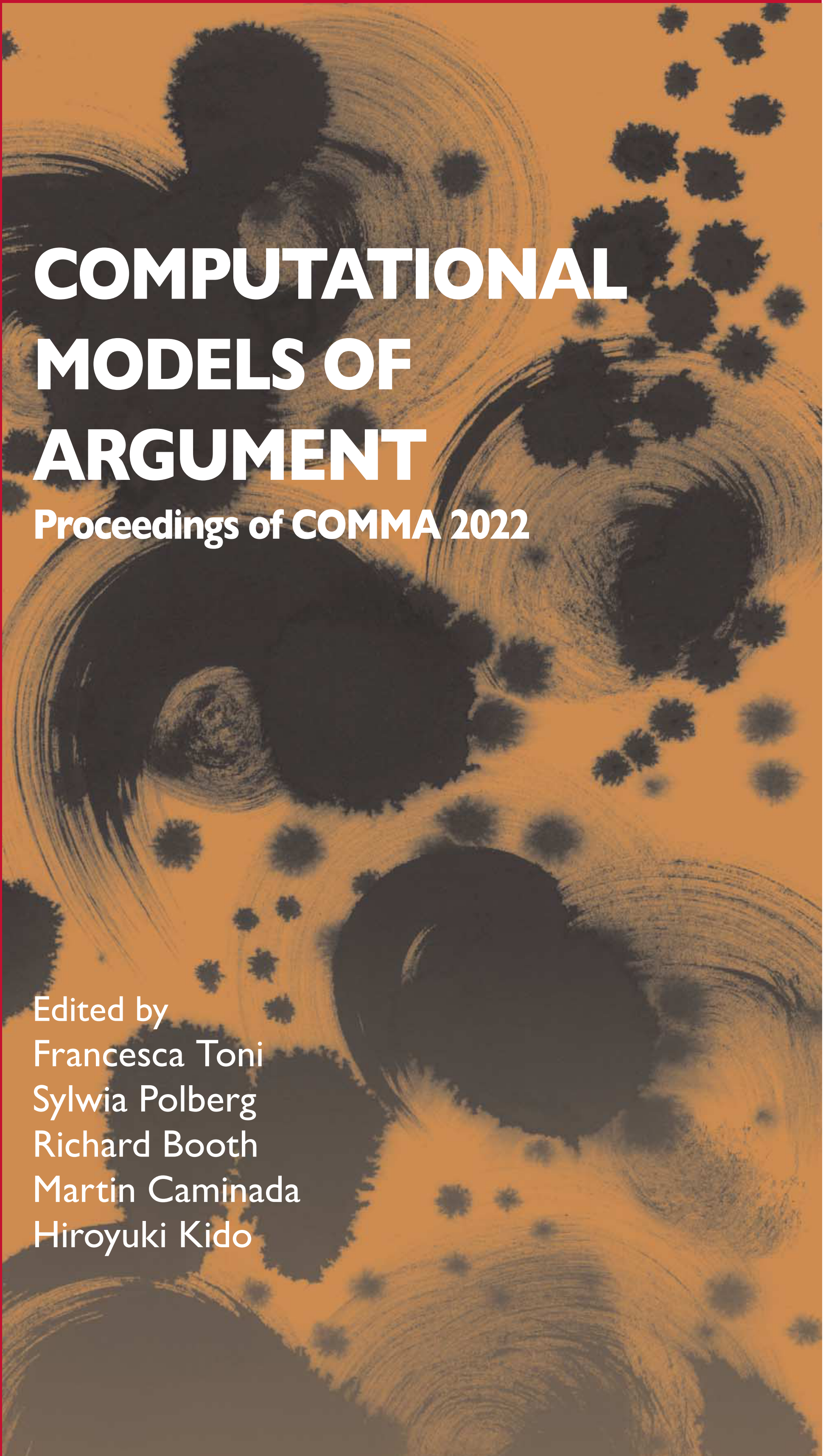
Frontiers
in
Artificial
Intelligence
and
Applications



COMPUTATIONAL MODELS OF ARGUMENT

Proceedings of COMMA 2022

Edited by
Francesca Toni
Sylwia Polberg
Richard Booth
Martin Caminada
Hiroyuki Kido

IOS Press

COMPUTATIONAL MODELS OF ARGUMENT

Argumentation has traditionally been studied across a number of fields, notably philosophy, cognitive science, linguistics and jurisprudence. The study of computational models of argumentation is a more recent endeavor, bringing together researchers from traditional fields and computer science and engineering within a rich, interdisciplinary matrix. Computational models of argumentation have been identified and used since the 1980s, and more recently an important role for argumentation in leading to principled decisions has emerged in several settings.

This book presents the proceedings of COMMA 2022, the 9th International Conference on Computational Models of Argument, held in Cardiff, Wales, United Kingdom, during 14 - 16 September 2022. The book contains 27 regular papers and 16 demo papers from a total of 75 submissions, as well as 3 invited talks from Prof. Paul Dunne (University of Liverpool), Prof. Iryna Gurevych (TU Darmstadt), and Prof. Antonis Kakas (University of Cyprus), which reflect the diverse nature of the field. Papers are a mix of theoretical and practical contributions; theoretical contributions include new formal models, the study of formal or computational properties of models, design for implemented systems and experimental research; practical papers include applications to law, machine learning and explainability. Abstract and structured accounts of argumentation are covered, as are relations between different accounts. Many papers focus on the evaluation of arguments or their conclusions given a body of arguments, with a continuation of a recent trend to study gradual or probabilistic notions of evaluation.

The book offers an overview of recent and current research and will be of interest to all those working with computational models of argumentation.

9 781643 683065

ISBN 978-1-64368-306-5 (print)
ISBN 978-1-64368-307-2 (online)
ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

COMPUTATIONAL MODELS OF ARGUMENT

Frontiers in Artificial Intelligence and Applications

The book series Frontiers in Artificial Intelligence and Applications (FAIA) covers all aspects of theoretical and applied Artificial Intelligence research in the form of monographs, selected doctoral dissertations, handbooks and proceedings volumes. The FAIA series contains several sub-series, including 'Information Modelling and Knowledge Bases' and 'Knowledge-Based Intelligent Engineering Systems'. It also includes the biennial European Conference on Artificial Intelligence (ECAI) proceedings volumes, and other EurAI (European Association for Artificial Intelligence, formerly ECCAI) sponsored publications. The series has become a highly visible platform for the publication and dissemination of original research in this field. Volumes are selected for inclusion by an international editorial board of well-known scholars in the field of AI. All contributions to the volumes in the series have been peer reviewed.
The FAIA series is indexed in ACM Digital Library; DBLP; EI Compendex; Google Scholar; Scopus; Web of Science: Conference Proceedings Citation Index – Science (CPCI-S) and Book Citation Index – Science (BKCI-S); Zentralblatt MATH.

Volume 353

Recently published in this series

Vol. 352. A.J. Tallón-Ballesteros (Ed.), Modern Management based on Big Data III – Proceedings of MMBD 2022
Vol. 351. A. Passerini and T. Schiex (Eds.), PAIS 2022 – 11th Conference on Prestigious Applications of Artificial Intelligence, 25 July 2022, Vienna, Austria (co-located with IJCAI-ECAI 2022)
Vol. 350. P. Morettin, Learning and Reasoning in Hybrid Structured Spaces
Vol. 349. A. De Filippo, Hybrid Offline/Online Methods for Optimization Under Uncertainty
Vol. 348. G. Cima, Abstraction in Ontology-based Data Management
Vol. 347. L.C. Jain, V.E. Balas, Q. Wu and F. Shi (Eds.), Design Studies and Intelligence Engineering
Vol. 346. E. Schweighofer (Ed.), Legal Knowledge and Information Systems – JURIX 2021: The Thirty-fourth Annual Conference, Vilnius, Lithuania, 8–10 December 2021
Vol. 345. A.J. Tallón-Ballesteros (Ed.), Proceedings of CECNet 2021 – The 11th International Conference on Electronics, Communications and Networks (CECNet), November 18–21, 2021

ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

Computational Models of Argument

Proceedings of COMMA 2022

Edited by

Francesca Toni

Department of Computing, Imperial College London, United Kingdom

Sylwia Polberg

School of Computer Science and Informatics, Cardiff University,
United Kingdom

Richard Booth

School of Computer Science and Informatics, Cardiff University,
United Kingdom

Martin Caminada

School of Computer Science and Informatics, Cardiff University,
United Kingdom

and

Hiroyuki Kido

School of Computer Science and Informatics, Cardiff University,
United Kingdom

IOS Press

Amsterdam • Berlin • Washington, DC

ISBN 978-1-64368-306-5 (print)
ISBN 978-1-64368-307-2 (online)
Library of Congress Control Number: 2022944149
doi: 10.3233/FAIA353

Publisher
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

For book sales in the USA and Canada:
IOS Press, Inc.
6751 Tepper Drive
Clifton, VA 20124
USA
Tel.: +1 703 830 6300
Fax: +1 703 830 2300
sales@iospress.com

LEGAL NOTICE

The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Preface

Argumentation has been traditionally studied across a number of fields, notably philosophy, cognitive science, linguistics and jurisprudence. The study of *computational models* of argumentation is a more recent endeavour, bringing together researchers across these traditional fields as well as computer scientists and engineers, amongst others, within a rich, interdisciplinary, exciting discipline with much to offer. Computational models of argumentation have emerged since the eighties. Starting with Pollock ("Defeasible reasoning", 1987), argumentation was identified as a way to understand defeasible reasoning, with the first systematic formal account of the evaluation of arguments given their internal structure and their relation with counterarguments. In AI, starting with Lin and Shoham ("Argument systems: A uniform basis for nonmonotonic reasoning", 1989), Dung ("Negations as hypotheses: An abductive foundation for logic programming", 1991), and Kakas, Kowalski, and Toni ("Abductive logic programming", 1992), argumentation was proposed as a unifying formalism for various existing forms of non-monotonic, default reasoning. This line of research led to the development of the seminal abstract argumentation frameworks by Dung ("On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games", 1995), awarded the AI Journal Classic Paper Award in 2018 in recognition of this paper's crucial role in making argumentation a mainstream research topic in AI. Furthermore, in the study of decision-making, Krause, Ambler, and Fox ("The development of a logic of argumentation", 1992), pointed to the important role for argumentation to lead to principled decisions (in general and in a medical setting). Today's computational models of argumentation share many goals with these early works, notably an awareness of the importance of formal models which lend themselves to be implemented as computer programs. These can then be integrated into "arguing" systems able to engage in argumentation-related activities with humans or with other systems. As such, computational models of argumentation require crossing bridges with a variety of disciplines, including computational linguistics, formal logic, social choice, game theory, graph theory and AI and law.

Since 2006 the biennial International Conference on Computational Models of Argument (COMMA) has provided a dedicated forum for presentation and discussion of the latest advancements in this interdisciplinary field, covering basic research, systems and innovative applications. The first COMMA was supported by the EU 6th Framework Programme project ASPIC and was hosted by the University of Liverpool in 2006. After the event, a steering committee promoting the continuation of the conference was established and, since then, the steady growth of interest in computational argumentation research worldwide has gone hand in hand with the development of the conference itself and of related activities by its underpinning community. Since the second edition, organized by IRIT in Toulouse in 2008, plenary invited talks by world-leading researchers and a software demonstration session became an integral part of the conference programme. The third edition, organized in 2010 by the University of Brescia in Desenzano del Garda, saw the addition of a best student paper award. The same year, the new journal Argument and Computation, closely related to the COMMA activities, was started.

Since the fourth edition, organized by the Vienna University of Technology in 2012, an Innovative Application Track and a section for Demonstration Abstracts were included in the proceedings. At the fifth edition, co-organized in 2014 by the Universities of Aberdeen and Dundee in Pitlochry, the main conference was preceded by the first Summer School on Argumentation: Computational and Linguistic Perspectives. The same year also saw the launch of the first International Competition on Computational Models of Argumentation (ICCMA). Since COMMA 2016, hosted by the University of Potsdam, the COMMA proceedings are Open Access. This COMMA was also the first that included additional satellite workshops in the programme. COMMA 2018 was hosted by the Institute of Philosophy and Sociology of the Polish National Academy of Sciences in Warsaw, Poland. It included an industry afternoon bringing together businesses, NGOs, academics and students interested in practical applications of argument technologies in industry. COMMA 2020 was organised in Italy for the second time, by the University of Perugia, but, due to the COVID pandemic, was run fully online. It was preceded by the 4th Summer School on Argumentation: Computational and Linguistic Perspectives (SSA 2020), and featured a demonstrations session and three satellite workshops: the International Workshop on Systems and Algorithms for Formal Argumentation (SAFA), initiated at COMMA 2016; a new Workshop on Argument Visualization, and the well-known Workshop on Computational Models of Natural Argument, established in 2001, at its 20th edition at COMMA 2020.

COMMA 2022 will be once again an in-person event, for the third time in the UK, but now in Cardiff, organised by Cardiff University. It will be preceded by the 5th Summer School on Argumentation, with a focus on “Explainability Perspective”, a topic that has grown over the last two editions of COMMA. COMMA 2022 will also be preceded by four workshops: CMNA 2022, the Workshop on Computational Models of Natural Argument (at its 21st edition at COMMA 2022); SAFA 2022, the 4th International Workshop on Systems and Algorithms for Formal Argumentation; as well as ArgXAI 2022, the 1st International Workshop on Argumentation for eXplainable AI, and ArgML 2022, the 1st International Workshop on Argumentation & Machine Learning. The latter two workshops reflect novel avenues being explored by the COMMA community, building bridges with data-centric AI.

The COMMA 2022 programme reflects the interdisciplinary nature of the field, and its contributions range from theoretical to practical. Theoretical contributions include new formal models, the study of formal or computational properties of models, design for implemented systems and experimental research. Practical papers include applications to law, machine learning and explainability. As in previous editions of COMMA, papers cover abstract and structured accounts of argumentation, as well as relations between different accounts. Many papers focus on the evaluation of arguments or their conclusions given a body of arguments, with a continuation of a recent trend to study gradual or probabilistic notions of evaluation.

COMMA 2022 also hosts a demonstration session, as in previous years, with 16 demos (one, NEXAS, described in a full paper) indicating that the field is ripe for models and methods to be integrated within a variety of applications.

The three invited talks also reflect the diverse nature of the field. Prof Paul Dunne, from the University of Liverpool, gives an overview of the study of computational complexity in argumentation; Prof Iryna Gurevych, from TU Darmstadt, discusses an important application area, namely dealing with misinformation in natural language; and

Prof Antonis Kakas, from the University of Cyprus, looks at theory-informed practical applications of argumentation.

Finally, we want to acknowledge the work of all those who have contributed in making the conference and its satellite events a success. We are grateful to IOS Press for publishing these proceedings and continuing to make them Open Access. As local and international sponsors of the conference, we would like to thank the School of Computer Science and Informatics at the Cardiff University and EurAI, the European Association of AI. We acknowledge steady support and encouragement by the COMMA steering committee, and are very grateful to the programme committee and additional reviewers, whose invaluable expertise and efforts have led to the selection, out of 75 submissions, of 26 full papers, 16 extended abstract for demos, and 1 full paper also describing a demo. The submission and reviewing process has been managed through the Easychair conference system, which we acknowledge for supporting COMMA since the first edition. Our thanks also to the COMMA 2022 workshops' organisers (in no particular order): Floriana Grasso, Nancy Green, Jodi Schneider, Simon Wells, Kristijonas yras, Timotheus Kampik, Oana Cocarascu, Antonio Rago, Isabelle Kuhlmann, Jack Mumford, Stefan Sarkadi, Sarah A. Gaggl, Jean-Guy Mailly, Matthias Thimm, Johannes P. Wallner and their programme committees and invited speakers. We also thank the COMMA invited speakers and the invited speakers at the summer school programme (Antonio Rago, Markus Ulbricht, Annemarie Borg and Federico Castagna) and the members of the Online Handbook of Argumentation for AI (OHAAI) Committee (Andreas Xydis, Jack Mumford, Stefan Sarkadi, Federico Castagna) for organising the student session during the summer school. Last but not least, we thank all the authors and participants for contributing to the success of the conference with their hard work and commitment.

Francesca Toni (Programme Chair)
Sylwia Polberg (Conference and Summer School Chair, Organizing Committee Member)
Richard Booth (Demo chair, Organizing Committee Member)
Martin Caminada (Organizing Committee Member)
Hiroyuki Kido (Organizing Committee Member)

July 2022

This page intentionally left blank

About the Conference

Program Chair

Francesca Toni, Imperial College London

Steering Committee

President: Bart Verheij, University of Groningen
Vice-President: Katie Atkinson, University of Liverpool
Secretary: Matthias Thimm, FernUniversität in Hagen
Elizabeth Black, King’s College London
Anthony Hunter, University College London
Maria Vanina Martinez, CONICET Universidad de Buenos Aires
Beishui Liao, Zhejiang University
Serena Villata, Inria, Côte d’Azur University, CNRS, I3S

Conference Chair

Sylwia Polberg, Cardiff University

Demo Chair

Richard Booth, Cardiff University

Local Organizing Committee

Sylwia Polberg, Cardiff University
Martin Caminada, Cardiff University
Richard Booth, Cardiff University
Hiroyuki Kido, Cardiff University

Programme Committee

Leila Amgoud, IRIT - CNRS
Ofer Arieli, The Academic College of Tel-Aviv
Katie Atkinson, University of Liverpool

Pietro Baroni, University of Brescia
Ringo Baumann, Leipzig University
Trevor Bench-Capon, University of Liverpool
Floris Bex, Utrecht University and Tilburg University
Stefano Bistarelli, University of Perugia
Elizabeth Black, King’s College London
Alexander Bochman, Holon Institute of Technology
Elise Bonzon, LIPADE Université Paris Cité
Richard Booth, Cardiff University
Annemarie Borg, Utrecht University
Gerhard Brewka, Leipzig University
Katarzyna Budzynska, Warsaw University of Technology
Elena Cabrio, Inria, Côte d’Azur University, CNRS, I3S
Martin Caminada, Cardiff University
Federico Cerutti, University of Brescia
Carlos Chesñevar, Universidad Nacional del Sur
Oana Cocarascu, King’s College London
Andrea Cohen, ICIC CONICET Universidad Nacional del Sur
Sylvie Coste-Marquis, CRIL, University of Artois and CNRS
Kristijonas Čyras, Ericsson Research
Yannis Dimopoulos, University of Cyprus
Sylvie Doutre, University of Toulouse Capitole IRIT
Paul Dunne, University of Liverpool
Wolfgang Dvořák, Vienna University of Technology
Xiuyi Fan, Nanyang Technological University
Sarah Alice Gaggl, TU Dresden
Alejandro Garcia, Universidad Nacional del Sur
Massimiliano Giacomin, University of Brescia
Lluis Godo, Artificial Intelligence Research Institute, IIIA - CSIC
Tom Gordon, University of Potsdam
Guido Governatori
Floriana Grasso, University of Liverpool
Jesse Heyninck, Open Universiteit
Anthony Hunter, University College London
Souhila Kaci, LIRMM CNRS University of Montpellier
Antonis Kakas, University of Cyprus
Gabriele Kern-Isberner, TU Dortmund
Hiroyuki Kido, Cardiff University
Sébastien Konieczny, CRIL, University of Artois and CNRS
Marie-Christine Lagasquie-Schiex, IRIT Université Toulouse III - Paul Sabatier
John Lawrence, University of Dundee
Piyawat Lertvittayakumjorn, Imperial College London
Beishui Liao, Zhejiang University
Quratul-Ain Mahesar, University of Huddersfield
Jean-Guy Mailly, LIPADE Université Paris Cité
Maria Vanina Martinez, CONICET Universidad de Buenos Aires
Sanjay Modgil, King’s College London

Maxime Morge, Université de Lille
Juan Carlos Nieves, Umeå University
Nir Oren, University of Aberdeen
Fabio Paglieri, ISTC-CNR Rome
Simon Parsons, University of Lincoln
Guilherme Paulino-Passos, Imperial College London
Sylwia Polberg, Cardiff University
Nico Potyka, Imperial College London
Henry Prakken, Utrecht University and University of Groningen
Antonio Rago, Imperial College London
Tjitze Rienstra, Maastricht University
Odinaldo Rodrigues, King's College London
Chiaki Sakama, Wakayama University
Francesco Santini, University of Perugia
Giovanni Sartor, University of Bologna and European University Institute
Isabel Sassoon, Brunel University London
Jodi Schneider, University of Illinois at Urbana Champaign
Guillermo R. Simari, Universidad Nacional del Sur
Mark Snaith, Robert Gordon University Aberdeen
Manfred Stede, University of Potsdam
Christian Strasser, Ruhr University Bochum
Carlo Taticchi, University of Perugia
Matthias Thimm, FernUniversität in Hagen
Francesca Toni, Imperial College London
Alice Toniolo, University of St Andrews
Paolo Torroni, University of Bologna
Markus Ulbricht, Leipzig University
Leon van der Torre, University of Luxembourg
Bart Verheij, University of Groningen
Srdjan Vesic, CRIL, University of Artois and CNRS
Serena Villata, Inria, Côte d'Azur University, CNRS, I3S
Johannes P. Wallner, Graz University of Technology
Simon Wells, Edinburgh Napier University
Emil Weydert, University of Luxembourg
Stefan Woltran, Vienna University of Technology
Adam Wyner, Swansea University

Additional Reviewers

Jack Mumford, University of Liverpool
Kenneth Skiba, FernUniversität in Hagen
Isabelle Kuhlmann, FernUniversität in Hagen
Madeleine Waller, King's College London
Pierpaolo Goffredo, Inria, Côte d'Azur University
Matthias König, Vienna University of Technology
Daphne Odekerken, Utrecht University

Santiago Marro, Inria, Côte d'Azur University
Victor David, CEDRIC Conservatoire National des Arts et Metiers
Maximilian Heinrich, Leipzig University
Elfia Bezou Vrakatseli, King's College London
Sébastien Gottifredi, ICIC CONICET Universidad Nacional del Sur

Sponsors

We are thankful to EurAI for sponsoring the talk by Antonis Kakas and to Cardiff University for supporting the event.

Contents

Invited Talks

Regular Papers

Demo Papers

Invited Talks

This page intentionally left blank

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220134

Well, to Be Honest, I Wouldn't Start from Here at All

(A Personal View of Complexity in Argumentation After 20 Years)

Paul E. Dunne
Department of Computer Science, University of Liverpool, UK

Abstract. Computational complexity theory and the related area of efficient algorithms have formed significant subfields of Abstract Argumentation going back over 20 years. There have been major contributions and an increased understanding of the computational issues that influence and beset effective implementation of argument methods. My aim, in this article, is to attempt to take stock of the standing of work in complexity theory as it presently is within the field of Computational Argument, as well as offering some personal views on its future direction.

Introduction

There is an English tourist on a walking holiday in Eire who, while wandering around, becomes aware that he has no idea of what direction he ought to take. He sees a farmer and, approaching him, then says "I say, old chap, I'm in a bit of a frightful mess here. I'm trying to find The Old Manor House, and I don't have the deuce of a notion how to get there. Be most awfully grateful if you could help." The farmer looks up, saying in reply "So it be the Old Manor House ye're wantin' then? Sure an' that's nott a problem at all. What ye want to be doin' is this: ye go across thon field, ye turn left at the hayrick, left mind ye now, nott right, ye carry on till ye reach the cowshed and then . . . No, no no, that's not it, that's not it at all. Here what ye need to do is go down to the brook, follow it about a hunner yards an' ye'll come to a bridge, ye cross that an' straight on another hunner yards or so till ye see the windmill, ye go to the right (right mind ye right nott left) and, and . . ." The farmer's voice trails off and he hesitates a long while before continuing. "Sure and it's a divvil of a problem this, divvil of a problem. Ye know what I'm thinking? Ye know what it is I'm thinking? I'm thinking if I wanted to get to The Old Manor House, *well, to be honest, I wouldn't start from here at all.*".

All of which is to make the point that sometimes it feels as if Computational Complexity Theory (or, more accurately, the practice of Complexity Theory within Computational Argument) is in a similar position to that of the English tourist: vaguely aware of an end it wishes to achieve but unsure of how best to get there in the most direct way, and, in consequence, finding its path diverted along the scenic detours offered by brooks and windmills. My purpose, in this article, is to consider the extent to which this viewpoint is justified. In doing so, after the short recap of Section 1, I consider, in Section 2, the origins of Computational Argument in a form that gave a model suited to algorithmic and complexity study: this, of course, is the watershed approach of Dung [27]. I will look at what grew from Dung's work over the ten years between its appearance and the inaugural COMMA in 2006. This forms the basis of Section 3. What would prove to be a discovery of crucial importance to complexity analysis of Dung's model appeared in a different context: what is now dubbed *The Standard Translation* of Dimopoulos and Torres [16]. The basis of this and its importance are considered in Section 4. As complexity and algorithmic study of Dung's model proceeded the concept of *Canonical Decision Problem* emerged as a means of focusing issues with new models. In Section 5 I revisit this canon examining what its impact on Computational Argument has been in Section 6. In Section 7 the central theme is that of areas of neglect: which and to what extent such lacunae matter. Conclusions are offered in Section 8.

This may seem to be, as is probably apparent from its opening, a rather unusual paper: there are no intricate technical analyses of existing models, no new models being presented and justified. Its primary stance is that of a personal reflection on the status of a specialist field in which I have spent twenty years researching. As such its opinions about significant landmarks are highly subjective and should not be regarded as definitive factual assertions. It is also rather exceptional in containing no occurence of the word "*divers*", as in "*divers models*".

1. Prelude: Algorithms and Complexity

As a topic of research the study of algorithms has a history going back over 2,000 years: Euclid presents methods for geometric constructions in *The Elements*; Eratosthenes a technique for identifying Prime Numbers; Newton and his successors would offer approaches to finding so-called zeros of functions, generalizing the discoveries of Cardano and Ferrari respecting closed form solutions for roots of small degree polynomials [12]. Euclid, however, not only presents solutions but poses problems: one of these, "squaring the circle" would remain unresolved until the close of the 19th Century (Lindemann [37,38]). I mention this background to stress not only the historical depth of algorithm study but also to highlight some consequences already beginning to be apparent as a consequence of Lindemann's discoveries. Lindemann demonstrated that an algorithmic problem could not be solved *within the system allowed for its solution* (Ruler-and-compass). Fifty years later the discoveries of Gödel and Turing [31,44], would show that the underlying system was only part of the cause: the phenomenon of not being solvable by algorithmic means was pervasive and exhibited by all programming systems. One result is to split the world of function computation into two parts: some computational process (i.e. algorithm) exists and no such process is possible. Study of the latter, in the guise of Recursive Function Theory, would give rise to many ideas (e.g. degrees of computability, The Arithmetic Hierarchy) already beginning to lose any tangible link to computational concerns as faced in reality. In as much as Computational Argument is linked to proof theory within classical logic such non-computability issues are present in argumentation.

The concern of non-computability while present is, however, not where the focus of algorithm study has been regarding Computational Argument. The objects we wish to identify *can* be discovered: the question arising from the investigations of algorithms and Computational Complexity theory is whether it is possible to do so "efficiently". The notion of what is meant by efficiently is that the resources used by the algorithm, be it some measure of run time or of memory demands, increase proportionately to some slowly growing function of the input size. Run time uses polynomial growth rate (formally n^k for some constant k on inputs with n items). Algorithms address specific computational problems. Computational Complexity asks questions concerning what are the best algorithms possible for a given problem. Here we find a common misusage: algorithms have resource demands, they do *not* have complexity; problems have an associated complexity (formally a problem belongs to a complexity *class* membership of which is witnessed by an algorithm whose requirements are captured through that class).

There are problems for which efficient algorithms have been discovered and there are those for which no efficient algorithm is possible. Since its formal development in the mid 1960s with Hartmanis and Stearns [32][1] Computational Complexity theory has faced an open issue of some considerable magnitude: we know for any resource demand that there are functions whose computation must use at least this bound, however, there are no typically encountered problems for which such bounds have been *formally proved*. This leaves many of the problems often met in practical settings in an uncertain state: it is believed, albeit on the basis of purely circumstantial indicators, that usable algorithmic solutions cannot be found but there is no definite proof of this. These circumstantial indicators are based on the argument that an efficient solution for one yields efficient methods for all. By far the most common measure of intractability arises from the technical theory of NP–completeness, see Garey and Johnson [30]: a defence of the claim "The problem P is intractable" being made by demonstrating that P is NP–complete. Such demonstrations involve taking a known NP–complete problem, Q say, and phrasing algorithms for it in terms of algorithms for P: if the rephrasing or *reduction* is efficient (written $Q \leq_p P$) then any fast algorithm for P perforce yields a fast algorithm for Q. Computational Complexity theory offers NP–completeness as one basis of intractability, there are, however, many others, cf. Johnson [33].

In summary a formal proof that some problem is NP–complete is viewed as sufficient to take that problem out of the realm of those for which efficient algorithmic methods are possible. This, of course, should not lead to giving up on solution methods or becoming resigned to inordinate performance demands. A demonstration of NP–completeness is just the start: having accomplished such, attention turns to a whole arsenal of mechanisms which have been proposed to force the over-demanding within tractable limits. Some of these approaches are reviewed within Section 7.

Before any of these researches – algorithm efficiency and problem complexity – can have a foundation established in argumentation a common basis for their investigation is essential. This is found not in propositional logic nor predicate calculus, not in some vague notion of natural language interpretation and reasoning: it is found in the classical structure of directed graphs. It is the landmark in the study of Computational Argument: the Abstract Argumentation Frameworks of Dung [27].

[1] There is an argument for Shannon [43], but the basic model is very different from that of classical complexity theory.

2. In the beginning

In directed graphs we have *nodes* and a relationship between pairs of nodes which is not required to be symmetric: the *links* or edges. The model of abstract argument proposed by Dung [27] views nodes as arguments and a link from argument p to argument q as expressing the fact "argument p *attacks* argument q". Its key assumption is that the arguments, within this structure, are atomic and indivisible. It does not attempt to rationalize *why* argument p attacks argument q but simply states it as an aspect of whatever scenario is being modelled. This separation of *what* is being described from *how* it is described avoids concerns with soundness and rationality: the inner level intricacies underlying the actual structure of a stated position are hidden. One *can*, of course, drill down into the formal structure of an atomic argument within Dung's formalism but in doing so the result will not be to treat a single node differently but to substitute a new collection of atomic arguments presenting the detail of what is being replaced. It is, however, in the reduction of argument relationships to *solely* the concept of attack that Dung's model achieves much of its power. If an argument attacks another then an immediate consequence is that the two cannot simultaneously be accepted by a rational agent: there is an inherent conflict in belief. But we can go further, if a chain of three arguments is such that x attacks y which attacks z then, in principle, both x and z may be rationally accepted since should an adversary dispute the validity of z by proposing y then y can in turn be disputed by advancing x. It is this interplay of attack and defence (although this term is not used by Dung) that positions Dung's abstraction as a powerful dialectical modelling system.

Look again at the two very basic examples of attack structure presented in the preceding paragraph: "x attacks y" and "x attacks y attacks z". In the former it is reasoned that at most one of the pair can be held; in the latter that because the attack on z has been countered the position described by the set $\{x, z\}$ is tenable. In these are found the other significant contribution of abstract argumentation frameworks. The interpretation of a collection of mutually endorsed positions as a subset of graph nodes satisfying given criteria: what has become known as abstract argument *semantics*.

In Dung's paper a succession of these is built up from the most basic (*conflict-free*) through more refined concepts (*admissible*, *complete*) culminating in quite sophisticated ideas (*preferred*, *stable*, *grounded*). As I recap in the next section these basic six would soon be added to in conjunction with the practice of developing new models that continues to be a feature of Computational Argument in the present day.

3. Schism: semantics and models

Dung's paper [27] appeared in 1995: introducing the basic graph model, methods for interpretation (i.e. argument semantics), properties of these semantics, more advanced notions (coherence, controversial arguments, infinite structures). By the time the first Computational Models of Argument conference was held in 2006 [18] not to mention the seminal special issue of the leading Artificial Intelligence research journal [7], Dung's basic model and half dozen semantics had burgeoned into such as semi-stable semantics (Caminada [11]), bipolarity (Cayrol and Lagasquie-Schiex [13]), preference-based argument (Amgoud and Cayrol [1]), symmetric frameworks (Coste-Marquis *et al.* [17]), Value-based argument frameworks (Bench-Capon [6]). Later yet would see Extended Argument Frameworks (Modgil [40]), half a dozen forms of weighted and probabilistic schemes, and, already in 2006 we had hypergraph structures offered as an alternative to the simple directed graph form (Nielsen and Parsons [42]).

Now it is really a matter of no importance *what* these objects are and how they are defined: all that is relevant is that they *are at all*. For what does the existence of a dozen different variant semantics and models say about Dung's approach?

One might claim that this proliferation indicates and attempts to correct some "inadequacy" or failure in Dung's model, but I would consider this view to over simplify. Take the case of semi-stable semantics. On the surface this deals with a problem with Dung's stable semantics: there are systems for which no stable semantics exists whereas every framework has defined semi-stable solutions and, furthermore, these coincide with stable sets should such be present. One can debate the extent to which the semi-stable approach affords a solution to this non-existence issue (I have, however, no intention of doing so). The point is simply that without exception all of the proposed new models and new semantics serve a purpose in that these have been anchored within some *perceived* problem arising in Dung's model, and irrespective of how such problems are addressed typically the basic formalism remains graph-theoretic and the accompanying semantics set-theoretic. While cases such as Modgil's Extended Argumentation Framework [40], let alone the recursive attack and infinite frameworks of Baroni *et al.* [3,4,5] stretch the notion of "directed graph" and "subset semantics" considerably these act in a manner which I would say remains recognizable.

In as much as there may be issues with, to use Simari's evocative phrase, "a plethora of semantics" (and models)[2] I think it is more a pedagogical concern than an inherent structural weakness in Dung's model: the effort of sifting and distinguishing the vast panoply of different suggestions and ideas presents steep demands for neophytes to the world of Computational Argument. This is potentially confusing and debatably unfortunate: what it is *not*, however, is an indication that its base is fundamentally flawed. In fact, when we look at Computational Complexity in Argument, as I shall now turn to, we find a remarkable unity in how these different methods can all be analysed. That such unity is possible is, in no small degree, due to the astonishing versatility of a quite fundamental mechanism: the Standard Translation of Dimopoulos and Torres [16].

4. Breakthrough: The Standard Translation

When Cook in [15] presented the first[3] NP–complete problem he adopted the problem of propositional satisfiability for this end: given an arbitrary propositional formula $\varphi(X)$ over a variable set X determine if there is a setting of X that will make φ **true**. In essence Cook showed that the behaviour of any reasonable computational system, such as a Turing machine, could be mimicked by asking about the satisfiability of an efficiently constructed propositional formula. Cook was able to refine the construction so that it continued to be valid for a special class of formulae: those presented in *Conjunctive Nor-*

[2] The distinguished contributor to work on Computational Argument, Guillermo Simari, coined the phrase "*plethora of semantics*" in his talk at the inaugural COMMA [39] describing the state that had already been reached.

[3] *pace* the claims of Levin [36].

mal Form (CNF). In what is now referred to as *The Standard Translation* a CNF-formula is changed into a Dung-style argumentation framework in which a named argument is accepted under some set-theoretic semantics if and only if the source CNF is satisfiable.

I will not repeat the details of this translation here: the interested reader may find these described in a number of general survey works. e.g. [25] and also within relevant specialist articles. It suffices to observe that the Standard Translation describes any CNF formula as a tripartite structure: an argument representing the formulae itself; a second set describing the clauses[4]; a final set presenting the individual literals ($\{x_i, \neg x_i : 1 \leq i \leq n\}$) used in the formula. Its nature is such that the formula argument will belong to a set satisfying the criteria of a given semantics if and only if a selection of the literal arguments can be made such that setting these to **true** provides a satisfying assignment of the original formula. From this elegant and basic construction the first demonstrations of intractable behaviour in abstract argumentation frameworks are obtained: credulous acceptance under Dung's admissibility semantics and the existence of a stable system of arguments are both shown to be NP–complete. The device also suggests some algorithmic directions. For just as a CNF-formula can be transmuted into an argument framework so too some properties of an argument framework can be encapsulated within propositional and thence CNF structures, e.g. Egly and Woltran [29].

With the sole exception of preference-based schemes the view provided by the Standard Translation has been adopted to demonstrate intractability results in all of the variant forms I mentioned earlier (value-based, weighted, extended); with respect to novel semantics (semi-stable, resolution-based, cf2, ideal, stage, naive) and with respect to different graph-theoretic restrictions (planar, acyclic, binary tree). It finds new applications within the study of argument semantics signatures and the concept of realizability (Dunne *et al.* [23]; Dvořák *et al.* [26]). It is not limited to demonstrations of intractability at the level of mere NP-completeness being powered up to Π_2 and Σ_2 completeness results in the cases of sceptical acceptance in Dung's preferred semantics (Dunne and Bench-Capon [21]), and, as demonstrated by Dvořák and Woltran [28], with respect to semi-stable questions. It links basic argumentation dialogue games to a very primitive (but sound and complete) proof calculus (Vreeswijk and Prakken [46]), opening up another avenue for complexity-theoretic studies (Dunne and Bench-Capon [22]).

5. The concept of Canonical Problem

In comparing one approach and semantics against an alternative technique – especially in the context of complexity and algorithmic study – we need to have some benchmark collection of ideas for such comparison. The formulation of 4 standard problems[5] provides this.

Hence given any proposed semantics, σ and graph-based model $M(\mathcal{X}, \mathcal{A})$ we have:

A. Existence (does any non-empty collection of items within M satisfy the criteria set by σ?)

[4] A clause being a disjunction of literals.

[5] The list presented in [25, Table 1.1] separates the existence problem into two: one allowing empty sets to satisfy criteria and another (called non-emptiness in [25]) phrased identically to that of our Existence problem. I have chosen to eschew considering the former problem as canonical.

B. Credulous Acceptance (is there some collection satisfying the conditions of σ and containing this argument, p?)
C. Sceptical Acceptance (does every collection satisfying σ have p as a member?)
D. Verification (does this collection meet the conditions prescribed by σ?)

From such a basis is obtained the standard agenda for complexity-theoretic analysis of new semantics and models: for each of the canonical problems determine exact bounds on its complexity. Notice this involves both algorithmic construction (witnessing that a problem *can* be solved within a particular resource bound) and demonstration that such algorithms cannot be significantly improved (categorization within a specific complexity class). Having addressed the basic questions attention often turns to variations within the model itself, e.g. graph-theoretic restrictions such as planarity, k-colourability, acyclicity.

6. What have Algorithmic Study and Complexity given to Argument?

Some of Dung's semantics can be dealt with efficiently for all of the canonical problems: for example the grounded semantics. Others, notably the preferred, not known to have efficient methods for any and, in fact, have strong indicators that no such methods are possible. Other mix the trivial (sceptical acceptance in the conflict free semantics) with straightforward efficient methods (verification in the naive semantics).

Such rough division may seem already to provide a case for advocating one semantics in preference to another. This, however, if applied naively does not give a good basis. One might put forward (at least) one reason as to why an efficiently solvable semantics (for example the grounded) is not necessarily to be favoured over one less so (for example the preferred semantics). Often it is found that the more tractable cases are only so on account of their limited expressive ability: hence the grounded semantics offers useful outcomes only within the sub-class of argument frameworks defined by directed graphs having at least one source node. Conflicting viewpoints, however, tend to produce models in which every argument attacks and is attacked by another: here the grounded semantics offers no information. Thus in total there is a balance between expressibility in modelling terms and algorithmic tractability with respect to the canonical problems.

While complexity-theoretic analysis does not realistically inform the *choice* of a semantics, it does provide an *awareness* of potential issues: the knowledge that the verification problem for preferred semantics is coNP–complete may be insufficient to reject it, however, it may, having identified some admissible set, discourage attempts to push that set to maximality.

The significant contribution that complexity theory has made to the study of argument is, I would say, in this notion of awareness. Irrespective of the semantic formalism, irrespective of the graph-theoretic model each of the four canonical problems raises a single question: is it possible to solve this problem efficiently? Of course I would not claim the related questions had been ignored prior to the first complexity studies of abstract argumentation, but I would claim that an awareness of the underyling issues is heightened by considering matters in a uniform style: that of the four canonical questions with respect to semantic criteria.

With respect to algorithmic contributions, it is appropriate at this point to raise one method which, in principle, offers an alternative semantic treatment: the labelling approach championed, most notably by Verheij [45] and Caminada (see, e.g. Modgil and

Caminada [41]). The basic labelling semantics uses three labels called IN, OUT, UNDEC with different semantics captured by the configurations allowed within legal labellings. Such approaches offer a solution to questions of semantics (via those arguments legally assigned the label IN) and algorithms (in processes for allowing a final labelling to evolve from an initial default labelling). In [41, Section 7, 127–8], Modgil and Caminada offer a cogent summary of the gains achieved through labelling semantics and, of interest from the perspective of the current topic, algorithms. Labelling is not, of course, some latter day Philosopher's Stone, turning the base metal of intractability to desired efficient solutions. It does, however, as noted by Modgil and Caminada offer useful insight into dialectical processes thereby linking esoteric algorithmic matter to how argument may be recognized in the "real world".

The algorithmic contribution to argument therefore consists not only of the tangible gains in computational efficiency but also in the rather more subtle effect of providing insight into the mechanism of argument itself.

In total, computational complexity theory provides evidence (albeit circumstantial) that looking for efficient under all conditions algorithms for a number of argumentation problems will be fruitless. Formal algorithm study has contributed efficient methods, and for many special cases, useful practical methods. Very often these stray into quite advanced algorithmic ideas, e.g. fixed-parameter tractability and its specific case of Courcelle's Theorem. Nonetheless I would claim that there are many avenues, well studied in classical algorithm theory as an angle on intractability, the investigation of which in argumentation has been barely touched. What are these? That is the question considered in the next section.

7. Omissions and Neglect

I remarked earlier that in the classical study of algorithms and complexity a demonstration that a problem is likely to be unsolvable does not signal the end of further investigation. Instead a whole range of possible means of coping come into play: randomized and probabilistic methods; approximation techniques, special cases, average case efficiency, fixed parameter tractable representations, backdoor techniques. Now it is certainly true that a number of these have been considered in computational argument. For example special case study dates back at least as far as the work of Coste-Marquis *et al.* [17] on symmetric frameworks; bipartite forms are shown to be tractable in Dunne [19]. Similarly a reasonable volume of work has accrued in the study of fixed parameter and backdoor methods.

Despite these, there are areas which have been at best neglected, at worst overlooked entirely. This may partly be a matter of fashion (for example the once very thriving area of so-called phase-transition phenomena, e.g.[14], where there have been only superficial studies). Phase-transition phenomena, in much as there is algorithmic potential, have always seemed to me, personally, to have elements of "smoke-and-mirrors", some of these aspects being discussed in Dunne, Gibbons and Zito [24]. So possibly it is not surprising that a full scale study of threshold phenomena in argumentation has yet to be undertaken: here is an approach becoming a historical curiosity, that promises much (efficient algorithmic solutions for generally intractable problems) but in reality actually delivers little of practical use (that is to say, there *are* efficiently *on average* fast methods but these are for cases on the verge and not for the classes of instance that will be seen in real contexts). So the famous "conjuring trick"[6] of Angluin and Valiant [2] is dependent on a combinatorial property of random graph structures: as with all good conjuring tricks the effect at first surprising loses interest once seen how it is achieved.

If phase transition phenomena are ideas whose time has passed the rather more focussed and related concern of average-case study is a very different matter. This, again, is an approach for which only superficial studies (if that) have been undertaken. Average-case studies work from a base in which input instances are chosen at random according to some probability distribution. The forms being studied in argumentation would thus be treated as random directed graph structures. In order to avoid the combinatorial legerdemain underpinning the performance in [2] (typical random directed graphs contain many links and this specific method performs well on such graphs), reliable evidence for usable average case argumentation algorithms needs to focus on graphs typical of those seen in reality. I would claim that the following problems have yet to be fully considered:

T1. Develop a model of random argumentation frameworks that reflects the characteristics of typical frameworks.
T2. Develop methods for generating random representatives within this model.
T3. Extend these to value-based (VAF) and abstract dialectical frameworks (ADFs).

Average-case investigation is well established as a field within algorithmics and has attracted a modicum of interest in argumentation. There is, however, another well studied (in graph theoretic terms) approach whose relevance to argumentation has received minimal attention. I refer here to the study of spectral properties. Directed graphs may be described as $(0, 1)$-matrices and the analysis of the eigenvalues and eigenvectors of $(0, 1)$-matrices has historically offered some insight into graph-theoretic problems, e.g. node colouring, Wilf [47]. One significant use of spectral analysis in other areas has been its application to ranking problems, most notably in the mechanics underpinning Google's search algorithm, see [9] but also other ranking environments, e.g. Keener [34], Kleinberg [35]. Such approaches and the importance of ranking as an issue in argumentation, e.g. Bonzon *et al.* [8] leave, to my mind, the comparative neglect of spectral analysis rather puzzling. A very basic and preliminary investigation is reported in [10], however its findings are very inconclusive. The following questions are, I think, worthy of more detailed study:

S1. Google's ordering approach involves identifying an eigenvector of a dominant eigenvalue within a rational valued matrix defined from webpage linkages. The linkage structure is a directed graph. In principle treating an argument framework in such a manner might give some insight into argument "importance". There is, however, a complication: in simple terms Google's page significance function is cumulative (if page X links to page Y which links to page Z, the score assigned Z will have positive contributions from the scores of X and Y). A naive translation to argument runs into an immediate problem: a higher score for X should result in a lower score for Y and thence a higher score for Z. The side-effect of non-monotonicity raises the question of formulating scoring functions for argument (akin to Google's technique) that might allow analysis of argument ranking as the ordering of components in an eigenvector.

[6] This description is a little bit unfair, however not one that I suspect the authors would dispute.

S2. The study in Butterworth and Dunne [10] looks at potential relationships between the eigenvalues associated with an argument and its acceptability under different semantics. A full comparative study of this has yet to be carried out.

I choose these simply as significant broad areas which, in my view, have yet to be the subject of sustained and systematic study. There are other specialist techniques found helpful in algorithm yet untried in argument. It may, also, be the case that such detailed studies will yield nothing of interest. We will not know this, however, unless we try.

8. Conclusion: “are we there yet?”

The trite and obvious answer to this question is, of course, not by some margin. What, however, do I intend precisely by “there”? A reasonable view would be to equate this with the general agenda of algorithmic and computational complexity studies. That is to say the classification of problem difficulty (computational complexity theory) combined with concerted attacks aimed at exorcising the worst side-effects of intractable behaviour. I think there can be no question that with respect to the first of these there has been notable success: one struggles to think of any significant argumentation problem (certainly with respect to the classical semantics of Dung [27]) whose complexity status remains open. Similar comprehensive achievements have been delivered with respect to new semantics (semi-stable, resolution-based, and, with one niggling gap, ideal) and also within alternative models (value-based, extended argumentation frameworks).

Against these contributions, my feeling is that too little attention has been given to the nature of efficient algorithmic methods. This is understandable, the analytic acrobatics brought to bear in engineering some intractability proofs (I trust the reader will excuse my citation of [19, Thm. 12], let alone [20, Corollary 5]) can be hard to resist. This, however, should not detract from the fact that the decision problems addressed arise from a real application setting, and thus effective solution, in at least as much as such can be developed, is a necessity. If there is one urgency I would identify from the body of work produced so far it is that of addressing algorithmic approaches.

References

[1] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematematics and Artificial Intelligence*, 34(1):197–215, 2002.

[2] D. Angluin, L.G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18(2):155–193, 1979.

[3] P. Baroni, F. Cerutti, P.E. Dunne, and M. Giacomin. Computing with infinite argumentation frameworks: the case of AFRAs. In *Proc. 1st Intnl. Workshop on Theory and Appl. of Formal Argumentation (TAFA)*, pages 197–214, 2011.

[4] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida. AFRA: argumentation framework with recursive attacks. *Int. J. Approx. Reason.*, 51(1):19–37, 2011.

[5] P. Baroni, F. Cerutti, P. E. Dunne, and M. Giacomin. Automata for infinite argumentation structures. *Artificial Intelligence*, 203:104-150, October 2013

[6] T.J.M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Jnl. Logic and Computation*, 13(3):429–448, 2003.

[7] T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[8] E. Bonzon, J. Delobelle, S. Konieczny, and N. Maudet. A comparative study of ranking-based semantics for abstract argumentation. In *Proc. of the 30th AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 914–920, 2016.

[9] K. Bryan and T. Leise. The $25,000,000,000 eigenvector: The linear algebra behind Google. *Siam Review*, 48(3):569–581, 2006.

[10] J. Butterworth and P. E. Dunne. Spectral techniques in argumentation framework analysis. In *Proc. 6th COMMA*, volume 287 of *FAIA*, pages 167–178. IOS Press, 2016.

[11] M. Caminada. Semi-stable semantics. In *Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006, pages 121–128.

[12] G. Cardano. *Artis Magnae, Sive de Regulis Algebraicis*. 1545.

[13] C. Cayrol and M. C. Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU)*, pages 378–389, 2005.

[14] P. Cheeseman, B. Kanefsky, and W. Taylor. Where the really hard problems are. *Proc. of the 12th IJCAI*, pages 331–337, 1991.

[15] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158. ACM, 1971

[16] Y. Dimopoulos and A. Torres. Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.*, 170(1-2):209–244, 1996.

[17] S. Coste-Marquis, C. Devred, and P. Marquis. Symmetric argumentation frameworks. In L. Godo, editor, *Proc.* $8^{\rm th}$ *European Conf. on Symbolic and Quantitative Approaches to Reasoning With Uncertainty (ECSQARU)*, volume 3571 of *LNAI*, pages 317–328. Springer-Verlag, 2005.

[18] P. E. Dunne and T. J. M. Bench-Capon (editors) *Computational Models of Argument: Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006.

[19] P. E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171:701–729, 2007.

[20] P. E. Dunne. The computational complexity of ideal semantics, *Artificial Intelligence*, 173(18):1559–1591, 2009.

[21] P. E. Dunne and T. J. M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 141(1/2):187–203, 2002.

[22] P. E. Dunne and T. J. M. Bench-Capon. Two party immediate response disputes: properties and efficiency. *Artificial Intelligence*, 149:221–250, 2003.

[23] P. E. Dunne, W. Dvořák, T. Linsbichler, and S. Woltran. Characteristics of Multiple Viewpoints in Abstract Argumentation. *Artificial Intelligence*, 228:153-178, November 2015

[24] P. E. Dunne, A. Gibbons, and M. Zito. Complexity-theoretic models of phase transitions in search problems. *Theoretical Computer Science*, 249(2):243–263, 2000,

[25] P. E. Dunne and M. Wooldridge. Complexity of abstract argumentation. In I. Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, Springer, Berlin, pages 85–104, 2009.

[26] W. Dvořák, P. E. Dunne and S. Woltran. Parametric Properties of Ideal Semantics. *Artificial Intelligence*, 202:1-28, September 2013

[27] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[28] W. Dvořák and S. Woltran. Complexity of semi-stable and stage semantics in argumentation frameworks. *Inf. Process. Lett.*, 110(11):425–430, 2010.

[29] U. Egly and S. Woltran Reasoning in Argumentation Frameworks using Quantified Boolean Formulas In *Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006, pages 12 133–144.

[30] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman New York, 2002.

[31] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. *Monatshefte für Mathematik und Physik*, v. 38 n. 1, 173–198, 1931

[32] J. Hartmanis and R. E. Stearns, On the Computational Complexity of Algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[33] D. S. Johnson, A Catalog of Complexity Classes, Algorithms and Complexity (editor: J. van Leeuwen), Handbook of Theoretical Computer Science, Elsevier, 67–161, 1990.

[34] J. P. Keener. The Perron-Frobenius theorem and the ranking of football teams. *SIAM Review*, 35(1):80–93, 1993

[35] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604–632, 1999

[36] L. A. Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.

[37] F. Lindemann. Über die Ludolph'sche Zahl. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*, 2:679–82, 1882.

[38] F. Lindemann. Über die Zahl π. *Math. Ann.*, 20:213–225, 1882.

[39] D. C. Martínez, A. J. García and G. R. Simari. On acceptability in abstract argumentation frameworks with an extended defeat relation. In *Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006, pages 273–278.

[40] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9–10):901–934, 2009.

[41] S. Modgil and M. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I. Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.

[42] S. H. Nielsen and S. Parsons. Computing preferred extensions for argumentation systems with sets of attacking arguments. In *Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006, pages 97–108.

[43] C. E. Shannon. The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1):59–98, 1949.

[44] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(1):230–265, 1937.

[45] B. Verheij. A labeling approach to the computation of credulous acceptance in argumentation. In *Proc. 20th IJCAI*, pages 623–628, 2007.

[46] G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proceedings of JELIA'2000, The 7th European Workshop on Logic for Artificial Intelligence.*, pages 224–238, Berlin, 2000. Springer LNAI 1919, Springer Verlag.

[47] H. S. Wilf. The eigenvalues of a graph and its chromatic number *Journal of the London Mathematical Society*, s1-42 (1):330–332, 1967

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220135

Detect – Debunk – Communicate: Combating Misinformation with More Realistic NLP

Iryna GUREVYCH [a,1]
[a] *Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt*
ORCiD ID: Iryna Gurevych https://orcid.org/0000-0003-2187-7621

Abstract. Dealing with misinformation is a grand challenge of the information society directed at equipping computer users with effective tools for identifying and debunking misinformation. Current Natural Language Processing (NLP) including fact-checking research fails to meet the requirements of real-life scenarios. In this talk, we show why previous work on fact-checking has not yet led to truly useful tools for managing misinformation, and discuss our ongoing work on more realistic solutions. NLP systems are expensive in terms of financial cost, computation, and manpower needed to create data for the learning process. With that in mind, we are pursuing research on detection of emerging misinformation topics to focus human attention on the most harmful, novel examples. We further compare the capabilities of automatic, NLP-based approaches to what human fact checkers actually do, uncovering critical research directions for the future. To edify false beliefs, we are collaborating with cognitive scientists and psychologists to automatically detect and respond to attitudes of vaccine hesitancy, encouraging anti-vaxxers to change their minds with effective communication strategies.

Keywords. fake news, misinformation, disinformation, fact-checking, low-resource NLP, rebuttal

[1] Corresponding Author

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220136

Argumentation: From Theory to Practice & Back

Antonis KAKAS
University of Cyprus, Cyprus

Keywords. Machine Argumentation, Applied Argumentation, Cognitive Argumentation

Machine and Human Argumentation differ in many respects. Yet, to have useful and effective applications of argumentation in AI these two forms of argumentation need to come close so that we have a natural form of interaction between humans and machines. This closeness or compatibility between machine and human argumentation is needed not only at the level of the form of communication but also at the internal operational level of the argumentation process. For example, the capability of an argumentative dialogue by an argumentation-based system and the usefulness of the explanations it offers would be enhanced when there is a deeper form of compatibility between the argumentative reasoning processes of these systems with that of human reasoning.

In order for real-life applications of argumentation based systems to achieve such a high-degree of natural compatibility while operating in an external dynamic environment that includes the "human in the loop", we need to address two **major challenges**:

- **Acquisition of Application Knowledge** What is an appropriate language level that would facilitate capturing the application knowledge either from the application expert and/or the application data? What is the appropriate cognitive-level of this language? Can this be (structured) Natural Language?
- **Middleware from Sensory Information to High-level Application Concepts** What are effective ways of comprehending the relevant part of the current application environment? How do we recognize the current state of affairs and the particular decision context in which the system finds itself?

To address these challenges we need software methodologies that facilitate the development of systems directly from the high-level application domain language, data and expertise. One such methodology is *SoDA* which together with the systems of *Gorgias* and *GorgiasCloud* offers Explainable Argumentation as a Service for online applications. Recently, these technologies have formed the basis for a start up company called **Argument Theory** in Paris which offers solutions to real-life application decision taking problems based on argumentation technology. Its first successful application concerns automated help in the annotation of documents for blind readers, while currently it is developing prototype systems for applications in the areas of medical decision support, personal assistants and policy compliance.

Such applications emphasize the need for a human-like form of machine argumentation. To help us address this we can study the synthesis of cognitive principles within formal computational frameworks of argumentation. Cognitive principles are drawn from our understanding of human reasoning as acquired across a wide range of disciplines, such as Cognitive Science, Philosophy and Linguistics. They would inform and regulate the computational process of argumentation to be cognitively compatible to human argumentation and reasoning. **Cognitive Argumentation** concerns such a study which together with its *COGNICA* system for explainable conditional reasoning, offers the opportunity for carrying out large scale empirical studies of human-machine reasoning interaction. For example, *COGNICA* is used to study the effect that machine explanations can have on humans when reasoning or deciding what action to pursue.

This page intentionally left blank

Regular Papers

This page intentionally left blank

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220138

Generating Contrastive Snippets for Argument Search

Milad ALSHOMARY, Jonas RIESKAMP and Henning WACHSMUTH
Paderborn University, Paderborn, Germany

Abstract. In argument search, snippets provide an overview of the aspects discussed by the arguments retrieved for a queried controversial topic. Existing work has focused on generating snippets that are representative of an argument's content while remaining argumentative. In this work, we argue that the snippets should also be *contrastive*, that is, they should highlight the aspects that make an argument unique in the context of others. Thereby, aspect diversity is increased and redundancy is reduced. We present and compare two snippet generation approaches that jointly optimize representativeness and contrastiveness. According to our experiments, both approaches have advantages, and one is able to generate representative yet sufficiently contrastive snippets.

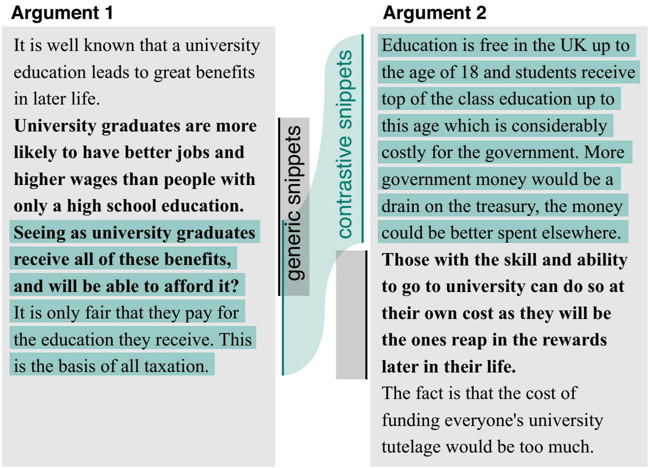
Keywords. snippet generation, argument search, argument presentation

1. Introduction

Most search engines present results along with short text excerpts of the underlying documents, so called *snippets*, in order to let users quickly assess the relevance of the results to their information needs [1]. In argument search, where the goal is to retrieve pro and con arguments on a queried controversial topic [2,3], snippets are of particular importance to provide an efficient overview of the spectrum of topical *aspects* covered by the retrieved arguments—without the need to go through all of them [4].

Standard snippet generation focuses on the overlap of the input query with the document [5,6], or abstractions thereof [7]. In the context of argument search, however, Alshomary et al. [4] argued in favor of snippets that summarize an argument's main claim and the main reason supporting the claim. In their experiments, the authors demonstrated that snippets generated towards this goal are more representative than generic content summaries and query-dependent snippets.

In this paper, we highlight the limitations of such argument snippet generation and propose an extended setting for the task in order to maximize the usability of the resulting snippets for argument search engines. Particularly, the extractive summarization approach of Alshomary et al. [4] addressed two goals of snippet generation: *representativeness* and *argumentativeness*. However, the top-ranked arguments retrieved by an argument search engine usually discuss the same queried controversial topic. Hence, an approach that aims to extract the main claim of an argument will tend to generate semantically similar snippets for several arguments. This behavior is highlighted in Figure 1, where two pro arguments are shown for the query "tuition fees". Here, a focus on



Argument 1

It is well known that a university education leads to great benefits in later life. **University graduates are more likely to have better jobs and higher wages than people with only a high school education. Seeing as university graduates receive all of these benefits, and will be able to afford it?** It is only fair that they pay for the education they receive. This is the basis of all taxation.

Argument 2

Education is free in the UK up to the age of 18 and students receive top of the class education up to this age which is considerably costly for the government. More government money would be a drain on the treasury, the money could be better spent elsewhere. **Those with the skill and ability to go to university can do so at their own cost as they will be the ones reap in the rewards later in their life.** The fact is that the cost of funding everyone's university tutelage would be too much.

Figure 1. Example arguments returned for the query "tuition fees" (we show only two here for simple illustration). In each case, the bold sentences represent a generic snippet for the respective argument, whereas those with a colored background form a more contrastive argument snippet.

only representativeness and argumentativeness would likely produce similar snippets for both arguments (e.g., based on the sentences marked bold), reducing the argument search engine's capability to provide an effective aspect overview.

To alleviate the outlined limitation, we propose to additionally maximize the *contrastiveness* of a snippet. We define an argument snippet as contrastive, if it highlights the uniqueness of the input argument compared to other arguments from the same context (say, those from the same result page in argument search). The input to our approach is a set of arguments from the same context. The output is a set of snippets that are argumentative and representative of their argument while being contrastive toward the others. By focusing more on contrastiveness, the new snippets (shown with a colored background in Figure 1) increase the coverage of the diverse aspects discussed in the input arguments. Naturally, achieving higher contrastiveness might result in lower representativeness; a trade-off that can be adjusted, as we show later in experiments.

We approach the extended task setting in two ways. First, we extend the graph-based approach of Alshomary et al. [4], which ranks sentences based on their centrality and argumentativeness, by contrastiveness. Here, we encode an extra term to discount the sentence's similarity to other arguments. Second, we exploit the resemblance of our setting to the comparative summarization task [8]. Concretely, we adapt the approach of Bista et al. [9] who model the latter task as the selection of a snippet that a powerful classifier can distinguish from other arguments but not from the input argument.

In our experiments, we evaluate the approaches on a dataset constructed from the corpus of the argument search engine *args.me*. In particular, we use controversial topics from Wikipedia along with entries from the query log of args.me itself [10] to retrieve argument collections from the args.me API. We quantify representativeness by computing the cosine similarity of a snippet with the average embedding of its argument, and we measure argumentativeness as a quality dimension using the model of [11] trained to measure the argumentative quality of sentences. For contrastiveness, finally, we adapt silhouette analysis [12] as a proxy to measure the quality of clusters whose centroids are the generated snippets. Our results demonstrate the trade-off between representativeness and contrastiveness, and they indicate how to balance this trade-off.

In a subsequent user study, we manually compared our approaches to a baseline that focuses only on representativeness. We demonstrate that both approaches generate snippets that highlight the arguments' uniqueness better, whereas the comparative summarization approach produces more representative snippets. Our findings support the applicability and importance of considering the contrastiveness of a snippet within argument search. For reproducibility, we make our code and resources publicly available.[1]

2. Related Work

For snippet generation in general, abstractive and extractive summarization techniques have been explored [13,14]. In both cases, a user's query may be considered during generation (query-dependent snippet) or disregarded (query-independent). Alshomary et al. [4] suggest that query-independent snippets are more suitable in an argument search scenario. Therefore, we also consider such snippets in this paper. The authors proposed a graph-based approach that forms snippets by selecting the two most important sentences in terms of their centrality in context and their argumentativeness. In contrast, we argue that a snippet should also highlight the argument's uniqueness in its context to maximize the diversity of snippets when presented to the end-user.

A branch of summarization research focuses on comparative summarization. Given a set of document groups, the task here is to generate summaries that are useful in comparing the differences between these groups [8,15]. Similarly, our goal is to obtain snippets that emphasize the differences between texts. However, our input is a set of arguments rather than groups of documents. Thus, we modify the recent comparative summarization approach of Bista et al. [16] by considering different surface features to maintain the snippet's argumentativeness.

There exists a body of research on the diversification of search results [17] that aims to retrieve diverse results while maintaining relevance to the queried topic in order to provide wider perspective. Differently, in this work, we are already given a set of arguments retrieved for some topic, and we aim to generate diverse snippets, where each snippet highlights the unique argumentative part of its argument.

3. Approaches

For the task of contrastive argument snippet generation, we define the input to be a set of $k \geq 2$ arguments $\mathbf{A} = \{A_1, \dots, A_k\}$ from the same context, for example, all arguments from a search engine's result page. We represent each $A \in \mathbf{A}$ simply as a set of sentences, $A := \{s_1, \dots, s_n\}$, where $n \geq 2$ usually differs across arguments. The output is one subset $S \subseteq A$ for each A, consisting of all sentences of the snippet (we predefine $|S| = 2$ in our experiments). Ideally, S is representative of A, argumentative on its own, and contrastive towards all arguments in $\mathbf{A} \setminus \{A\}$.

In this section, we propose two alternative approaches to the defined task. The first, *Contra-PageRank*, extends the work of Alshomary et al. [4] by modeling the dissimilarity

[1] https://github.com/MiladAlshomary/contrastive-snippet-generation

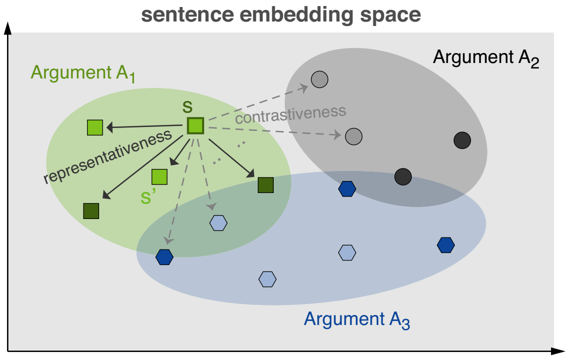


Figure 2. The idea of both approaches, illustrated for three arguments in a sentence embedding space. A sentence s is used for a snippet of an argument A_1, if its joint representativeness and contrastiveness are higher than for other sentences s' of the same argument. Argumentativeness (brighter symbols) is considered as well.

of each sentence $s_i \in A$ to all sentences from $\mathbf{A} \setminus \{A\}$. The second, *Comp-Summarizer*, adapts the work of Bista et al. [16] to select a snippet S that can be distinguished from $\mathbf{A} \setminus \{A\}$ but not from A. Both thus follow the idea to include a sentence s in S, if it is both representative of A and contrastive to $\mathbf{A} \setminus \{A\}$. We illustrate this idea in Figure 2.

3.1. Graph-based Summarization

Alshomary et al. [4] proposed a graph-based approach that utilizes PageRank [18] to score sentences in terms of their centrality in context and argumentativeness. The two top-scored sentences are then extracted in their original order to form a snippet. We modify the underlying scoring function $P(s_i)$ in two ways: First, we compute the centrality of a sentence $s_i \in A$ based only on the sentences in its covering argument A rather than all sentences from the whole context of $\mathbf{A}$—to avoid conflicts with our second adaptation. Second, we extend the bias term that represents the initial sentence probability to account not only for argumentativeness (*arg*) but also for contrastiveness. The contrastiveness here is quantified as a discount on the similarity (*sim*) of s_i to other arguments in the context. As a result, we reformulate the PageRank score of $s_i \in A$ as follows:

$$P(s_i) := \quad d_1 \cdot \sum_{s_j \in A, i \neq j} \frac{sim(s_i, s_j)}{\sum_{s_k \in A, j \neq k} sim(s_j, s_k)} \cdot P(s_j) \tag{1}$$

$$+ d_2 \cdot \frac{arg(s_i)}{\sum_{s_j \in A} arg(s_j)} \quad - \quad d_3 \cdot \frac{sim(s_i, \mathbf{A} \setminus \{A\})}{\sum_{s_j \in A} sim(s_j, \mathbf{A} \setminus \{A\})} \tag{2}$$

Here, the argumentativeness score $arg(s_i)$ of each $s_i \in A$ and the similarity score $sim(s_i, \mathbf{A} \setminus \{A\})$ are computed directly to form the initial bias score of each sentence. Following the original approach [4], a graph is then constructed for each argument A by modeling each sentence $s \in A$ as a node and creating an undirected edge $\{s, s'\}$ for each pair $s, s' \in A$, $s \neq s'$, weighted with $sim(s_i, s_j)$. Finally, PageRank is applied to generate a score $P(s)$ for each s. As in [4], we start with equal initial scores for all sentences and iteratively update them until near-convergence. We rank all sentences of a given argument A by score and generate a snippet from the two top-ranked sentences concatenated in their original order.

3.2. Comparative Summarization

Given the resemblance of our task to comparative summarization, we model the task in line with the mentioned approach of Bista et al. [16]: For an argument A, the goal is to find snippet sentences $S \subseteq A$ subject to (a) S being representative of A, and (b) S being contrastive to $\mathbf{A} \setminus \{A\}$. This is conceptualized via a condition for each objective: (a) No classifier y can be trained that distinguishes sentences in S from those in $A \setminus S$, reflecting the snippet's representativeness. (b) A classifier y' can be trained that can differentiate sentences in S from those in other arguments from the context $\mathbf{A} \setminus \{A\}$, reflecting the snippet's contrastiveness.

Regarding the classifiers, condition (a) aims to minimize the accuracy of y, whereas (b) aims to maximize the accuracy of y'. Since finding such classifiers is an intractable problem in general, [9] used maximum mean discrepancy (MMD) [19] as an estimation of the classifiers' effectiveness. Given a set of arguments $\mathbf{A}$, the goal is then to find the snippet sentences $S \subseteq A$ of all arguments $A \in \mathbf{A}$ that maximize the following term:

$$\sum_{A \in \mathbf{A}} \left(-MDD^2(S, \{A\}) + \lambda \cdot MMD^2(S, \mathbf{A} \setminus \{A\}) \right) \tag{3}$$

Here, λ is a parameter to control the influence of contrastiveness (second addend the term above). This formulation models representativeness based on the similarity between sentences (first addend). It can be solved in an unsupervised way by greedily selecting sentences that satisfy the objective.

However, there may be other features that signal sentence importance which are not reflected by similarity (e.g., argumentativeness in our case). For this, [16] introduced learnable functions that map sentence features into an importance score and integrate them into the objective function of a supervised MMD variant. Given a training set $\mathscr{T}$ with tuples of argument A, generic snippet $\bar{S} \subseteq A$, and context $\mathbf{A}$, the goal is to minimize (note the switched signs) the following adjusted term:

$$\frac{1}{|\mathscr{T}|} \sum_{(A, \bar{S}, \mathbf{A}) \in \mathscr{T}} \left(MDD^2(\bar{S}, A, \theta) - \lambda \cdot MMD^2(\bar{S}, \mathbf{A} \setminus \{A\}, \theta) \right) \tag{4}$$

Here, $\theta \in \mathbb{R}^m$ denotes a vector of learned feature weights. The adjusted variant requires the definition of sentence features that reflect its likelihood of appearing in $\bar{S}$. Hence, we consider the following $m = 6$ features in our implementation:

1. *Position.* Position of the sentence in the argument
2. *Word count.* Number of words in the sentence
3. *Noun count.* Number of nouns in the sentence
4. *TF-ISF.* TF-IDF on the sentence level
5. *LexRank.* Scores obtained from LexRank [20]
6. *Argumentativeness.* Count of words from a claim lexicon [4]

4. Experiments

We now analyze the trade-off between representativeness and contrastiveness in snippet generation, and we explore how to adjust it via hyper-parameters of the two approaches. We will present the data collection and preprocessing, implementation details, as well as the automatic and manual evaluations that we carried out.

4.1. Data

For evaluation, we need a dataset of arguments grouped into contexts. Since our work is motivated by the idea of argument search engines, we use the *args.me* corpus of [10] as the source. Particularly, we considered all arguments in the corpus belonging to the same debate as a context, resulting in 5457 contexts with an average of 5.2 arguments per context, we call it *argsme* dataset. Such contexts suit the training of *Comp-Summarizer* since we can use argument conclusions to derive generic snippets. Second, we mimicked how arguments are grouped into contexts in search by querying the *args.me* API[2] once using Wikipedia's list of controversial issues,[3] and once using queries from the *args.me* query log.

We call the former dataset *controversial-contexts* containing 600 context with an average of 7.5 arguments per context, while the latter is called *query-log* containing 476 contexts with an average of 7.0 arguments. Since query-log is best in representing the realistic search scenario, we use it below for the final evaluation.[4]

4.2. Implementation Details

For both approaches, we preprocess all input arguments in a number of cleansing steps, namely we remove debate artifacts[5], references, enumeration symbols, and sentences shorter than three characters. We measured sentences similarity in terms of the cosine of their embeddings generated with Sentence-BERT [22]. In the following, we give further implementation details of the two approaches.

Contra-PageRank Recall that our graph-based summarization has three parameters, d_1–d_3, for representativeness, argumentativeness, and contrastiveness respectively. In our experiments, we tested different parameter values between 0.1 to 0.9 with a step size of 0.1 on the *controversial-contexts* dataset. We consider *Contra-PageRank* with $d_3 = 0$ as a baseline, since it disregards contrastiveness.

Comp-Summarizer To obtain ground-truth generic snippets $\bar{S}$ that are necessary for the supervised training, we followed the previous work [4] in considering the argument's conclusion a proper generic snippet. To this end, we used the *args.me* corpus and heuristically generated generic snippets based on the sentences' overlap with the conclusion.[6] We assessed different combinations of values for the hyperparameters, including the contrastiveness weight λ. We used 5-fold cross-validation to evaluate each combination, aiming to minimize the average loss on the data. The optimization worked for 300 epochs with a learning rate of 0.1.

[2] https://www.args.me/api-en.html

[3] https://en.wikipedia.org/wiki/Wikipedia:List_of_controversial_issues

[4] All three collected datasets will be made publicly available upon acceptance.

[5] The artifacts are mostly utterances of social interaction between debaters [21].

[6] The algorithm used is the one that was proposed by Bista et al. [16].

Table 1. Automatic evaluation scores of *Contra-PageRank* for three selected combinations of hyperparameter values. The best value in each column is marked in bold.

d_1	d_2	d_3	**Contrastiveness**	**Argumentativeness**	**Representativeness**
1.0	0.0	0.0	0.045	**0.647**	**0.800**
0.5	0.7	0.2	0.050	0.630	0.675
0.8	0.9	0.7	**0.060**	0.622	0.594

4.3. Automatic Evaluation

No datasets with ground-truth contrastive snippets exist, and the manual creation of such snippets is arguably arduous. Therefore, we stick to automatic measures that intrinsically assess snippet quality below, in order to evaluate different parameter value combinations and to select some for the manual evaluation.

In particular, we capture *contrastiveness* in terms of silhouette analysis score, an intrinsic cluster measure for quantifying clusters quality, as follows. Given a set of snippets $\mathbf{S} = \{S_1, \ldots S_k\}$ generated for a set of arguments $\mathbf{A} = \{A_1, \ldots A_k\}$, we pseudo-cluster the embedding of all arguments' sentences, with each snippet S_i as one centroid.[7] This way, we can quantify the clusters' quality using silhouette analysis: The more contrastive snippets are, the better the clusters they form, reflected in a higher silhouette score. As for *representativeness*, we compute the mean similarity between the sentences of a snippet S and those of the respective argument A. Finally, we approximate *argumentativeness* by argument quality, employing the BERT model of [11] trained on a regression task to predict the argumentative quality score of a sentence.[8]

Results Table 1 presents three selected combinations of parameter values that demonstrate the limits of contrastiveness and representativeness for *Contra-PageRank* as well as their trade-off: As expected, setting d_1 to 1 (and, thus, ignoring the other terms) maximizes representativeness, while the best contrastiveness score comes from increasing d_3 to 0.7 (third row). In the second row, we show a value combination that better balances representativeness and contrastiveness. As for argumentativeness, we observed little differences across parameters, which could be the result of the simple lexicon-based method of weighting argumentativeness.[9]

In Table 2, we explore the trade-off between representativeness and argumentativeness for *Comp-Summarizer*, showing evaluation scores for selected values of the contrastiveness weight λ. Analogously, a higher λ results in more contrastiveness but less representativeness, while ignoring the contrastiveness term ($\lambda = 0.000$) leads to the best representativeness. A medium value (here, $\lambda = 0.500$) yields a better balance between the three scores.

4.4. Manual Evaluation

To gain more reliable insights into the effectiveness of our approaches in generating contrastive and representative snippets, we conducted a study with four human annotators, none of which was an author of this paper (university students with good English

[7] A snippet's embedding is averaged from its sentences' embeddings.

[8] We implemented the topic-independent version of the model.

[9] The effect of adding argumentativeness was also rather low in the original paper [4].

Table 2. Automatic evaluation scores of *Comp-Summarizer* for three different values of the contrastiveness weight λ. The best value in each column is marked in bold.

λ	Contrastiveness	Argumentativeness	Representativeness
0.000	0.059	**0.637**	**0.823**
0.500	0.074	0.632	0.803
0.875	**0.086**	0.624	0.720

Table 3. Manual evaluation results for the three compared approaches on a sample of 50 cases: Contrastiveness, in terms of the percentage of generated snippets that were seen most representative of *their* input argument, and representativeness, in terms of the average and median score. Results highlighted with * and ** are significantly better than Arg-PageRank with confidence level of 95% and 90% respectively.

Approach	Contrastiveness	Representativeness Score	
		Average (± Std.)	Median
Contra-PageRank	***83%**	**3.13 (± 1.15)	3
Comp-Summarizer	*81%	****3.76** (± 1.25)	**4**
Arg-PageRank	65%	3.50 (± 1.35)	**4**

skills). We chose the variants of the two approaches that yielded best contrastiveness above: the third row of Table 1 for *Contra-PageRank*, and the third of Table 2 for *Comp-Summarizer*. As a baseline focusing on representativeness, we also included the *Contra-PageRank* variant in the first row of Table 1, which is similar to the approach of [4], except for computing centrality only based on the argument's sentences rather the whole context. Accordingly, we refer to this baseline as *Arg-PageRank*.

For evaluation, we randomly selected 50 samples of three arguments, $\mathbf{A} = \{A_1, A_2, A_3\}$, and we repeated the following process once for each of the three approaches. For each sample, we first generated the respective snippets, $\mathbf{S} = \{S_1, S_2, S_3\}$. For every snippet $S_i \in \mathbf{S}$, two annotators then manually rated how representative S_i is on a 5-point Likert scale, once for each argument in $\mathbf{A}$. We defined representativeness to our annotators by how much the snippet is covering the main gist, thought, or quintessence of the argument.[10] From this, we infer that S_i is contrastive, if it obtained a higher representativeness score for A_i than for all $A_j \neq A_i$.Before doing so, we made one adjustment, though: Since all three approaches are extractive, the annotators would have easily recognized the argument from which S_i was extracted and, consequently, have scored that argument higher. To avoid this bias, we applied automatic rewriting to all snippets using the PEGASUS transformer [23].

Results The average inter-annotator agreement of the two annotator pairs was substantial, 0.74 in terms Krippendorff's α, suggesting reliable results. Table 3 shows each approach's contrastiveness as the percentage of cases where a generated snippet, S_i, got the highest representativeness score for its input argument, A_i. *Contra-PageRank* generated contrastive snippets most often (83%), while *Arg-PageRank* led to contrastive snippets only in 65% of all cases. In other words, 35% of the snippets of *Arg-PageRank* were mistakenly seen as representative of other arguments by the annotators. This result underlines the importance of fostering snippets to be contrastive. The best trade-off is

[10] For an easy task distribution, we divided the 50 samples into two sets of 25 samples and gave each set to two annotators.

Table 4. Example arguments on *Cloud Seeding* along with the snippet generated for each by our two approaches and the baseline

Topic: Cloud Seeding
Argument-1: Cloud seeding should be used worldwide. This is because, according to both EcoHearth.com and Weather Modifications.org , cloud seeding is safe and virtually harmless to the environment. It can safely cause rain in drought-ravaged areas and keep farms from failing. We should institute cloud seeding in areas where it is necessary. **- ArgPageRank's Snippet:** Cloud seeding should be used in certain areas **- Comp-Summarizer's snippet:** Cloud seeding is safe and harmless to the environment according to both EcoHearth.com and Weather Modifications.org **- Contra-PageRank's snippet:** Cloud seeding should be used in certain areas
Argument-2: Thank you, instigator for providing the resolution. I accept all the proposed terms. Comments I'd like to confirm whether the the embryonic dust cloud theory follows as the popular scientific consensus that a planetary system is created from a nebular of ionised gas where denser and more compact regions form the precursors to a planetary system's celestial bodies. I'd also like to ask who coined "Embryonic Dust Cloud Theory" as I don't want to be unintentionally misrepresenting a scientist's work which may slightly differ from the widely accepted theory. **- ArgPageRank's Snippet** I would like to know who came up with the idea of "Embryonic Dust Cloud Theory" as I don't want to be misrepresenting a scientist's work which may slightly differ from the widely accepted theory **- Comp-Summarizer's snippet** I'd like to confirm that the popular scientific consensus is that a planetary system is created from a nebular of ionised gas where denser and more compact regions form the precursors to a planetary system's heavenly bodies **- Contra-PageRank's snippet** I'd like to confirm that the popular scientific consensus is that a planetary system is created from a nebular of ionised gas where denser and more compact regions form the precursors to a planetary system's heavenly bodies.
Argument-3: Since you have failed to give me an example of an instance where another material has been used instead of silver iodide and was successful, i'll have to ignore that argument. You stated yourself it was lethal. It doesn't matter if the chemical is fairly diluted, it is still dangerous and can cause serious harm to ecosystems. The testing of the soil is faulty and unreliable, so it very possible other studies don't have accurate information. In conclusion, cloud seeding should not be used. This is because it is plainly unnatural and has already wreaked havoc on several ecosystems. Silver Iodide is a harmful chemical that should never be used in the first place. Vote Con! Thanks for the good debate. **- ArgPageRank's Snippet:** There will be no new evidence or arguments to be formed during this round. **- Comp-Summarizer's snippet:** Since you didn't give me an example of an instance where another material was used instead of silver iodide, I'll have to ignore that argument. **- Contra-PageRank's snippet:** Cloud seeding should not be used because the chemical is still dangerous and can cause serious harm to the environment.

achieved by *Comp-Summarizer* which generated the most representative snippets while maintaining contrastiveness almost as often as *Contra-PageRank* (81%).

Example analysis In Table 4, we present three arguments on the topic *Cloud Seeding*, along with the snippets generated by each of the approaches. These snippets are the paraphrased version of the top two sentences selected from the argument. We notice that the baseline ArgPageRank tends to select general sentences like *"Cloud seeding should be used in certain areas."* or *"no new evidence or arguments to be found.."*, while CompSummarizer generated snippets that focus on aspects unique to the argument like *scientific consensus"* and *"harmless to the environment"*.

5. Conclusion

In this work, we argued for the importance of *contrastive* snippets in argument search, that is, snippets that emphasize an argument's unique aspects in the context of others. Building on related work, we have proposed two extractive summarization approaches. Despite room for improvement, our experiments showed their effectiveness and the inhererent trade-off between snippet's contrastiveness and representativeness. While the graph-based summarizer turned out to foster contrastiveness most, the comparative summarizer seems to balance the trade-off better. By focusing on both representativeness and representativeness, we believe that argument snippet generation can produce snippets that help in distinguishing different arguments efficiently.

6. Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021 - 438445824. We would also like to thank the reviewers and the participants who took part in the manual evaluation of our methods.

References

[1] Marcos MC, Gavin F, Arapakis I. Effect of Snippets on User Experience in Web Search. In: Proceedings of the 16th HCI; 2015. p. 47.

[2] Wachsmuth H, Potthast M, Al-Khatib K, Ajjour Y, Puschmann J, Qu J, et al. Building an Argument Search Engine for the Web. In: Proceedings of the 4th Workshop on Argument Mining; 2017. p. 49-59.

[3] Daxenberger J, Schiller B, Stahlhut C, Kaiser E, Gurevych I. Argumentext: argument classification and clustering in a generalized search scenario. Datenbank-Spektrum. 2020;20(2):115-21.

[4] Alshomary M, Düsterhus N, Wachsmuth H. Extractive snippet generation for arguments. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2020. p. 1969-72.

[5] White RW, Ruthven I, Jose JM. Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval; 2002. p. 57-64.

[6] Penin T, Wang H, Tran T, Yu Y. Snippet generation for semantic web search engines. In: Asian Semantic Web Conference. Springer; 2008. p. 493-507.

[7] Chen WF, Syed S, Stein B, Hagen M, Potthast M. Abstractive Snippet Generation. In: Proceedings of The Web Conference 2020; 2020. p. 1309-19.

[8] Wang D, Zhu S, Li T, Gong Y. Comparative document summarization via discriminative sentence selection. ACM Transactions on Knowledge Discovery from Data (TKDD). 2013;7(1):1-18.

[9] Bista U, Mathews A, Shin M, Menon AK, Xie L. Comparative document summarisation via classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. p. 20-8.

[10] Ajjour Y, Wachsmuth H, Kiesel J, Potthast M, Hagen M, Stein B. Data Acquisition for Argument Search: The args. me Corpus. In: Proceedings of the KI; 2019. p. 48-59.

[11] Gretz S, Friedman R, Cohen-Karlik E, Toledo A, Lahav D, Aharonov R, et al. A large-scale dataset for argument quality ranking: Construction and analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. p. 7805-13.

[12] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics. 1987;20:53-65.

[13] Croft B, Metzler D, Strohman T. Search Engines: Information Retrieval in Practice. 1st ed. Addison-Wesley; 2009.

[14] Gholamrezazadeh S, Salehi MA, Gholamzadeh B. A Comprehensive Survey on Text Summarization Systems. In: Proceedings of the 2nd CSA; 2009. p. 1-6.

[15] Li L, Zhou K, Xue GR, Zha H, Yu Y. Enhancing diversity, coverage and balance for summarization through structure learning. In: Proceedings of the 18th international conference on World wide web; 2009. p. 71-80.

[16] Bista U, Mathews AP, Menon AK, Xie L. SupMMD: A Sentence Importance Model for Extractive Summarisation using Maximum Mean Discrepancy. In: Cohn T, He Y, Liu Y, editors. Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020. vol. EMNLP 2020 of Findings of ACL. Association for Computational Linguistics; 2020. p. 4108-22. Available from: https://doi.org/10.18653/v1/2020.findings-emnlp.367.

[17] Maxwell D, Azzopardi L, Moshfeghi Y. The impact of result diversification on search behaviour and performance. Information Retrieval Journal. 2019;22(5):422-46.

[18] Page L, Brin S, Motwani R, Winograd T. The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab; 1999.

[19] Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A. A Kernel Two-Sample Test. Journal of Machine Learning Research. 2012;13(1):723-73.

[20] Erkan G, Radev DR. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research. 2004;22:457-79.

[21] Dorsch J, Wachsmuth H. Semi-Supervised Cleansing of Web Argument Corpora. In: Proceedings of the 7th Workshop on Argument Mining. Online: Association for Computational Linguistics; 2020. p. 19-29. Available from: https://aclanthology.org/2020.argmining-1.3.

[22] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics; 2019. Available from: https://arxiv.org/abs/1908.10084.

[23] Zhang J, Zhao Y, Saleh M, Liu P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning. PMLR; 2020. p. 11328-39.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220139

Explainable Logic-Based Argumentation

Ofer ARIELI [a,1], AnneMarie BORG [b], Matthis HESSE [c], Christian STRASSER [c]
[a] *School of Computer Science, Tel-Aviv Academic College, Israel*
[b] *Dept of Information and Computing Sciences, Utrecht University, the Netherlands*
[c] *Institute for Philosophy II, Ruhr University Bochum, Germany*

Abstract. Explainable artificial intelligence (XAI) has gained increasing interest in recent years in the argumentation community. In this paper we consider this topic in the context of logic-based argumentation, showing that the latter is a particularly promising paradigm for facilitating explainable AI. In particular, we provide two representations of abductive reasoning by sequent-based argumentation frameworks and show that such frameworks successfully cope with related challenges, such as the handling of synonyms, justifications, and logical equivalences.

Keywords. Explainable AI, sequent-based argumentation, abductive logics

1. Introduction

EXplainable Artificial Intelligence (XAI) is an AI research area aimed at providing explanation to inferences and decisions made by intelligent systems [1]. *Argumentative* XAI is a fast growing area that studies XAI by means of computational argumentation (see, e.g., the recent survey papers in [2,3]).

Computational argumentation is based here on *argumentation frameworks* (AFs) [4], which are pairs of set of arguments and attack relation between the arguments, where conclusions are derived by determining subsets of arguments that can collectively be accepted in the framework. In *logic-based argumentation* [5,6] the arguments are instantiated by applying an underlying logic. Studying argumentative XAI from a logic-based perspective has several advantages. Beyond the fact that explanations in this context can be justified in a logical and rational manner, a logic-based setting is especially suitable for modeling *abductive reasoning* [7], which can be viewed as inference to the best explanation. Thus, it allows also for 'backwards reasoning', seeking for explanations for drawing conclusions from a set of observations.

In this work, we show that logic-based argumentation (and in particular sequent-based argumentation [6,8]) provides robust mechanisms for abductive reasoning in argumentative settings. In particular, we consider two ways in which abductive reasoning can be modeled by sequent-based argumentation. The first one is based on the derived argumentative conclusions, where explanations can be determined in terms of entailment relations. In the other approach, abductive reasoning is represented *within* the frameworks, where explanations are incorporated in the arguments and in the attack relations. The two approaches are then related and are used for providing information on how explanations are justified relative to the assumptions.

[1] Supported by the Israel Science Foundation (grant 550/19).

2. Preliminaries; Sequent-Based Argumentation

In this paper, we denote by $\mathfrak{L}$ a propositional language. Atomic formulas in $\mathfrak{L}$ are denoted by p,q,r, formulas are denoted by $\phi,\psi,\delta,\gamma,\varepsilon$, sets of formulas are denoted by X, S, E, and finite sets of formulas are denoted by Γ,Δ,Π,Θ, all of which can be primed or indexed. The set of atomic formulas appearing in the formulas of S is denoted $\mathsf{Atoms}(\mathsf{S})$. The set of the (well-formed) formulas of $\mathfrak{L}$ is denoted $\mathsf{WFF}(\mathfrak{L})$, the power set of $\mathsf{WFF}(\mathfrak{L})$ is denoted $\wp(\mathsf{WFF}(\mathfrak{L}))$. Sequent-based argumentation is then described as follows:

- **The base logic** is an arbitrary propositional logic, namely a pair $\mathsf{L} = \langle \mathfrak{L}, \vdash \rangle$, consisting of a language $\mathfrak{L}$ and a consequence relation $\vdash$ on $\wp(\mathsf{WFF}(\mathfrak{L})) \times \mathsf{WFF}(\mathfrak{L})$. $\vdash$ is assumed to satisfy: *reflexivity* ($\mathsf{S} \vdash \phi$ if $\phi \in \mathsf{S}$), *monotonicity* (if $\mathsf{S}' \vdash \phi$ and $\mathsf{S}' \subseteq \mathsf{S}$, then $\mathsf{S} \vdash \phi$), and *transitivity* (if $\mathsf{S} \vdash \phi$ and $\mathsf{S}', \phi \vdash \psi$ then $\mathsf{S}, \mathsf{S}' \vdash \psi$).

Let $\mathsf{L} = \langle \mathfrak{L}, \vdash \rangle$ be a logic and let S be a set of $\mathfrak{L}$-formulas. The $\vdash$*-closure of* S is the set $\mathsf{CN}_\mathsf{L}(\mathsf{S}) = \{\phi \mid \mathsf{S} \vdash \phi\}$. We say that S is $\vdash$*-consistent*, if there are no formulas $\phi_1, \ldots, \phi_n \in \mathsf{S}$ for which $\vdash \neg(\phi_1 \wedge \cdots \wedge \phi_n)$.

- **The language** $\mathfrak{L}$ contains at least a $\vdash$-negation operator $\neg$, satisfying $p \nvdash \neg p$ and $\neg p \nvdash p$ (for atomic p), and a $\vdash$-conjunction operator $\wedge$, for which $\mathsf{S} \vdash \psi \wedge \phi$ iff $\mathsf{S} \vdash \psi$ and $\mathsf{S} \vdash \phi$. We denote by $\bigwedge\Gamma$ the conjunction of all the formulas in Γ. We shall sometimes assume the availability of a deductive implication $\to$, satisfying $\mathsf{S}, \psi \vdash \phi$ iff $\mathsf{S} \vdash \psi \to \phi$.

- **Arguments** based on a logic $\mathsf{L} = \langle \mathfrak{L}, \vdash \rangle$ are single-conclusioned L-*sequents* [9], namely: expressions of the form $\Gamma \Rightarrow \psi$, where $\Rightarrow$ is a symbol that does not appear in $\mathfrak{L}$, and such that $\Gamma \vdash \psi$. Γ is called the argument's support (also denoted $\mathsf{Supp}(\Gamma \Rightarrow \psi)$) and ψ is the argument's conclusion (denoted $\mathsf{Conc}(\Gamma \Rightarrow \psi)$). Given a set S of $\mathfrak{L}$-formulas (premises), an S-based argument is an L-argument $\Gamma \Rightarrow \psi$, where $\Gamma \subseteq \mathsf{S}$. We denote by $\mathrm{Arg}_\mathsf{L}(\mathsf{S})$ the set of all the L-arguments that are based on S.

We distinguish between two types of non-intersecting premises: a $\vdash$-consistent set X of strict (i.e., non-attacked) premises, and a set S of defeasible premises. Their non-defeasible character will give them a special status when we define argumentative attacks below. We write $\mathrm{Arg}^{\mathsf{X}}_\mathsf{L}(\mathsf{S})$ for the set $\mathrm{Arg}_\mathsf{L}(\mathsf{X} \cup \mathsf{S})$. In particular, $\mathrm{Arg}^{\emptyset}_\mathsf{L}(\mathsf{S}) = \mathrm{Arg}_\mathsf{L}(\mathsf{S})$.

- **Attack rules** are sequent-based inference rules for representing attacks between sequents. Such rules consist of an attacking argument (the first condition of the rule), an attacked argument (the last condition of the rule), conditions for the attack (the other conditions of the rule) and a conclusion (the eliminated attacked sequent). The outcome of an application of such a rule is that the attacked sequent is 'eliminated' (or 'invalidated'; see below the exact meaning of this). The elimination of $\Gamma \Rightarrow \phi$ is denoted by $\Gamma \not\Rightarrow \phi$.

Given a set X of strict (non-attacked) formulas, some common attack rules are:

- Defeat: $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg\bigwedge\Gamma_2 \quad \Gamma_2, \Gamma_2' \Rightarrow \psi_2}{\Gamma_2, \Gamma_2' \not\Rightarrow \psi_2}$ $(\Gamma_2 \neq \emptyset,\ \Gamma_2 \cap \mathsf{X} = \emptyset)$

- Direct Defeat: $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg\gamma \quad \Gamma_2, \gamma \Rightarrow \psi_2}{\Gamma_2, \gamma \not\Rightarrow \psi_2}$ $(\gamma \notin \mathsf{X})$

- Undercut: $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg\bigwedge\Gamma_2 \quad \neg\bigwedge\Gamma_2 \Rightarrow \psi_1 \quad \Gamma_2, \Gamma_2' \Rightarrow \psi_2}{\Gamma_2, \Gamma_2' \not\Rightarrow \psi_2}$ $(\Gamma_2 \neq \emptyset,\ \Gamma_2 \cap \mathsf{X} = \emptyset)$

- Direct Undercut: $\dfrac{\Gamma_1 \Rightarrow \psi_1 \quad \psi_1 \Rightarrow \neg\gamma \quad \neg\gamma \Rightarrow \psi_1 \quad \Gamma_2, \gamma \Rightarrow \psi_2}{\Gamma_2, \gamma \not\Rightarrow \psi_2}$ $(\gamma \notin \mathsf{X})$
- Consistency Undercut: $\dfrac{\Gamma_1 \Rightarrow \neg \bigwedge \Gamma_2 \quad \Gamma_2, \Gamma_2' \Rightarrow \psi}{\Gamma_2, \Gamma_2' \not\Rightarrow \psi}$ $(\Gamma_2 \neq \emptyset, \Gamma_2 \cap \mathsf{X} = \emptyset, \Gamma_1 \subseteq \mathsf{X})$

For instance, in the particular case where $\Gamma_1 = \emptyset$, consistency undercut indicates that an argument with an inconsistent support is eliminated.

• **A (sequent-based) argumentation framework** (AF), based on the logic L and the attack rules in AR, for a set of defeasible premises S and a $\vdash$-consistent set of strict premises X, is a pair $\mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{AR}}(\mathsf{S}) = \langle \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S}), \mathsf{A} \rangle$ where $\mathsf{A} \subseteq \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S}) \times \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$ and $(a_1, a_2) \in \mathsf{A}$ iff there is a rule $\mathsf{R}_{\mathsf{X}} \in \mathsf{AR}$, such that a_1 R_{X}-attacks a_2. In what follows we shall use AR and A interchangeably, denoting both of them by A.

• **Semantics** of sequent-based frameworks are defined as usual by Dung-style extensions [4]: Let $\mathbb{AF} = \mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S}) = \langle \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S}), \mathsf{A} \rangle$ be an argumentation framework and let $\mathbb{E} \subseteq \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$ be a set of arguments. It is said that: $\mathbb{E}$ *attacks* a if there is an $a' \in \mathbb{E}$ such that $(a', a) \in \mathsf{A}$, $\mathbb{E}$ *defends* a if $\mathbb{E}$ attacks every attacker of a, and $\mathbb{E}$ is *conflict-free* (cf) if for no $a_1, a_2 \in \mathbb{E}$ it holds that $(a_1, a_2) \in \mathsf{A}$. We say that $\mathbb{E}$ is *admissible* if it is conflict-free and defends all of its elements. A *complete (*cmp*) extension* of $\mathbb{AF}$ is an admissible set that contains all the arguments that it defends. By this, various argumentative semantics may be defined. For instance, the *grounded (*grd*) extension* of $\mathbb{AF}$ is the $\subseteq$-minimal complete extension of $\mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$, a *preferred (*$\mathsf{prf}$*) extension* of $\mathbb{AF}$ is a $\subseteq$-maximal complete extension of $\mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$, and a *stable (*$\mathsf{stb}$*) extension* of $\mathbb{AF}$ is a conflict-free set in $\mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$ that attacks every argument not in it.[2] We denote by $\mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF})$ the set of all the extensions of $\mathbb{AF}$ of type sem.

• **Entailments** induced from an argumentation framework $\mathbb{AF} = \mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S}) = \langle \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S}), \mathsf{A} \rangle$ are based on the extensions derived from $\mathbb{AF}$ under a semantics sem:

- *Skeptical entailment:* $\mathsf{S} \mid\!\sim^{\cap,\mathsf{sem}}_{\mathsf{L},\mathsf{A},\mathsf{X}} \phi$ if there is an argument $a \in \bigcap \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF})$ such that $\mathsf{Conc}(a) = \phi$.
- *Weakly skeptical entailment:* $\mathsf{S} \mid\!\sim^{\Cap,\mathsf{sem}}_{\mathsf{L},\mathsf{A},\mathsf{X}} \phi$ if for every extension $\mathbb{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF})$ there is an argument $a \in \mathbb{E}$ such that $\mathsf{Conc}(a) = \phi$.
- *Credulous entailment:* $\mathsf{S} \mid\!\sim^{\cup,\mathsf{sem}}_{\mathsf{L},\mathsf{A},\mathsf{X}} \phi$ iff there is an argument $a \in \bigcup \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF})$ such that $\mathsf{Conc}(a) = \phi$.

Example 1. Consider an AF, based on classical logic CL and the following set of defeasible assumptions:

$$\mathsf{S} = \left\{ \begin{array}{l} \mathtt{clear\_skies},\ \mathtt{rainy},\ \mathtt{clear\_skies} \to \neg\mathtt{rainy},\ \mathtt{rainy} \to \neg\mathtt{sprinklers}, \\ \mathtt{rainy} \to \mathtt{wet\_grass},\ \mathtt{sprinklers} \to \mathtt{wet\_grass} \end{array} \right\}$$

Suppose further that there are no strict assumptions ($\mathsf{X} = \emptyset$) and that the only attack rule is undercut (Ucut). Then, for instance, the arguments

$$\begin{array}{l} a_1:\ \mathtt{clear\_skies}, \mathtt{clear\_skies} \to \neg\mathtt{rainy} \Rightarrow \neg\mathtt{rainy}, \\ a_2:\ \mathtt{rainy}, \mathtt{clear\_skies} \to \neg\mathtt{rainy} \Rightarrow \neg\mathtt{clear\_skies} \end{array}$$

[2] Further extensions and the relations among them are discussed e.g. in [10].

Ucut-attack each other. In this case there are two stable/preferred extensions $\mathbb{E}_1$ and $\mathbb{E}_2$, where $a_1 \in \mathbb{E}_1$ and $a_2 \in \mathbb{E}_2$. It follows, for instance, that with respect to these semantics, $\texttt{wet\_grass}$ credulously follows from the framework (since, e.g. $\texttt{rainy}, \texttt{rainy} \to \texttt{wet\_grass} \Rightarrow \texttt{wet\_grass}$ is in $\mathbb{E}_2$), but it does follow skeptically (since there is no argument in $\mathbb{E}_1$ whose conclusion is $\texttt{wet\_grass}$).

3. Abductive Reasoning in Sequent-Based Frameworks

Abductive reasoning is a common method of providing explanations in logic-based contexts. Sequent-based formalisms are particularly adequate for this, as instead of the usual understanding of a sequent $\Gamma, \Delta \Rightarrow \phi$ by 'ϕ is a conclusion of $\Gamma \cup \Delta$', one may intuitively read it as 'Δ is a (prima facia) explanation of ϕ in the presence of Γ'. This kind of 'backward reasoning' is also our starting point for showing the usefulness of sequent-based frameworks for abductive reasoning. We then proceed in two directions, external and internal ones, for defining abductive reasoning in sequent-based argumentation.

3.1. Explanations: External View

We start with an 'external' approach, which is based on argumentative entailment relations. Let $\mathsf{L} = \langle \mathfrak{L}, \vdash \rangle$ be a logic and $\mid\!\sim$ a non-monotonic entailment induced by it.[3] Given sets of strict (X) and defeasible (S) assumptions, an *explanation* E of an *explanandum* ϕ with respect to $\mid\!\sim$, is a finite set that satisfies at least the following two properties:

***Sufficiency* (w.r.t. $\mid\!\sim$):** $\mathsf{X}, \mathsf{S}, \mathsf{E} \mid\!\sim \phi$
***Consistency* (w.r.t. $\vdash$):** $\mathsf{X} \nvdash \neg \bigwedge \mathsf{E}$

Thus, the set of explanations should be $\vdash$-consistent with the strict assumptions, and together with the strict and defeasible assumptions they are sufficient for $\mid\!\sim$-inferring the explanandum ϕ. We call these two conditions the *basic explanation properties*.

The basic explanation properties per-se may sometimes be too weak, and so they are usually accompanied with further conditions. The following ones are inspired by [11]:

***Non-vacuity* (w.r.t. $\vdash$):** $\mathsf{E} \nvdash \phi$
***Minimality* (w.r.t. $\mid\!\sim$):** $\mathsf{S}, \mathsf{E}' \not\mid\!\sim \phi$ for every E' for which $\mathsf{E}, \mathsf{X} \vdash \bigwedge \mathsf{E}'$ and $\mathsf{E}', \mathsf{X} \nvdash \bigwedge \mathsf{E}$.

Non-vacuity prevents self-explanations, and minimality assures the conciseness of the explanations. In order to make sure that the explanation is indeed necessary (i.e., the explanandum cannot be inferred from the assumptions alone), the property of non-idleness ($\mathsf{X}, \mathsf{S} \not\mid\!\sim \phi$) or strict non-idleness ($\mathsf{X} \nvdash \phi$) may be required. Here it will be convenient to use the following argumentative variations of this property:

***Non-idleness* (w.r.t. sem):** there is no $a \in \bigcup \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S}))$ s.t. $\mathsf{Conc}(a) = \phi$.
***Strict non-idleness* (w.r.t. sem):** there is no $a \in \bigcup \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\emptyset))$ s.t. $\mathsf{Conc}(a) = \phi$.

By the above principles, external argumentative explanations are defined as follows:

[3] In our case, $\mid\!\sim$ is the entailment induced from a framework that is based on L.

Definition 1. Given a framework $\mathbb{AF} = \mathbb{AF}^{\mathsf{X}}_{\mathsf{L,A}}(\mathsf{S})$ based on a logic $\mathsf{L} = \langle \mathfrak{L}, \vdash \rangle$, a finite set E of $\mathfrak{L}$-formulas is called:

- *external skeptical* sem-*explanation* of ϕ if it satisfies $\mathrel{|\!\sim}^{\cap,\mathsf{sem}}_{\mathsf{L,A,X}}$-sufficiency ($\mathsf{X},\mathsf{S},\mathsf{E} \mathrel{|\!\sim}^{\cap,\mathsf{sem}}_{\mathsf{L,A,X}} \phi$), $\vdash$-consistency ($\mathsf{X} \nvdash \neg \bigwedge \mathsf{E}$), and holds in every sem-extension: for every $\mathbb{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF}^{\mathsf{X}}_{\mathsf{L,A}}(\mathsf{S} \cup \mathsf{E}))$ there is $a \in \mathbb{E}$, such that $\mathsf{Conc}(a) = \bigwedge \mathsf{E}$.
- *external weakly-skeptical* sem-*explanation* of ϕ if it satisfies $\mathrel{|\!\sim}^{\Cap,\mathsf{sem}}_{\mathsf{L,A,X}}$-sufficiency ($\mathsf{X},\mathsf{S},\mathsf{E} \mathrel{|\!\sim}^{\Cap,\mathsf{sem}}_{\mathsf{L,A,X}} \phi$), $\vdash$-consistency ($\mathsf{X} \nvdash \neg \bigwedge \mathsf{E}$), and holds in every sem-extension: for every $\mathbb{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF}^{\mathsf{X}}_{\mathsf{L,A}}(\mathsf{S} \cup \mathsf{E}))$ there is $a \in \mathbb{E}$, such that $\mathsf{Conc}(a) = \bigwedge \mathsf{E}$.
- *external credulous* sem-*explanation* of ϕ if it satisfies $\mathrel{|\!\sim}^{\cup,\mathsf{sem}}_{\mathsf{L,A,X}}$-sufficiency ($\mathsf{X},\mathsf{S},\mathsf{E} \mathrel{|\!\sim}^{\cup,\mathsf{sem}}_{\mathsf{L,A,X}} \phi$), $\vdash$-consistency ($\mathsf{X} \nvdash \neg \bigwedge \mathsf{E}$), and holds in some sem-extension: there is some $\mathbb{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF}^{\mathsf{X}}_{\mathsf{L,A}}(\mathsf{S} \cup \mathsf{E}))$ and $a \in \mathbb{E}$, such that $\mathsf{Conc}(a) = \bigwedge \mathsf{E}$.

Example 2. Consider again the framework in Example 1. Note that $\mathsf{E} = \{\mathtt{sprinklers}\}$ is a (stable and preferred) credulous explanation for `wet_grass`. Indeed, using the notations of Example 1, the framework that is based on $\mathsf{S} \cup \mathsf{E}$ has two stable/preferred extensions: $\mathbb{E}'_1$ and $\mathbb{E}'_2 = \mathbb{E}_2$ (see Figure 1). In $\mathbb{E}'_1$ the grass is wet since the sprinklers are activated, and in $\mathbb{E}'_2$ the grass is wet since it rains.

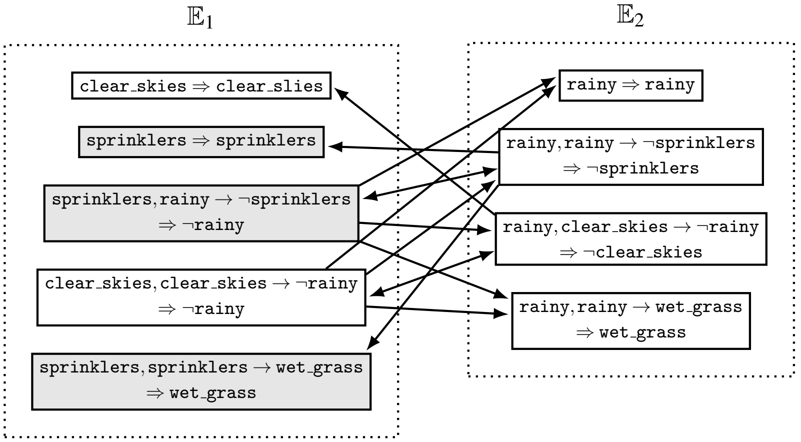

$\mathbb{E}_1$
$\mathbb{E}_2$
clear_skies ⇒ clear_slies
sprinklers ⇒ sprinklers
sprinklers, rainy → ¬sprinklers ⇒ ¬rainy
clear_skies, clear_skies → ¬rainy ⇒ ¬rainy
sprinklers, sprinklers → wet_grass ⇒ wet_grass
rainy ⇒ rainy
rainy, rainy → ¬sprinklers ⇒ ¬sprinklers
rainy, clear_skies → ¬rainy ⇒ ¬clear_skies
rainy, rainy → wet_grass ⇒ wet_grass

Figure 1. Part of the AF of Example 2. The arguments with dark background are added by the explanation.

3.2. Explanations: Internal View

We now turn to the 'internal' approach, where abductive explanations are handled by ingredients of the framework. We do so by considering another type of sequents, called 'abductive sequents'. These are expressions of the form $\phi \Leftarrow \Gamma, [\varepsilon]$,[4] and it intuitively means that 'ϕ may be inferred from Γ under the assumption that ε holds'. Note that while $\Gamma \subseteq \mathsf{S} \cup \mathsf{X}$, ε may not be an assumption, but rather a hypothetical explanation of the conclusion.

[4] Note the reverse direction of the sequent sign, to emphasize the backward inference in this case.

Abductive sequents may be produced by the following rule that roughly models the usual idea of abductive inference as backwards reasoning:

$$\frac{\varepsilon, \Gamma \Rightarrow \phi}{\phi \Leftarrow \Gamma, [\varepsilon]} \ \text{(Abduction)}$$

In our running example, this rule will allow us to produce abductive sequents such as

$$\texttt{wet\_grass} \Leftarrow [\texttt{sprinklers}], \texttt{sprinklers} \rightarrow \texttt{wet\_grass}$$

that provides an alternative explanation to the wetness of the grass (i.e., `sprinklers`, in addition to `rainy`), or

$$\neg\texttt{rainy} \Leftarrow [\texttt{sprinklers}], \texttt{rainy} \rightarrow \neg\texttt{sprinklers}$$

that provides another possible evidence for refuting the defeasible assumption that it is rainy (i.e., `sprinklers`, in addition to the assumption that the sky is clear).

Since abductive reasoning is a form of non-monotonic reasoning, which in logic-based argumentation is modeled with the attack relations, we need a way to attack abductive sequents. To this end, we consider rules similar to those from Section 2, e.g.:

$$\frac{\Gamma_1 \Rightarrow \phi_1 \quad \phi_1 \Rightarrow \neg\gamma \quad \phi_2 \Leftarrow [\varepsilon], \Gamma_2}{\phi_2 \not\Leftarrow [\varepsilon], \Gamma_2} \ \gamma \in (\Gamma_2 \cup \{\varepsilon\}) \setminus \mathsf{X} \ \text{(Abductive Direct Defeat)}$$

which models an attack on a subset of the assumptions and a hypothetical explanation of an abductive sequent. Note that this attack rule assures, in particular, the consistency of explanations with the strict assumptions, thus it renders the following rule admissible:

$$\frac{\Gamma_1 \Rightarrow \neg\varepsilon \quad \phi \Leftarrow [\varepsilon], \Gamma_2}{\phi \not\Leftarrow [\varepsilon], \Gamma_2} \ \Gamma_1 \subseteq \mathsf{X} \ \text{(Consistency)}$$

Abductive reasoning has to fulfill certain requirements to ensure proper behavior also in the internal view. This time, attack rules may be introduced for obtaining counterparts of the properties discussed in Section 3.1 for the external view. Note that, since abductive sequents are now derived according to the underlying sequent calculus and the abduction rule introduced above, the sufficiency property is automatically satisfied. Attack rules for the other properties are given next.

Non-vacuity Rules for preventing self-explanations:

$$\frac{\vdash \varepsilon \rightarrow \phi \quad \phi \Leftarrow [\varepsilon]}{\phi \not\Leftarrow [\varepsilon]} \ \text{(Non Vacuity)}$$

Thus, in our running example, $\texttt{wet\_grass} \Leftarrow [\texttt{wet\_grass}]$ is excluded.

Minimality Rules for assuring that explanations will be as general as possible.

$$\frac{\phi \Leftarrow [\varepsilon_1], \Gamma_1 \quad \vdash \varepsilon_2 \rightarrow \varepsilon_1 \quad \nvdash \varepsilon_1 \rightarrow \varepsilon_2 \quad \phi \Leftarrow [\varepsilon_2], \Gamma_2}{\phi \not\Leftarrow [\varepsilon_2], \Gamma_2} \ \text{(Minimality)}$$

This rule assures that in our example $\texttt{sprinklers} \wedge \texttt{irrelevant\_fact}$ should not explain $\texttt{wet\_grass}$, since $\texttt{sprinklers}$ is a more general and so more relevant explanation.

Non-Idleness The [strict] assumptions should not already explain the explanandum.

$$\frac{\Gamma_1 \Rightarrow \phi \qquad \phi \Leftarrow [\varepsilon], \Gamma_2}{\phi \not\Leftarrow [\varepsilon], \Gamma_2} \text{ (Defeasible Non Idleness)}$$

$$\frac{\Gamma_1 \Rightarrow \phi \qquad \phi \Leftarrow [\varepsilon], \Gamma_2}{\phi \not\Leftarrow [\varepsilon], \Gamma_2} \; \Gamma_1 \subseteq \mathsf{X} \text{ (Strict Non Idleness)}$$

Note that defeasible non-idleness excludes the explanation $\texttt{sprinklers}$ for $\texttt{wet\_grass}$, since the latter is already inferred from the defeasible assumptions (assuming that it is rainy), while strict non-idleness will allow this alternative explanation (since $\texttt{wet\_grass}$ cannot be inferred from the strict assumptions).

The next step is to adapt sequent-based argumentation frameworks to an abductive setting, using abductive sequents, the new inference rule, and additional attack rules. Given a sequent-based framework $\mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S})$, an *abductive sequent-based framework* $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$ is constructed by adding to the arguments in $\mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$ also abductive arguments, produced by Abduction, and where $\mathsf{A}^\star$ is obtained by adding to the attack rules in A also (some of) the rules for maintaining explanations that are described above. Explanations according to the internal view are then defined as follows:

Definition 2. Given an abductive sequent-based framework $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$ as described above, a finite set E of $\mathfrak{L}$-formulas is called:

- *internal skeptical* sem-*explanation* of ϕ, if there is $\Gamma \subseteq \mathsf{S}$ such that the abductive argument $\phi \Leftarrow [\bigwedge \mathsf{E}], \Gamma$ is in every sem-extension of $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$.
- *internal weakly-skeptical* sem-*explanation* of ϕ, if in every sem-extension of $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$ there is an abductive argument $\phi \Leftarrow [\bigwedge \mathsf{E}], \Gamma$ for some $\Gamma \subseteq \mathsf{S}$
- *internal credulous* sem-*explanation* of ϕ, if there is $\Gamma \subseteq \mathsf{S}$ such that the abductive argument $\phi \Leftarrow [\bigwedge \mathsf{E}], \Gamma$ is in some sem-extension of $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$.

Example 3. As noted above, $\texttt{wet\_grass} \Leftarrow [\texttt{sprinklers}], \texttt{sprinklers} \to \texttt{wet\_grass}$ is producible by the Abduction rule from the sequent-based framework in Example 1, and belongs to a stable/preferred extension of the corresponding abductive sequent-based framework. Therefore, $\texttt{sprinklers}$ credulously stb/prf-explains $\texttt{wet\_grass}$ also according to Definition 2.

3.3. Explanations: Relating the Two Views

Next, we relate the two approaches for producing argumentative explanations by abductive reasoning in sequent-based frameworks. In what follows we restrict ourselves to singleton explanations in the assumptions language.[5] We consider $\{\text{ConUcut}\} \subset \mathsf{A} \subseteq \{\text{ConUcut}, \text{DirectDefeat}, \text{DirectUndercut}\}$. The main results are the following:

[5] Thus, using the notations of the previous sections, $\mathsf{E} = \{\varepsilon\}$, where $\mathsf{Atoms}(\varepsilon) \subseteq \mathsf{Atoms}(\mathsf{S} \cup \mathsf{X})$.

Theorem 1. *Let* $\mathbb{AF} = \mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S})$ *where* $\mathsf{L} = \mathsf{CL}$*,* A *is as specified above, and* $\mathbb{AAF} = \mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$ *where* $\mathsf{A}^\star = \mathsf{A} \cup \{\text{Abductive Direct Defeat}\}$*. For* $\mathsf{sem} \in \{\mathsf{stb},\mathsf{prf}\}$*,* E *is an external weakly skeptical (resp. skeptical)* sem*-explanation of* ϕ *w.r.t.* $\mathbb{AF}$ *iff* E *is an internal weakly skeptical (resp. skeptical)* sem*-explanation of* ϕ *w.r.t.* $\mathbb{AAF}$*. Moreover,* E *satisfies non-vacuity and/or strict non-idleness iff the non-vacuity and/or the strict non-idleness attack rule is added to* $\mathsf{A}^\star$*.*

Theorem 2. *Let* $\mathbb{AF} = \mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S})$ *where* $\mathsf{L} = \mathsf{CL}$*,* A *is as specified above, and* $\mathbb{AAF} = \mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$ *where* $\mathsf{A}^\star = \mathsf{A} \cup \{\text{Abductive Direct Defeat}\}$*. Then* E *is an external weakly skeptical (resp. skeptical)* grd*-explanation of* ϕ *w.r.t.* $\mathbb{AF}$ *iff* E *is an internal weakly skeptical (resp. skeptical)* grd*-explanation of* ϕ *w.r.t.* $\mathbb{AAF}$*. Moreover,* E *satisfies non-vacuity and/or strict non-idleness iff the non-vacuity and/or the strict non-idleness attack rule is added to* $\mathsf{A}^\star$*.*

The proofs of Theorems 1 and 2 are based on the correspondence to reasoning with maximally consistent sets of assumptions, shown in [12]. Next, we sketch the proof of Theorem 1 for $\mathsf{A}^\star = \mathsf{A} \cup \{\text{Abductive Direct Defeat}\}$ and the weakly skeptical version (the proof for the skeptical version and the proof of Theorem 2 are similar). In the proof, $\mathsf{MCS}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$ is the set of the maximally $\vdash$-consistent subsets of S, which are also $\vdash$-consistent with X.

Proof outline of Theorem 1. [$\Rightarrow$] Suppose that $\mathsf{E} = \{\varepsilon\}$ is an external weakly skeptical sem-explanation of ϕ w.r.t. $\mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S})$, where $\mathsf{sem} \in \{\mathsf{stb},\mathsf{prf}\}$. In particular, $\mathsf{S},\varepsilon \mid\!\sim^{\Cap,\mathsf{sem}}_{\mathsf{L},\mathsf{A},\mathsf{X}} \phi$, and for every $\mathbb{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S} \cup \{\varepsilon\})$ there is $a \in \mathbb{E}$, such that $\mathsf{Conc}(a) = \varepsilon$. By [12, Theorem 1], ($\dagger$) for all $\Delta \in \mathsf{MCS}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S} \cup \{\varepsilon\})$ we have that $\mathsf{X},\Delta \vdash \phi$ and $\mathsf{X},\Delta \vdash \varepsilon$.

Let now $\mathbb{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S}))$. Then $\mathbb{E} \cap \mathsf{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S}) \in \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S}))$, and so, by [12, Theorem 1] again, $\mathbb{E} \cap \mathsf{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S}) = \mathsf{Arg}^{\mathsf{X}}_{\mathsf{L}}(\Delta)$ for some $\Delta \in \mathsf{MCS}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$. By ($\dagger$), for all $\Omega \in \mathsf{MCS}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$, $\Omega,\mathsf{X} \nvdash \neg\varepsilon$. So, $\mathsf{X},\Delta \nvdash \neg\varepsilon$. Thus $\Delta \cup \{\varepsilon\} \in \mathsf{MCS}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S} \cup \{\varepsilon\})$. By ($\dagger$), there is some finite $\Gamma \subseteq \Delta \setminus \{\varepsilon\}$, for which $\mathsf{X},\Gamma,\varepsilon \vdash \phi$. It follows that $\phi \Leftarrow [\varepsilon],\Gamma$ is an abductive argument in $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$.

Note that $\mathsf{X},\Delta \not\vdash \neg\gamma$ for all $\gamma \in (\Gamma \cup \{\varepsilon\}) \setminus \mathsf{X}$, otherwise $\mathsf{X},\Delta \vdash \neg\varepsilon$, in a contradiction to ($\dagger$) and the consistency of $\Gamma \subseteq \Delta$. Thus $\phi \Leftarrow [\varepsilon],\Gamma$ is not abductively attacked by any element of $\mathbb{E}$, and so $\phi \Leftarrow [\varepsilon],\Gamma \in \mathbb{E}$. It follows that ε is an internal weakly skeptical sem-explanation of ϕ w.r.t. $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$.

[$\Leftarrow$] Suppose that $\mathsf{E} = \{\varepsilon\}$ is an internal weakly skeptical sem-explanation of ϕ w.r.t. $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$. Let $\mathbb{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathsf{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S} \cup \{\varepsilon\}))$. By [12, Theorem 1], $\mathbb{E} = \mathsf{Arg}^{\mathsf{X}}_{\mathsf{L}}(\Delta)$ for some $\Delta \in \mathsf{MCS}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S} \cup \{\varepsilon\})$. Then $\mathsf{X},\Delta \nvdash \neg\varepsilon$, and so $\Delta' = \Delta \cap \mathsf{S} \in \mathsf{MCS}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$. Let $\mathbb{E}'$ be the set of all the $(\mathsf{X} \cup \Delta)$-based sequents and $(\mathsf{X} \cup \Delta)$-based abducitive sequents. It can be shown that $\mathbb{E}' \in \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S}))$. Thus, there is an $\phi \Leftarrow [\varepsilon],\Gamma \in \mathbb{E}$, and $\Gamma,\varepsilon \vdash \phi$ (for $\Gamma \subseteq \Delta \cup \mathsf{X}$). Thus, $\mathsf{X},\Delta \vdash \phi$ and $\mathsf{X},\Delta \vdash \varepsilon$. It follows that ε is an external weakly skeptical sem-explanation of ϕ w.r.t. $\mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S})$. □

We note that not in all cases the external and internal explanations coincide, even when $\mathsf{L} = \mathsf{CL}$ and $\mathsf{A} = \{\text{Direct Defeat}, \text{ConUcut}\}$. The next example illustrates this:

Example 4. Let $\mathsf{L} = \mathsf{CL}$, $\mathsf{A} = \{\text{Direct Defeat}, \text{ConUcut}\}$, $\mathsf{S} = \{p, \neg p \wedge q\}$ and $\mathsf{X} = \{q \wedge r \to s\}$. Then:

1. $q \wedge r$ is an external weakly-skeptical $\mathtt{stb}$-explanation of s, since the corresponding sequent-based framework has two stable extensions: $\mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\{p, q \wedge r\})$ and $\mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\{\neg p \wedge q, q \wedge r\})$, both of which contain arguments for $q \wedge r$ and for $q \wedge r \to s$. Note that this explanation satisfies Non-vacuity (s does not follow from $q \wedge r$).
2. $q \wedge r$ is an internal weakly-skeptical $\mathtt{stb}$-explanation of s, since the corresponding abductive sequent framework also has two stable extensions, both with the abducible sequent $s \Leftarrow [q \wedge r], q \wedge r \to s$.This holds also when the non-vacuity and/or strict non-idleness attack rules are part of the framework.

This is in accordance with Theorem 1. Suppose now that minimality is imposed. Then:

1. $q \wedge r$ remains an external weakly-skeptical $\mathtt{stb}$-explanation of s, since it satisfies the minimality condition.
2. $q \wedge r$ is *no longer* an internal weakly-skeptical $\mathtt{stb}$-explanation of s, since one extension also contains a minimality attacker of $s \Leftarrow [q \wedge r], q \wedge r \to s$, namely: $s \Leftarrow [r], \neg p \wedge q, q \wedge r \to s$.

4. Some Further Considerations

In this section we briefly comment on some other aspects of argumentation explanation.

4.1. Handling of Synonyms and Antonyms

Synonyms and antonyms may be handled by the strict assumptions, as they should not be revised. This may be done either to clarify the meaning of some terminology used by defeasible formulas, or for extending the vocabulary describing the domain of discourse. For instance, suppose that in our running example we add the strict assumption $\mathsf{X} = \{\mathtt{blue\_skies} \leftrightarrow \mathtt{clear\_skies}\}$. Then, since

$$\mathtt{blue\_skies}, \mathtt{blue\_skies} \leftrightarrow \mathtt{clear\_skies}, \mathtt{clear\_skies} \to \neg\mathtt{rainy} \vdash \neg\mathtt{rainy}$$

we derive, by the Abduction rule, the abductive sequent

$$\neg\mathtt{rainy} \Leftarrow [\mathtt{blue\_skies}], \mathtt{blue\_skies} \leftrightarrow \mathtt{clear\_skies}, \mathtt{clear\_skies} \to \neg\mathtt{rainy}$$

Thus, under stable or preferred semantics, $\mathtt{blue\_skies}$ explains $\neg\mathtt{rainy}$. Similarly, $\mathtt{blue\_skies}$ explains $\neg\mathtt{wet\_grass}$, etc.

4.2. Keeping Track of Explanations; Explanations Justifications

In the context of defeasible reasoning explanatory arguments are threatened by defeaters. While abductive sequents $\phi \Leftarrow [\varepsilon], \Gamma$ state that in the context Γ the explanandum ϕ is deducible from the explanation ε, it contains no information of how this explanation is justified against the background of possible defeaters. In the terminology of argumentation theory, abductive sequents cover the illative tier (support) but not the dialectic tier (defeating defeaters) of argumentation [13,14]. In order to keep track of the latter, we incorporate some ideas in the spirit of [15], adapted to logic-based argumentation in general and abductive argumentation frameworks in particular.

Let $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$ be an abductive sequent-based framework with a set $\mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$ of ordinary and abductive arguments, and a set A of attack rules on $\mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S}) \times \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$. For a semantics sem and operator $\Box \in \{\cup, \cap\}$, we consider the following sets:

- $\mathsf{AbdArg}(\phi, [\varepsilon]) = \{a \in \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S}) \mid a \text{ is of the form } \phi \Leftarrow \Gamma, [\varepsilon] \text{ for some } \Gamma \subseteq \mathsf{S}\}$
- $\mathsf{AbdArg}^{\Box}_{\mathsf{sem}}(\phi, [\varepsilon]) = \{a \in \mathsf{AbdArg}(\phi, [\varepsilon]) \mid a \in \Box \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S}))\}$

Thus, $\mathsf{AbdArg}^{\Box}_{\mathsf{sem}}(\phi, [\varepsilon])$ consists of all the abductive arguments in which ε explains ϕ (namely, the elements of $\mathsf{AbdArg}(\phi, [\varepsilon])$), and that belong to the intersection (if $\Box = \cap$) or the union (if $\Box = \cup$) of all the sem-extensions of $\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})$.

To justify the explanation of ϕ by ε with respect to sem and $\Box$, we therefore need to compute the supports of the arguments that defend the elements in $\mathsf{AbdArg}^{\Box}_{\mathsf{sem}}(\phi, [\varepsilon])$ (divided by sem-extensions)

- $\mathsf{Def}_{\mathbb{E}}(a) = \{\mathsf{Supp}(b) \mid b \in \mathbb{E}, b \text{ defends } a \text{ in } \mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S})\}$
- $\mathsf{Justify}^{\Box}_{\mathsf{sem}}(\phi, [\varepsilon]) = \{\mathsf{Def}_{\mathbb{E}}(a) \mid a \in \mathsf{AbdArg}^{\Box}_{\mathsf{sem}}(\phi, [\varepsilon]), \mathbb{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathbb{AAF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}^\star}(\mathsf{S}))\}$

Example 5. Suppose that we want to justify the comment in Example 3 that `sprinklers` credulously stb-explains `wet_grass`. For this, note that:

1. The abductive sequent $a = \texttt{wet\_grass} \Leftarrow [\texttt{sprinklers}], \texttt{sprinklers} \to \texttt{wet\_grass}$ is in $\mathsf{AbdArg}(\texttt{wet\_grass}, [\texttt{sprinklers}])$ and $\mathsf{AbdArg}^{\cup}_{\mathsf{stb}}(\texttt{wet\_grass}, [\texttt{sprinklers}])$.

2. By Abductive Defeat, the abductive sequent a in Item 1 is attacked by the sequent $b = \texttt{rainy}, \texttt{rainy} \to \neg\texttt{sprinklers} \Rightarrow \neg\texttt{sprinklers}$, which in turn is counter-attacked (using Defeat) by $c = \texttt{clear\_skies}, \texttt{clear\_skies} \to \neg\texttt{rainy} \Rightarrow \neg\texttt{rainy}$. It follows that c defends a.

3. By the introduced notation, $\mathsf{Supp}(c) = \{\texttt{clear\_skies}, \texttt{clear\_skies} \to \neg\texttt{rainy}\}$ is in $\mathsf{Def}_{\mathbb{E}}(a)$, where $\mathbb{E}$ is one of the two stable extensions of the abductive argumentation framework under consideration. Thus, for these a and $\mathbb{E}$, we have:

$$(\star)\quad \begin{array}{l} \mathsf{Def}_{\mathbb{E}}(a) \in \mathsf{Justify}^{\cup}_{\mathsf{stb}}(\texttt{wet\_grass}, [\texttt{sprinklers}]), \\ \{\texttt{clear\_skies}, \texttt{clear\_skies} \to \neg\texttt{rainy}\} \in \mathsf{Def}_{\mathbb{E}}(a). \end{array}$$

An intuitive description of $(\star)$ is the following: `sprinklers` is an explanation for `wet_grass`. The set $\{\texttt{clear\_skies}, \texttt{clear\_skies} \to \neg\texttt{rainy}\}$ is a justification for this explanation. Indeed, it is assumed that the sky is clear, and in that case there is no rain. Therefore, the wetness of the grass can be explained by the operation of the sprinklers.

4.3. Explanations Reduction; Avoiding Logically Equivalent Explanations

By its definition, if ε explains ϕ (either internally or externally), then – unless the range of the explanations is restricted – every formula that is logically equivalent to ε according to the base logic L also explains ϕ. This 'explosion' in the number of explanations may be avoided in several ways, e.g., by introducing appropriate attack rules that exclude logically equivalent alternatives of a derived explanation, or by switching to equivalence classes of logically equivalent formulas (see, e.g., [16]). Briefly, the idea is the following:

1. equivalence in L is defined as usual by: $\psi \equiv \phi$ iff $\psi \vdash \phi$ and $\phi \vdash \psi$.
2. classes of arguments are defined by: $[\![\Gamma \Rightarrow \psi]\!] = \{\Delta \Rightarrow \phi \mid \Delta \in [\![\Gamma]\!], \phi \in [\![\psi]\!]\}$, where: $[\![\psi]\!] = \{\phi \mid \phi \equiv \psi\}$ and $[\![\psi_1, \ldots, \psi_n]\!] = \{\{\phi_1, \ldots, \phi_n\} \mid \forall 1 \leq i \leq n\ \phi_i \in [\![\psi_i]\!]\}$.

Now, given a framework $\mathbb{AF}^{\mathsf{X}}_{\mathsf{L},\mathsf{A}}(\mathsf{S}) = \langle \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S}), \mathsf{A} \rangle$ we switch to a framework whose arguments are classes $[\![a]\!]$ for $a \in \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$, and where $[\![a]\!]$ attacks $[\![b]\!]$ if there are some $a' \in [\![a]\!] \cap \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$ and $b' \in [\![b]\!] \cap \mathrm{Arg}^{\mathsf{X}}_{\mathsf{L}}(\mathsf{S})$ such that $(a', b') \in \mathsf{A}$. As usual, one has to show *independence of the choice of representatives*. This is rather routine.

5. Discussion and Conclusion

Abduction has been widely applied in different deductive systems, such as adaptive logics (see, e.g., [17,18]), and AI-based disciplines, perhaps the most prominent one is logic programing (see [19,20] for surveys). Argumentation-based approaches include frameworks for agent-based dialogues [21,22] and assumption-based argumentation frameworks [23]. In [24,25] abduction is studied as the problem of adding arguments to a given argumentation framework so that a given argument is rendered acceptable.

Our approach offers several novelties. In terms of knowledge representation we transparently represent abductive inferences by an explicit inference rule that produces abductive arguments. The latter are a new type of hypothetical arguments that are subjected to potential defeat. A variety attack rules address the quality of the offered explanation and thereby model critical questions [26] and meta-argumentative reasoning [27]. This is both natural and philosophically motivated, as argued in [28], where also a gap in argumentative accounts of abduction is identified. Instead of imposing desiderata on abductive inferences from the outside we incorporate them in the argumentative reasoning process. Our framework offers a high degree of modularity, and in comparison to approaches in logic programming we allow for fully propositional base logics. Desiderata on abductive arguments can be disambiguated in various ways by simply changing the attack rules, all in the same base framework. This allows for a thorough logical analysis and disambiguation of these properties as demonstrated in Theorems 1, 2 and Example 4.

The presented work is mainly focused on representation considerations. In future work we plan to take advantage of the uniformity of the sequent-based methods for explanation, and carry them on to more expressive logics (involving, e.g., preference relations among arguments) and to other types of explanations. We also plan to further develop meta-theoretical results concerning our setting and incorporate other approaches to the dialectic tier of explanation, such as *related admissibility* [14] or *strong explanation* [29].

References

[1] Adadi A, Berrada M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access. 2018;6:52138–52160.

[2] Čyras K, Rago A, Albini E, Baroni P, Toni F. Argumentative XAI: A Survey. In: Proc. of the 30th International Joint Conference on Artificial Intelligence, (IJCAI'21). ijcai.org; 2021. p. 4392–4399.

[3] Vassiliades A, Bassiliades N, Patkos T. Argumentation and explainable artificial intelligence: a survey. The Knowledge Engineering Review. 2021;36:e5.

[4] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77(2):321–357.

[5] Besnard P, Hunter A. A review of argumentation based on deductive arguments. In: Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. vol. 1. College Publications; 2018. p. 437–484.

[6] Arieli O, Straßer C. Sequent-based logical argumentation. Argument & Computation. 2015;6(1):73–99.

[7] Lipton P. Inference to the Best Explanation. Routledge; 2004. Second edition.

[8] Borg A. Assumptive sequent-based argumentation. Journal of Applied Logics – IfCoLog Journal of Logics and Their Applications. 2020;7(3):227–294.

[9] Gentzen G. Untersuchungen über das logische Schließen I, II. Mathematische Zeitschrift. 1934;39:176–210, 405–431.

[10] Baroni P, Caminada M, Giacomin M. Abstract argumentation frameworks and their semantics. In: Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. vol. 1. College Publications; 2018. p. 159–236.

[11] Meheus J, Verhoeven L, Van Dyck M, Provijn D. Ampliative Adaptive Logics and the Foundation of Logic-Based Approaches to Abduction. Logical and Computational Aspects of Model-Based Reasoning. 2002;25:39–71.

[12] Arieli O, Borg A, Straßer C. Characterizations and classifications of argumentative entailments. In: Proc. 18th Conference on Knowledge Representation and Reasoning (KR'21); 2021. p. 52–62.

[13] Johnson RH. Manifest rationality: A pragmatic theory of argument. Routledge; 2000.

[14] Fan X, Toni F. On computing explanations in argumentation. In: Proc. 29th AAAI Conference on Artificial Intelligence (AAAI'15); 2015. p. 1496–1502.

[15] Borg A, Bex F. A Basic Framework for Explanations in Argumentation. IEEE Intelligent Systems. 2021;36(2):25–35.

[16] Amgoud L, Besnard P, Vesic S. Equivalence in logic-based argumentation. Journal of Applied Non Classical Logics. 2014;24(3):181–208.

[17] Lycke H. The Adaptive Logics Approach to Abduction. In: Logic, Philosophy and History of Science in Belgium; 2008. p. 35–41.

[18] Meheus J, Batens D. A Formal Logic for Abductive Reasoning. Logic Journal of the IGPL. 2006;14(2):221–236.

[19] Denecker M, Kakas A. Abduction in Logic Programming. In: Computational Logic: Logic Programming and Beyond. vol. 2407 of Lecture Notes in Computer Science. Springer; 2002. p. 402–436.

[20] Kakas A, Michael L. Abduction and argumentation for explainable machine learning: A Position Survey. arXiv preprint arXiv:201012896. 2020;.

[21] Bex F, Budzynska K, Walton D. Argumentation and explanation in the context of dialogue. Explanation-aware Computing ExaCt 2012. 2012;9:6.

[22] Arioua A, Croitoru M. Formalizing Explanatory Dialogues. In: Proc. 9th Conf. on Scalable Uncertainty Management (SUM'15). vol. 9310 of Lecture Notes in Computer Science. Springer; 2015. p. 282–297.

[23] Wakaki T. Extended abduction in assumption-based argumentation. In: Proc. IEA/AIE. vol. 11606 of Lecture Notes in Computer Science. Springer; 2019. p. 593–607.

[24] Sakama C. Abduction in Argumentation Frameworks. Journal of Applied Non-Classical Logics. 2018;28(2-3):218–239.

[25] Booth R, Gabbay DM, Kaci S, Rienstra T, Van Der Torre LW. Abduction and Dialogical Proof in Argumentation and Logic Programming. In: Proc. 21st European Conf. on Artificial Intelligence (ECAI'14). IOS Press; 2014. p. 117–122.

[26] Walton D, Reed C, Macagno F. Argumentation Schemes. Cambridge University Press; 2008.

[27] Boella G, Gabbay D, van der Torre L, Villata S. Meta-Argumentation Modelling I: Methodology and Techniques. Studia Logica. 2009;93(2–3):297–355.

[28] Olmos P. Abduction and comparative weighing of explanatory hypotheses: an argumentative approach. Logic Journal of the IGPL. 2019;.

[29] Ulbricht M, Wallner JP. Strong explanations in abstract argumentation. In: Proc. 23rd AAAI Conference on Artificial Intelligence. AAAI Press; 2021. p. 6496–6504.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220140

Admissibility in Strength-Based Argumentation: Complexity and Algorithms

Yohann BACQUEY [a] Jean-Guy MAILLY [a] Pavlos MORAITIS [a,b] Julien ROSSIT [a]
[a] *Université Paris Cité, LIPADE, F-75006 Paris, France*
yohann.bacquey@etu.u-paris.fr,{jean-guy.mailly,pavlos.moraitis,julien.rossit}@u-paris.fr
[b] *Argument Theory, Paris, France*

Abstract. Recently, Strength-based Argumentation Frameworks (StrAFs) have been proposed to model situations where some quantitative strength is associated with arguments. In this setting, the notion of accrual corresponds to sets of arguments that collectively attack an argument. Some semantics have already been defined, which are sensitive to the existence of accruals that collectively defeat their target, while their individual elements cannot. However, until now, only the surface of this framework and semantics have been studied. Indeed, the existing literature focuses on the adaptation of the stable semantics to StrAFs. In this paper, we push forward the study and investigate the adaptation of admissibility-based semantics. Especially, we show that the strong admissibility defined in the literature does not satisfy a desirable property, namely Dung's fundamental lemma. We therefore propose an alternative definition that induces semantics that behave as expected. We then study computational issues for these new semantics, in particular we show that complexity of reasoning is similar to the complexity of the corresponding decision problems for standard argumentation frameworks in almost all cases. We then propose a translation in pseudo-Boolean constraints for computing (strong and weak) extensions. We conclude with an experimental evaluation of our approach which shows in particular that it scales up well for solving the problem of providing one extension as well as enumerating them all.

Keywords. abstract argumentation, argument strength, accrual of arguments

1. Introduction

Among widespread knowledge representation and reasoning techniques proposed in the literature of Artificial Intelligence over the last decades, Abstract Argumentation [1] is an intuitive but yet powerful tool for dealing with conflicting information. Since then, the initial work of Dung has been actively extended and enriched in many directions, *e.g.* considering other kinds of relations between arguments [2] or additional information associated with arguments or attacks [3,4]. Among them, Strength-based Argumentation Frameworks (StrAFs) [5] allow to associate a quantitative information with each argument. This information is a weight that intuitively represents the intrinsic strength of an argument, and is then naturally combined with attacks between arguments to induce a defeat relation that allows either to confirm an attack between two arguments or to cancel it, if the attacked argument is stronger than the attacker (w.r.t. their respective weights). StrAFs extend further this notion of defeat among arguments by building a defeat that is based on a collective attack of a group of arguments (or accrual) and by offering associate semantics. Within these semantics, arguments can collectively defeat arguments that they cannot defeat individually. Intuitively speaking, these accrual-sensitive semantics allow some kind of compensation among arguments, where the accumulation of weak arguments can create a synergy and get rid of a stronger one they collectively attack. This reasoning approach allows to produce extensions that are not considered when applying classical semantics. In [5] the authors presented the basics of StrAFs inspired by Dung's semantics for abstract argumentation along with some theoretical and computational results concerning classical issues related to abstract argumentation (*i.e.* acceptability semantics, semantics inclusion, extensions existence and verification, etc.). In this paper we propose a state of the art advancement in StrAFs by presenting original theoretical and computational results related to different aspects. More particularly the contribution of this work lies into the following aspects. The semantics proposed in [5] exist in two versions (namely strong, and weak). Roughly speaking, a set is strongly conflict-free iff none of its elements attacks another one, whereas a set is weakly conflict-free iff it does not contain any (successful) accrual against one of its elements. After detecting that strong admissibility fails to satisfy a desirable property in Dung's abstract argumentation frameworks, namely his Fundamental Lemma [1], we propose an alternative definition for strong admissibility in order to remedy this issue and we define new admissibility-based semantics for StrAFs. Furthermore, we study the complexity of reasoning with these semantics and in particular we show that, surprisingly, the complexity does not increase w.r.t. the complexity of reasoning with AFs. For computing the extensions under these semantics, we propose algorithms based on pseudo-Boolean constraints.

2. Background Notions

We assume that the reader is familiar with abstract argumentation [1]. We consider finite *argumentation frameworks* (AFs) $\langle \mathscr{A}, \mathscr{R} \rangle$, where $\mathscr{A}$ is the set of *arguments*, and $\mathscr{R} \subseteq \mathscr{A} \times \mathscr{A}$ is the *attack relation*. We will use $\mathrm{cf}(AF)$ and $\mathrm{ad}(AF)$ to denote, respectively, the conflict-free and admissible sets of an AF AF, and $\mathrm{co}(AF)$, $\mathrm{pr}(AF)$ and $\mathrm{st}(AF)$ for its extensions under the complete, preferred and stable semantics. For more details on the semantics of AFs, we refer the interested reader to [1,6]. A *Strength-based Argumentation Framework* (StrAF) [5] is a triple $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ where $\mathscr{A}$ and $\mathscr{R}$ are arguments and attacks, and $\mathscr{S} : \mathscr{A} \to \mathbb{N}$ is a *strength function*. An example of such a StrAF is depicted at Figure 1, where nodes represent arguments, edges represent attacks, and the numbers close to the nodes represent the arguments strength. These strengths intuitively represent the intrinsic robustness associated with an argument and allow to

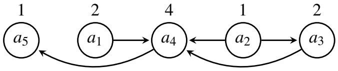


Figure 1. A StrAF Example

induce a *defeat* relation: an argument a defeats another argument b when a attacks b and the strength associated with b does not overcome that with a. This framework also offers the notion of collective defeat, *i.e.* sets of arguments that can jointly defeat their target while they cannot do so separately. First, we call an *accrual* a set of arguments that collectively attack a same target, *i.e.* a set $\kappa \subseteq \mathscr{A}$ s.t. $\exists c \in \mathscr{A}$ s.t. $\forall\ a \in \kappa$, $(a,c) \in \mathscr{R}$. Moreover, we say that κ is *an accrual that attacks c*. Then, for $\kappa' \subseteq \mathscr{A}$ an accrual, κ *attacks* κ' iff $\exists a \in \kappa'$ s.t. κ attacks a.

Example 1. *Consider again StrAF from Figure 1. We can observe several examples of accruals,* e.g. $\kappa_1 = \{a_1, a_2\}$ *and* $\kappa_2 = \{a_1, a_3\}$*, that both attack* a_4*. Notice that any attack* $(a_i, a_j) \in \mathscr{R}$ *induces an accrual* $\{a_i\}$ *attacking* a_j*.*

We need to assess the collective strength of an accrual.

Definition 1 (Collective Strength)**.** *Let StrAF* $= \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ *be a StrAF and* $\kappa = \{a_1, ..., a_n\} \subseteq \mathscr{A}$ *be an accrual. Then the collective strength associated with* κ *is* $\text{coval}_{\oplus}(\kappa) = \oplus(\mathscr{S}(a_1), \ldots, \mathscr{S}(a_n))$ *where* $\oplus$ *is an aggregation operator.*

The operator $\oplus$ must satisfy some properties discussed in [5]. An example of suitable operator is $\oplus = \sum$. If $\oplus$ is clear from the context, we simply write coval for $\text{coval}_{\oplus}$.

Definition 2 (Collective Defeat)**.** *Let StrAF* $= \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ *be a StrAF,* $a \in \mathscr{A}$*, and* $\oplus$ *an aggregation operator. Then, an accrual* κ defeats *a with respect to* $\text{coval}_{\oplus}$*, denoted by* $\kappa \rhd_{\oplus} a$*, iff* $\kappa \subseteq \mathscr{A}$ *is an accrual that attacks a and* $\text{coval}_{\oplus}(\kappa) \geq \mathscr{S}(a)$*. If* $\oplus$ *is clear from the context, we use* $\kappa \rhd a$ *instead of* $\kappa \rhd_{\oplus} a$*.*

In the rest of the paper, we focus on $\oplus = \sum$ in examples and the pseudo-Boolean encoding defined in Section 4.2. But, unless explicitly stated otherwise, our results remain valid for any $\oplus$ satisfying the properties from [5].

Definition 3. *Let StrAF* $= \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ *be a StrAF,* $\oplus$ *an aggregation operator and* $\kappa \subseteq \mathscr{A}$*,* $\kappa' \subseteq \mathscr{A}$ *two accruals. Then* κ defeats κ'*, denoted by* $\kappa \rhd_{\oplus} \kappa'$*, iff* $\exists a \in \kappa'$ *s.t.* $\kappa \rhd_{\oplus} a$*.*

Example 2. *Continuing Example 1, notice that* $\text{coval}_{\sum}(\kappa_1) = 3 < \mathscr{S}(a_4)$*, so* $\kappa_1 \not\rhd a_4$*. On the contrary,* $\text{coval}_{\sum}(\kappa_2) = 4 \geq \mathscr{S}(a_4)$*, so* $\kappa_2 \rhd a_4$*.*

StrAF semantics rely on two possible adaptations of the notion of conflict-freeness:

Definition 4 (Conflict-freeness/Defence)**.** *Given StrAF* $= \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ *a StrAF,* $\oplus$ *an aggregation operator, and* $S \subseteq \mathscr{A}$*,*

- *S is* strongly conflict-free *iff* $\nexists a, b \in S$ *s.t.* $(a,b) \in \mathscr{R}$*.*
- *S is* weakly conflict-free *iff there are no accruals* $\kappa_1 \subseteq S$ *and* $\kappa_2 \subseteq S$ *s.t.* $\kappa_1 \rhd_{\oplus} \kappa_2$*.*
- *S defends an element* $a \in \mathscr{A}$ *iff for all accruals* $\kappa_1 \subseteq \mathscr{A}$*, if* $\kappa_1 \rhd_{\oplus} a$*, then there exists an accrual* $\kappa_2 \subseteq S$ *s.t.* $\kappa_2 \rhd_{\oplus} \kappa_1$*.*

Intuitively, strongly conflict-free sets are “classically” conflict-free, *i.e.* there is no attack between two arguments members of such a set. On the contrary, weakly conflict-free sets are “defeat-free”: attacks between arguments are permitted as long as they do not result in a defeat neither individual nor collective. We use (respectively) $\text{cf}_S^{\oplus}$ and $\text{cf}_W^{\oplus}$ to denote these sets (or simply cf_S and cf_W when $\oplus$ is clear from the context). Then, admissibility and extension-based semantics can be defined either strong or weak. Namely:

Definition 5 (Semantics for StrAFs [5]). *Given $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ a StrAF, $\oplus$ an aggregation operator, and $S \subseteq \mathscr{A}$ a strong (resp. weak) conflict-free set,*

- *S is a* strong (resp. weak) admissible set *iff S defends all elements of S.*
- *S is a* strong (resp. weak) preferred extension *iff S is a $\subseteq$-maximal strong (resp. weak) admissible set.*
- *S is a* strong (resp. weak) stable extension *iff $\forall a \in \mathscr{A} \backslash S$, $\exists \kappa \subseteq S$ s.t. $\kappa \rhd_{\oplus} a$.*

For σ an extension-based semantics and $X \in \{S, W\}$ meaning respectively *strong* and *weak*, we use $\sigma_X^{\oplus}(StrAF)$ to denote the X-σ extensions of $StrAF$. We drop $\oplus$ from the notation where there is no possible ambiguity. It is proven in [5] that Dung's AFs are a subclass of StrAFs, where strong and weak semantics coincide. This result is useful for proving complexity results. However, in [5] authors only focus on complexity issues for the (weak and strong) stable semantics.

3. Admissibility-based Semantics for StrAFs

In our study, we investigate computational issues for admissibility-based semantics. A formal definition of (weak or strong) complete semantics is missing in [5], but matching the definition of (weak or strong) admissibility with the classical definition of the complete semantics, a straightforward definition can be stated as follows:

Definition 6. *Let $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ be a StrAF, and $\oplus$ an aggregation operator. A strong (resp. weak) admissible set $S \subseteq \mathscr{A}$ is a* strong (resp. weak) complete extension *of StrAF if S contains all the arguments that it defends.*

Now we study these semantics and in particular, we show that surprisingly this intuitive definition of the complete semantics based on strong admissibility fails to satisfy a desirable property, namely the Fundamental Lemma, which states that admissible sets can be extended by the arguments that they defend. This leads us to redefine strong admissibility (and the associated complete and preferred semantics) in Section 3.1. On the contrary, the definition of weak admissibility is proved to be suitable in Section 3.2.

3.1. Revisiting Strong Admissibility

First, we observe that the usual inclusion relation between the preferred and complete semantics is not satisfied for the strong semantics of StrAFs. Moreover, the universal existence of complete extensions does not hold either.

Proposition 1. *There exists StrAF s.t.* $\mathrm{pr}_S(StrAF) \not\subseteq \mathrm{co}_S(StrAF)$, *and* $\mathrm{co}_S(StrAF) = \emptyset$.

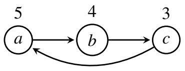


Figure 2. Example proving that $\mathrm{pr}_S(StrAF) \not\subseteq \mathrm{co}_S(StrAF)$

The StrAF from Figure 2 shows that the Fundamental Lemma does not hold for strong semantics of StrAFs: $\{a\}$ is strongly admissible, and defends c, but $\{a\} \cup \{c\}$ is not strongly admissible. A way to solve this issue is to redefine strong admissibility:

Definition 7 (Strong Semantics Revisited)**.** *Let $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ be a StrAF and $\oplus$ an aggregation operator. A set $S \in \mathrm{cf}_S(StrAF)$* strongly defends *an argument a if S defends a against all the accruals that defeat it,* i.e. *$\forall \kappa \subseteq \mathscr{A}$ s.t. $\kappa \rhd a$, $\exists \kappa' \subseteq S$ s.t. $\kappa' \rhd \kappa$, and $S \cup \{a\}$ is strongly conflict-free. Then, a set $S \subseteq \mathscr{A}$ is* strongly admissible *if it is strongly conflict-free and it strongly defends all its elements. Moreover, S is a* strong preferred extension *iff S is a $\subseteq$-maximal strong admissible set; and S is a* strong complete extension *iff S strongly defends all its elements.*

If we consider again the StrAF from Figure 2, observe that this time, the strongly admissible set $\{a\}$ does not strongly defends c, since $\{a, c\}$ is not strongly conflict-free. Thus $\{a\}$ is a strong complete extension of this StrAF, following Definition 7. Now, Dung's Fundamental Lemma can be adapted to strong admissibility.

Lemma 1 (Fundamental Lemma for Strong Admissibility)**.** *Let $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ be a StrAF and $\oplus$ an aggregation operator. Let $S \subseteq \mathscr{A}$ be a strongly admissible set, and a, a' two arguments that are strongly defended by S against all their defeaters. Then, $S' = S \cup \{a\}$ is strongly admissible.*

Lemma 1 implies a relation between strong preferred and complete extensions:

Proposition 2. *For any StrAF and $\oplus$, $\mathrm{pr}_S^{\oplus}(StrAF) \subseteq \mathrm{co}_S^{\oplus}(StrAF)$.*

This guarantees the existence of at least one strong complete extension for any StrAF: since $\emptyset$ is a strong admissible set for any *StrAF*, then *StrAF* admits some $\subseteq$-maximal strong admissible sets, *i.e.* $\mathrm{pr}_S(StrAF) \neq \emptyset$, which implies $\mathrm{co}_S(StrAF) \neq \emptyset$.

Example 3. *Let us consider again the StrAF provided by Figure 1. Its strongly admissible sets are $\mathrm{ad}_S(StrAF) = \{\emptyset, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_2\}, \{a_1, a_3, a_5\}\}$. Then, the strong preferred and complete extensions are $\mathrm{pr}_S(StrAF) = \mathrm{co}_S(StrAF) = \{\{a_1, a_2\}, \{a_1, a_3, a_5\}\}$.*

Finally, we prove that the new definition of strong admissibility does not change the fact that strong stable extensions are strongly admissible (and even strong preferred).

Proposition 3. *For any StrAF and $\oplus$, $\mathrm{st}_S^{\oplus}(StrAF) \subseteq \mathrm{pr}_S^{\oplus}(StrAF)$.*

3.2. Properties of the Weak Semantics

Regarding now weak semantics as defined in [5], the usual result still holds for StrAFs.

Lemma 2 (Fundamental Lemma for Weak Admissibility)**.** *Let $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ be a StrAF and $\oplus$ an aggregation operator. Given $S \in \mathrm{ad}_W(StrAF)$, and a, a' two arguments that are defended by S, $S' = S \cup \{a\}$ is weakly admissible, and S' defends a'.*

Proposition 4. *For any StrAF and $\oplus$, $\mathrm{pr}_W^{\oplus}(StrAF) \subseteq \mathrm{co}_W^{\oplus}(StrAF)$.*

Similarly to what we have noticed previously for strong admissibility, $\emptyset$ is weakly admissible for any StrAF. This implies the existence of at least one weak preferred extension, and then one weak complete extension for any StrAF.

Example 4. *Consider again the StrAF from Figure 1. One identifies the weakly admissible sets $\mathrm{ad}_W(StrAF) = \mathrm{ad}_S(StrAF) \cup \{\{a_2, a_3\}, \{a_1, a_2, a_3\}, \{a_1, a_2, a_3, a_5\}\}$. Then, $\mathrm{pr}_W(StrAF) = \mathrm{co}_W(StrAF) = \{\{a_1, a_2, a_3, a_5\}\}$.*

Contrary to the case of stable semantics [5], we do not have $\mathrm{co}_S(StrAF) \subseteq \mathrm{co}_W(StrAF)$. This comes from the fact that our strong and weak complete semantics are not based on the same notion of defense. However, we observe that, for any StrAF, each strong complete extension is included in some weak preferred (and complete) extension.

Proposition 5 (Strong/Weak Semantics Relationship). *Given $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ and $\oplus$ an aggregation operator, $\forall E \in \mathrm{co}_S(StrAF)$, $\exists E' \in \mathrm{pr}_W(StrAF)$ s.t. $E \subseteq E'$.*

We do not need a counterpart to Proposition 3: the definition of semantics based on weak admissibility is not modified, so [5, Proposition 1] still holds in this case.

3.3. Dung Compatibility

Previous work on StrAFs showed that this framework generalizes Dung's AF, with a correspondence of StrAF semantics with AF semantics in this case. Following the new definition of strong admissible sets, one might fear that this property does not hold for strong admissibility-based semantics. However, we show here that it still does, as well as for weak complete semantics. Let us recall the transformation of an AF into a StrAF [5].

Definition 8. *Given an argumentation framework $AF = \langle \mathscr{A}, \mathscr{R} \rangle$, the StrAF associated with AF is $StrAF_{AF} = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ with $\mathscr{S}\ (a) = 1, \forall a \in \mathscr{A}$ and $\mathrm{coval} = \sum$.*

Now we can state the following proposition, that extends Dung Compatibility from [5] to the semantics studied in this paper.

Proposition 6 (Dung Compatibility). *Let $AF = \langle \mathscr{A}, \mathscr{R} \rangle$ be an AF, and $StrAF_{AF} = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ from Def. 8. For $\sigma \in \{\mathrm{ad}, \mathrm{pr}, \mathrm{co}\}$, $\sigma(AF) = \sigma_X(StrAF_{AF})$, for $X \in \{S, W\}$.*

4. Complexity and Algorithms

Now we provide some insight on computational issues for admissibility-based semantics of StrAFs, *i.e.* we identify the computational complexity of several classical reasoning problems under these semantics, and we provide algorithms (based on pseudo-Boolean encoding) for solving them. While the complexity results are generic regarding the choice of $\oplus$, the algorithms focus on $\oplus = \sum$.

4.1. Complexity Analysis

We assume that the reader is familiar with basic notions of complexity, and otherwise we refer to [7] for details on complexity in argumentation, and [8] for a more general overview of complexity. We focus on three classical reasoning problems in abstract argumentation, namely *verification* ("Is a given set of arguments an extension?"), *credulous acceptability* ("Is a given argument in some extension?") and *skeptical acceptability* ("Is a given argument in each extension?"). Formally, for $\sigma \in \{\mathrm{ad}, \mathrm{pr}, \mathrm{co}\}$ and $X \in \{S, W\}$:

- σ-X-Ver: Given $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ and $S \subseteq \mathscr{A}$, is S a member of $\sigma_X(StrAF)$?
- σ-X-Cred: Given $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ and $a \in \mathscr{A}$, is a in some $S \in \sigma_X(StrAF)$?
- σ-X-Skep: Given $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ and $a \in \mathscr{A}$, is a in each $S \in \sigma_X(StrAF)$?

We recall that these reasoning problems are already considered only for the (weak and strong) stable semantics in [5]. In the following, we assume a fixed $\oplus$, that can be computed in polynomial time. This is not a very strong assumption, since it is the case with the classical aggregation operators (*e.g.* $\sum, \max, \ldots$). Proposition 6 implies that the complexity of reasoning with standard AFs provides a lower bound complexity of reasoning with StrAFs. So we focus on identifying upper bounds.

Proposition 7. *The complexity of the decision problems σ-X-Ver, σ-X-Cred and σ-X-Skep is as described in Table 1.*

	σ-X-Ver	σ-X-Cred	σ-X-Skep
ad_X	P	NP-c	Trivial
co_X	P	NP-c	in coNP
pr_X	coNP-c	NP-c	Π_2^P-c

Table 1. Complexity of reasoning for σ_X with $\sigma \in \{\mathrm{ad}, \mathrm{co}, \mathrm{pr}\}$ and $X \in \{S, W\}$. Trivial means that all instances are trivially "NO" instances, and $\mathscr{C}$-c means $\mathscr{C}$-complete, for $\mathscr{C}$ a complexity class in the polynomial hierarchy.

As for the strong (resp. weak) stable semantics [5], we prove here that the higher expressivity of StrAFs (compared to AFs) does not come at the price of a complexity blow-up. Only the case of skeptical acceptability under strong (resp. weak) complete semantics requires a deeper analysis, since we only provide the coNP upper bound. We observe that the choice of the weak or strong semantics does not impact on the complexity.

4.2. Algorithms

For computing the strong (resp. weak) admissible sets and complete extensions, we propose a translation of StrAF semantics in pseudo-Boolean (PB) constraints [9]. Such a constraint is an (in)equality $\sum_i w_i \times l_i \# k$ where w_i and k are positive integers, and $\# \in \{>, \geq, =, \neq, \leq, <\}$. l_i is a literal, *i.e.* $l_i = v_i$ or $l_i = \overline{v_i} = 1 - v_i$, where v_i is a Boolean variable. Determining whether a set of PB constraints has a solution is a NP-complete problem, that generalizes the Boolean satisfiability (SAT) problem. Despite the high complexity of this problem, it can be efficiently solved in many cases, see *e.g.* [10,11].

Strong and Weak Conflict-freeness. Now we describe our PB encoding of StrAF semantics. For ensuring self-containment of the paper, we recall the encoding of strong and weak conflict-freeness [5]. Given $StrAF = \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ and coval $= \sum$, we define a set of Boolean variables $\{x_i \mid a_i \in \mathscr{A}\}$ associated with each argument, where $x_i = 1$ means that a_i belongs to the set of arguments characterized by the solutions of the PB constraints. Then, strong and weak conflict-freeness are respectively encoded by (1) and (1'):

(1) $\forall (a_i, a_j) \in \mathscr{R}$, add the constraint $x_i + x_j \leq 1$
(1') $\forall a \in \mathscr{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} \mathscr{S}(a_i) \times x_i < x \times \mathscr{S}(a) + \bar{x} \times M$

with M an arbitrary large natural number that is greater than the sum of the strengths of the arguments (*i.e.* $M > \sum_{a \in \mathscr{A}} \mathscr{S}(a)$), $\Gamma^-(a) = \{b \mid (b, a) \in \mathscr{R}\}$ is the set of attackers of $a \in \mathscr{A}$, and x is the Boolean variable associated with a.[1] A solution to the set of

[1] Notice that the constraints referring to $\Gamma^-(a)$ must be added even when $\Gamma^-(a) = \emptyset$.

constraints **(1)** (resp. **(1')**) yields a strong (resp. weak) conflict-free set $E = \{a_i \mid x_i = 1\}$. We prove this claim with Proposition 8. First, let us introduce some notations. Given $S \subseteq \mathscr{A}$, $\omega_S : \{x_i \mid a_i \in \mathscr{A}\} \to \{0,1\}$ is a mapping s.t. $\omega_S(x_i) = 1$ iff $a_i \in S$.

Proposition 8. *Given StrAF* $= \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ *and* $S \subseteq \mathscr{A}$*,* $S \in \mathrm{cf}_S(\mathit{StrAF})$ *(resp.* $S \in \mathrm{cf}_W(\mathit{StrAF})$*) iff* ω_S *satisfies the set of constraints* **(1)** *(resp.* **(1')***).*

Strong and Weak Admissibility. For encoding strong (resp. weak) admissibility, one must add to the set of constraints **(1)** (resp. **(1')**) some new constraints that represent the strong defense (resp. defense) property. To do so, one needs to introduce new Boolean variables $\{y_i \mid a_i \in \mathscr{A}\}$ s.t. $y_i = 1$ means that a_i is defeated by the set of arguments characterized by the solution of the PB constraints. Then, three constraints are added (the same ones for strong and weak admissibility):

(2) $\forall a \in \mathscr{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} \mathscr{S}(a_i) \times x_i \geq y \times \mathscr{S}(a)$
(3) $\forall a \in \mathscr{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} \mathscr{S}(a_i) \times x_i \leq \overline{y} \times \mathscr{S}(a) + y \times M$
(4) $\forall a \in \mathscr{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} \mathscr{S}(a_i) \times \overline{y_i} \leq x \times \mathscr{S}(a) + \overline{x} \times M$

The sets of constraints **(2)** and **(3)** ensure that $y = 1$ iff a is defeated by some $\kappa \subseteq E = \{a_i \mid x_i = 1\}$, and the constraints **(4)** ensure that E defends all its elements. The following proposition shows the correctness of the encodings.

Proposition 9. *Given StrAF* $= \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ *and* $S \subseteq \mathscr{A}$*,* $S \in \mathrm{ad}_S(\mathit{StrAF})$ *(resp.* $S \in \mathrm{ad}_W(\mathit{StrAF})$*) iff* ω_S *satisfies the sets of constraints* **(1)** *(resp.* **(1')***),* **(2)***,* **(3)** *and* **(4)***.*

Strong and Weak Complete Semantics. Now, for computing the strong (resp. weak) extensions, one must consider the sets of constraints **(1)** (resp. **(1')**), **(2)**, **(3)** and **(4)**, and add a last set of constraints, respectively **(5)** for strong complete semantics, and **(5')** for weak complete semantics:

(5) $\forall a \in \mathscr{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} (\mathscr{S}(a_i) \times \overline{y_i}) + \sum_{a_i \in \Gamma^-(a)} (M \times x_i) + \sum_{a_i' \in \Gamma^+(a)} (M \times x_i') \geq \overline{x} \times \mathscr{S}(a)$
(5') $\forall a \in \mathscr{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} \mathscr{S}(a_i) \times \overline{y_i} \geq \overline{x} \times \mathscr{S}(a)$

where $\Gamma^+(a) = \{b \in \mathscr{A} \mid (a,b) \in \mathscr{R}\}$ is the set of arguments attacked by a. These constraints ensure that an argument is not accepted only if it is not (strongly) defended. Again, we prove the correctness of the encodings:

Proposition 10. *Given StrAF* $= \langle \mathscr{A}, \mathscr{R}, \mathscr{S} \rangle$ *and* $S \subseteq \mathscr{A}$*,* $S \in \mathrm{co}_S(\mathit{StrAF})$ *(resp.* $S \in \mathrm{co}_W(\mathit{StrAF})$*) iff* ω_S *satisfies the sets of constraints* **(1)** *(resp.* **(1')***),* **(2)***,* **(3)***,* **(4)** *and* **(5)** *(resp.* **(5')***).*

Acceptability and Verification. Obtaining one (resp. each) solution for one of the sets of constraints defined previously corresponds to obtaining one (resp. each) extension of the StrAF under the corresponding semantics. For checking whether a given argument a_i is credulously accepted, one simply needs to add the constraint $x_i = 1$. If a solution exists, then it corresponds to an extension that contains a_i, proving that this argument is credulously accepted. Otherwise, a_i is not credulously accepted. For skeptical acceptability, one needs to add the constraint $x_i = 0$. In this case, a solution exhibits an extension that does not contain a_i, thus this argument is not skeptically accepted. In the case where no solution exists, then the argument is skeptically accepted. Finally, for checking whether a set of arguments $S \subseteq \mathscr{A}$ is an extension, one needs to add the constraints $x_i = 1$ for each $a_i \in S$, as well as $x_i = 0$ for each $a_i \in \mathscr{A} \setminus S$. A solution exists for the new set of constraints iff S is an extension under the considered semantics.

Strong and Weak Preferred Semantics. Finally, let us mention an approach to handle reasoning with strong and weak preferred semantics. Because of the higher complexity of skeptical reasoning under these semantics (recall Proposition 7), it is impossible (under the usual assumption that the polynomial hierarchy does not collapse) to find a (polynomial) encoding of these semantics in PB constraints. However, PB solvers can be used as oracles to find (with successive calls) preferred extensions. Algorithm 1 describes our method to do this for strong preferred semantics (replacing **(1)** by **(1')** provides an algorithm for weak preferred semantics). At start, we add the four constraints corresponding to a strong (resp. weak) admissible set and solve the instance, with the PB solver as a coNP oracle. Then we force the arguments within the extension to stay in the next one by adding the constraint on line 4. To avoid getting the same solution as in the previous step, we make sure that at least one argument outside the previous extension will be in the next one (line 5). This method iteratively extends an admissible set into a preferred extension, that is finally returned when the solver cannot find any (larger) solution.

Algorithm 1 Compute a strong preferred extension

```
$P$ = PB problem with constraints (1), (2), (3) and (4)
while $P.solve() \neq null$ do
    $E \leftarrow P.solve()$
    $P.add\_constraint(x_1 + x_2 + \cdots + x_n = n)$, with $E = \{a_1, a_2, \ldots, a_n\}$
    $P.add\_constraint(x_1 + x_2 + \cdots + x_m \geq 1)$, with $\mathscr{A} \setminus E = \{a'_1, a'_2, \ldots, a'_m\}$
end while
return $E$
```

5. Experimental Evaluation

For estimating the scalability of our method based on pseudo-Boolean constraints, we present now some results obtained from our experimental evaluation using two prominent PB solvers: Sat4j [12] and RoundingSat [11]. While Sat4j is based on saturation, RoundingSat uses the division rule (see [11] for a discussion on both approaches). We focus here on the most relevant results; full results are presented in [13].

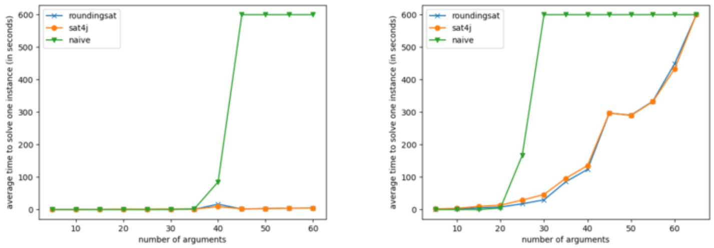
Benchmark Generation. We generate benchmarks in a format adapted to StrAFs, inspired by ASPARTIX formalism [14]. We consider two classes of randomly generated graphs. First, with the Erdös–Rényi model (ER) [15], given a set of arguments $\mathscr{A}$, and $p \in [0,1]$, we generate a graph such that for each $(a,b) \in \mathscr{A} \times \mathscr{A}$, a attacks b with a probability p. We consider two values for the probability, namely $p \in \{0.1, 0.5\}$. Then, with the Barabási–Albert (BA) model [16], a graph of n nodes is grown by attaching new nodes with m edges that are preferentially attached to existing nodes with a high degree. These types of graphs have been frequently used for studying computational aspects of formal argumentation, in particular during the ICCMA competitions [17]. The choice of a generation model provides the arguments $\mathscr{A}$ and attacks $\mathscr{R}$. We attach a random strength $\mathscr{S}(a) \in \{1,\dots,20\}$ to each $a \in \mathscr{A}$. For each generation model, we build 20 StrAFs for each $|\mathscr{A}| \in \{5,10,15,\dots,60\}$. Parameters ($p \in \{0.1, 0.5\}$ for ER, $m = 1$ for BA) are chosen to avoid graphs with a high density of attacks, that would prevent the existence of meaningful extensions (*e.g.* non-empty ones). Larger StrAFs (with $|\mathscr{A}| \in \{5,10,\dots,250\}$) have been generated with the same parameters ($p \in \{0.1, 0.5\}$ for ER, $m = 1$ for BA) for studying the problem of providing one extension.

Experimental Setting. The experiments were run on a Windows computer (using Windows Subsystem for Linux), with an Intel Core i5-6600K 3.50GHz CPU and 16GB of RAM. The timeout is set to 600 seconds (same as the timeout at ICCMA [18]).

Results. We are interested in the semantics σ_X, with $\sigma \in \{\mathrm{pr}, \mathrm{st}, \mathrm{co}\}$ and $X \in \{S, W\}$. The encodings for st_X ($X \in \{S, W\}$) are those proposed in [5], while the encoding for the other semantics are those described in Section 4.2. For each generated *StrAF*, and each of these semantics σ_X, the two tasks we are interested in consist in enumerating all extensions and finding one extension. We first focus on the runtime for enumerating σ_X extensions, which provides an upper bound of the runtime for solving other classical reasoning tasks. To do so, we use a Python script that converts a StrAF into a set of PB constraints.[2] The set of extensions is then obtained in a classical iterative way: once an extension is returned by the PB solver, we add a new constraint that forbids this extension, and we call again the solver on this updated set of PB constraints. This process is repeated until the set of constraints becomes unsatisfiable, which means that all the extensions have been obtained. Concerning the preferred extensions, this iterative approach is combined with Algorithm 1. In order to measure the performance of our approach, and since there is no other computational approach for StrAF semantics yet, we also implemented a so-called *naive* algorithm that enumerates all sets of arguments and then checks, for each of them, if it is a σ_X extension. Figure 3 presents the average runtimes w.r.t. instance sizes (*i.e.* $|\mathscr{A}|$) for various semantics and StrAF families as described before. As a first result, we observe in Figure 3a that runtime for enumerating extensions (with the PB approach) is reasonable (*i.e.* less than a minute) for most of the cases considered in our study, when the PB approach is used, while the naive approach reaches the timeout for most of the large instances (in particular, all the instances with $|\mathscr{A}| \geq 45$). The average runtimes are higher in only two situations: the enumeration of strong preferred and strong complete extensions, with the BA graphs. However, even in such situations where the enumeration is harder (*e.g.* for pr_S-extensions on BA graphs, as depicted on Figure 3b), the PB solvers clearly outperform the naive algorithm, which reaches the timeout in every instance when $|\mathscr{A}| \geq 30$, while the PB approach can enumerate extensions for larger graphs.

We also study the classical problem of providing one extension, for StrAFs of larger sizes (recall that here $|\mathscr{A}| \in \{5,10,\dots,250\}$). Figure 4 shows that the PB solvers (in particular, Sat4j) provide one extension for these large graphs under two minutes, even for the preferred semantics (which is the hardest one, in our study, from the computational point of view). Concerning the respective performances of the two PB solvers, Figure 4 shows that RoundingSat processes faster for fast-to-compute instances (*i.e.* the smallest ones), while Sat4j outperforms it for instances of larger size. While we do not have explanations for this phenomenon, a plausible assumption is that it is related to the difference

[2] The script is available here: `https://github.com/BacqueyYohann/SolvingStrafs`.



(a) co_W on ER graphs ($p = 0.1$)

(b) pr_S on BA graphs

Figure 3. Enumeration runtime

of the underlying algorithms (saturation for Sat4j and division rule for RoundingSat). Similar things have been observed for SAT solvers used in the case of standard AFs [19].

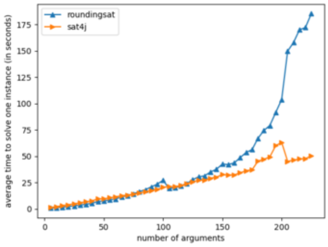


Figure 4. Finding one extension runtime under pr_s on BA graphs

As a general conclusion on our experimental analysis, we observe that the PB approach for reasoning with StrAFs generally scales up well, for both problems of enumerating extensions and providing one extension.

6. Conclusion

Strength-based Argumentation Frameworks (StrAFs) have originally been proposed in [5]. Contrary to this work, in this paper we focused on admissibility-based semantics. We showed that the weak admissibility-based semantics defined in the original work satisfy some expected properties, namely Dung's Fundamental Lemma. However, the definition for strong admissibility proposed in [5] does not yield semantics that behave as expected. This has conducted us to revisit the definition of strong admissibility, and this allowed us to introduce strong complete and preferred semantics. We have also enhanced the StrAFs literature by studying the computational complexity of classical reasoning problems for these semantics, and we have shown that it is the same as for the corresponding tasks in Dung's framework, in spite of the increase of expressivity. Then we have proposed a method based on pseudo-Boolean constraints for computing the extensions of a StrAF under the various semantics defined in this paper, and we have empirically evaluated the scalability of this approach for the new semantics defined in this paper, as well as the (weak and strong) stable semantics from [5]. Future work include the study of (weak and strong) grounded semantics, and tight complexity results for the skeptical reasoning under the (weak and strong) complete semantics. We also plan to analyze the relation between StrAFs and other frameworks, in particular the comparison of the signatures of StrAFs semantics and SETAFs semantics [20,21,22]. Finally, we want to study argument strength and accrual in a context of structured argumentation.

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artif Intell. 1995;77(2):321-58.

[2] Cayrol C, Lagasquie-Schiex M. Bipolarity in argumentation graphs: Towards a better understanding. Int J Approx Reason. 2013;54(7):876-99.

[3] Amgoud L, Cayrol C. A Reasoning Model Based on the Production of Acceptable Arguments. Annals of Mathematics and Artificial Intelligence. 2002;34(1-3):197-215.

[4] Bench-Capon TJM. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. J Log Comput. 2003;13(3):429-48.

[5] Rossit J, Mailly JG, Dimopoulos Y, Moraitis P. United We Stand: Accruals in Strength-based Argumentation. Argument Comput. 2021;12(1):87-113.

[6] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: Handbook of Formal Argumentation. College Publications; 2018. p. 159-236.

[7] Dvorák W, Dunne PE. Computational Problems in Formal Argumentation and their Complexity. In: Handbook of Formal Argumentation. College Publications; 2018. p. 631-88.

[8] Arora S, Barak B. Computational Complexity - A Modern Approach. Cambridge UP; 2009.

[9] Roussel O, Manquinho VM. Pseudo-Boolean and Cardinality Constraints. In: Handbook of Satisfiability; 2009. p. 695-733.

[10] Martins R, Manquinho VM, Lynce I. Open-WBO: A Modular MaxSAT Solver,. In: Proc. of SAT'14; 2014. p. 438-45.

[11] Elffers J, Nordström J. Divide and Conquer: Towards Faster Pseudo-Boolean Solving. In: Proc. of IJCAI'18; 2018. p. 1291-9.

[12] Le Berre D, Parrain A. The Sat4j library, release 2.2. J Satisf Boolean Model Comput. 2010;7(2-3):59-6.

[13] Bacquey Y, Mailly JG, Moraitis P, Rossit J. Admissibility in Strength-based Argumentation: Complexity and Algorithms (Extended Version with Proofs). arXiv; 2022. Available from: https://arxiv.org/abs/2207.02258.

[14] Dvorák W, Gaggl SA, Rapberger A, Wallner JP, Woltran S. The ASPARTIX System Suite. In: Proc. of COMMA'20; 2020. p. 461-2.

[15] Erdös P, Rényi A. On Random Graphs. I. Publicationes Mathematicae. 1959;6:290-7.

[16] Barabasi AL, Albert R. Emergence of Scaling in Random Networks. Science. 1999;286(5439):509-12.

[17] Gaggl SA, Linsbichler T, Maratea M, Woltran S. Design and results of the Second International Competition on Computational Models of Argumentation. Artif Intell. 2020;279.

[18] Lagniez JM, Lonca E, Mailly JG, Rossit J. Introducing the Fourth International Competition on Computational Models of Argumentation. In: Proc. of SAFA'20; 2020. p. 80-5.

[19] Gning S, Mailly JG. On the Impact of SAT Solvers on Argumentation Solvers. In: Proc. of SAFA'20. vol. 2672; 2020. p. 68-73.

[20] Nielsen SH, Parsons S. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In: Proc. of ArgMAS'06. Springer; 2006. p. 54-73.

[21] Dvorák W, Fandinno J, Woltran S. On the expressive power of collective attacks. Argument Comput. 2019;10(2):191-230.

[22] Flouris G, Bikakis A. A comprehensive study of argumentation frameworks with sets of attacking arguments. Int J Approx Reason. 2019;109:55-86.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220141

A Generalized Notion of Consistency with Applications to Formal Argumentation

Pietro BARONI [a,1], Federico CERUTTI [a] and Massimiliano GIACOMIN [a]
[a] *DII, University of Brescia, Italy*
ORCiD ID: Pietro Baroni https://orcid.org/0000-0001-5439-9561, Federico Cerutti https://orcid.org/0000-0003-0755-0358, Massimiliano Giacomin https://orcid.org/0000-0003-4771-4265

Abstract. We propose a generic notion of consistency in an abstract labelling setting, based on two relations: one of intolerance between the labelled elements and one of incompatibility between the labels assigned to them, thus allowing a spectrum of consistency requirements depending on the actual choice of these relations. As a first application to formal argumentation, we show that traditional Dung's semantics can be put in correspondence with different consistency requirements in this context. We consider then the issue of consistency preservation when a labelling is obtained as a synthesis of a set of labellings, as is the case for the traditional notion of argument justification. In this context we provide a general characterization of consistency-preserving synthesis functions and analyze the case of argument justification in this respect.

Keywords. Consistency, Argumentation semantics, Argument justification

1. Introduction

In formal argumentation, the presence of conflicts between arguments is a key aspect that calls for mechanisms able to produce sensible reasoning outcomes. In particular, it is typically required that these outcomes satisfy properties which have intuitively to do with the notion of consistency. For instance, in abstract argumentation semantics [1,2] either extensions or labellings are typically required to satisfy the property of conflict-freeness, while, moving from abstract to structured argumentation, it is desired that the conclusions of arguments regarded as acceptable are not contradictory, as indicated by the properties of direct and indirect consistency in [3]. While consistency appears to permeate the field of formal argumentation as a crucial component, to our knowledge no attempts are available in the literature to provide a general formal treatment of this notion, consistency-related definitions being usually embedded in the context of specific formalisms, without a common reference framework. This appears to be a limitation regarding the possibility of bridging together the consistency notions considered in different formalisms and possibly investigating variations and developments thereof.

To fill this gap, in this paper we introduce a generalized notion of (in)consistency applicable in any context where a labelling approach is adopted. The proposed notion re-

[1] Corresponding Author: Pietro Baroni, DII, University of Brescia, Italy; E-mail: pietro.baroni@unibs.it

lies on two basic elements: an intolerance relation between the labelled elements and an incompatibility relation between the labels, as presented in Section 2. As a first example of the application of the proposed concept, we show in Section 3 that Dung's traditional semantics can be put in correspondence with different consistency requirements, particularly with different incompatibility relations. As a further step, in Section 4, we consider the issue of consistency preservation when a labelling derives from a set of other labellings. We provide some results concerning consistency preservation when a labelling is obtained through a synthesis function and apply these concepts to the case of deriving the argument justification status. The relationships of this work with previous literature and various perspectives of future development are finally discussed in Section 5.

2. Generalizing consistency for labelling-based systems

In a variety of contexts the assessments of entities of various kind are expressed by assigning them a label taken from a predefined set. In many cases these sets of labels have an intuitive underlying order according to some notion of *positivity*. In order to provide a common ground to characterize different assessment labels and to relate and compare them, we first introduce the notion of assessment classes.

Definition 1 *A set of assessment classes is a set C equipped with a total order $\leq$ and including a maximum and a minimum element, which are assumed to be distinct.*

In the following we will abbreviate the term 'set(s) of assessment classes' as sac(s). Intuitively, the order is meant to capture an abstract distinction between different levels of positivity of the assessment, with $c_1 \leq c_2$ meaning that c_2 corresponds to an at least as positive assessment as c_1 (whatever a positive assessment means in a given context). In the following we will mostly use a tripolar sac $C^3 = \{\mathsf{pos}, \mathsf{mid}, \mathsf{neg}\}$ with $\mathsf{neg} \leq \mathsf{mid} \leq \mathsf{pos}$ and the intuitive meaning that pos corresponds to a definitely positive assessment, neg to a definitely negative assessment, and mid to an intermediate situation. The basic idea, expressed by the following definition, is that a sac is used to classify the elements of a set of labels according to their level of positivity. Note that the elements of a sac are called classes because in general more than one label can be mapped to the same class.

Definition 2 *Given a set of assessment classes C, a C-classified set of assessment labels is a set Λ equipped with a total function $C_\Lambda : \Lambda \to C$. The total preorder induced on Λ by C_Λ will be denoted by $\preceq$ where $\lambda_1 \preceq \lambda_2$ iff $C_\Lambda(\lambda_1) \leq C_\Lambda(\lambda_2)$. As usual, $\lambda_1 \prec \lambda_2$ will denote $\lambda_1 \preceq \lambda_2$ and $\lambda_2 \npreceq \lambda_1$*

We will abbreviate the term 'set(s) of assessment labels' as sal(s) and omit 'C-classified', when C is not ambiguous. Also, to distinguish preorders referring to different sals, given a sal Λ we will denote the relevant preorder as $\preceq_\Lambda$.

The notion of labelling based on a sal is the usual one.

Definition 3 *Given a sal Λ and a set S a Λ-labelling of S is a function $L : S \to \Lambda$.*

Different sals can be used to express assessments in distinct, but possibly related, evaluation contexts. For instance, in the context of argument acceptance evaluation based on the labelling-based version of Dung's semantics [1,2], the sal $\Lambda^{\mathrm{IOU}} =$

$\{\mathsf{in}, \mathsf{out}, \mathsf{und}\}$ is used, while in Defeasible Logic Programming (*DeLP*) arguments are marked as D(efeated) or U(ndefeated) corresponding to the use of the sal $\Lambda^{\mathrm{De}} = \{\mathsf{D}, \mathsf{U}\}$, and in [4] an approach using the set of four labels $\Lambda^{\mathrm{JV}} = \{+, -, \pm, \emptyset\}$ is proposed. We assume that the sals mentioned above are C^3-classified as follows: $C^3_{\Lambda\mathrm{IOU}} = \{(\mathsf{in}, \mathsf{pos}), (\mathsf{out}, \mathsf{neg}), (\mathsf{und}, \mathsf{mid})\}$; $C^3_{\Lambda\mathrm{De}} = \{(\mathsf{D}, \mathsf{neg}), (\mathsf{U}, \mathsf{pos})\}$; $C^3_{\Lambda\mathrm{JV}} = \{(-, \mathsf{neg}), (+, \mathsf{pos}), (\pm, \mathsf{mid}), (\emptyset, \mathsf{mid})\}$.

We can now introduce a generalized notion of inconsistency in this formal context. Intuitively, an inconsistency arises when two elements of a set which cannot stand each other are assigned labels which are 'too positive' altogether.

This suggests that, in general terms, inconsistency can be understood as arising from two components: an intolerance relation at the level of the assessed elements, indicating who cannot stand whom, and an incompatibility relation at the level of the labels, indicating which pairs of positive assessments correspond to a clash if ascribed to a pair of elements connected by the intolerance relation. In the following we will assume that an incompatibility relation on assessment labels is always induced by an incompatibility relation on assessment classes.

Definition 4 *Given a set S, an intolerance relation on S is a binary relation int $\subseteq S \times S$, where $(s_1, s_2) \in$ int indicates that s_1 is intolerant of s_2 and will be denoted as $s_1 \odot s_2$, while $(s_1, s_2) \notin$ int will be denoted as $s_1 \ominus s_2$.*

Note that we do not make any assumption on the intolerance relation, in particular it needs not to be symmetric.

To exemplify, in languages equipped with negation, typically intolerance between language elements coincides with negation (a symmetric relation where each element has exactly one opposite), however more general forms of contrariness have been considered in argumentation contexts, where the corresponding intolerance relation may not be symmetric and allows the existence of multiple contraries for an element [5,6]. At the argument level, the attack relation in Dung's frameworks can be regarded as an example of intolerance relation.

Definition 5 *Given a sac C, an incompatibility relation on C is a relation inc $\subseteq C \times C$, where $(c_1, c_2) \in$ inc indicates that c_1 is incompatible with c_2 and will be denoted as $c_1 \boxdot c_2$, while $(c_1, c_2) \notin$ inc will be denoted as $c_1 \boxminus c_2$. Given a C-classified sal Λ, we define the induced incompatibility relation inc$' \subseteq \Lambda \times \Lambda$ as follows: for every $\lambda_1, \lambda_2 \in \Lambda$, $(\lambda_1, \lambda_2) \in$ inc$'$ iff $(C_\Lambda(\lambda_1), C_\Lambda(\lambda_2)) \in$ inc. With a little abuse of notation we will also denote $(\lambda_1, \lambda_2) \in$ inc$'$ as $\lambda_1 \boxdot \lambda_2$, and analogously for $\lambda_1 \boxminus \lambda_2$. Given a label λ, we define the set of labels which are compatible with λ as sc$(\lambda) \triangleq \{\lambda' \in \Lambda \mid (\lambda, \lambda') \notin$ inc$'\}$.*

We remark again that incompatibility refers to the situation where labels are assigned to entities which are linked by intolerance. For intance, in a context where statements are assessed and intolerance between them corresponds to contradiction, two (not necessarily distinct) positive labels expressing belief should be incompatible: they cannot be assigned to two contradictory statements, since you cannot believe both of them.

We can now introduce our generalized notion of inconsistency of a labelling.

Definition 6 *Given a set S equipped with an intolerance relation int, a sac C equipped with an incompatibility relation inc, and a C-classified sal Λ, a Λ-labelling L of S is int-inc-inconsistent iff $\exists s_1, s_2 \in S$ such that $s_1 \odot s_2$ and $L(s_1) \boxdot L(s_2)$.*

We say that a labelling is int-inc-consistent if it is not int-inc-inconsistent and that a set $\mathscr{L}$ of labellings is int-inc-consistent if every $L \in \mathscr{L}$ is int-inc-consistent.

From the intuition underlying Definition 6, some rather natural properties can be identified for an incompatibility relation on a sac C, based on the idea that inconsistency arises from a sort of 'excess of simultaneous positiveness' in the assessment of some elements linked by intolerance. First, an obvious requirement is that $\max(C) \boxdot \max(C)$: two maximally positive labels cannot be ascribed together to conflicting elements. More generally one can observe that if $c_1 \boxdot c_2$, then for every pair c_1', c_2' such that $c_1 \leq c_1'$ and $c_2 \leq c_2'$ it must hold that $c_1' \boxdot c_2'$, since the simultaneous positiveness expressed by c_1' and c_2' is not lesser that the one expressed by c_1 and c_2. We call such an incompatibility relation *monotonic* and take this property for granted in the following.

Note that $\max(C) \boxdot \max(C)$ is a consequence of the monotonicity property if one assumes that inc is not empty. Accordingly we define, for any sac C, the minimal nonempty incompatibility relation as $\underline{\text{inc}}_C = \{(\max(C), \max(C))\}$.

It also follows that, to avoid a degenerate situation where every labelling is inconsistent, it must hold that $\min(C) \boxminus \min(C)$.

Moreover, assuming that the intolerance relation is not empty, for $\max(C)$ to be attainable for every element without necessarily generating inconsistencies it must be the case that the following stronger condition (implying the previous one) holds: $\max(C) \boxminus \min(C)$ and $\min(C) \boxminus \max(C)$ or equivalently $\nexists c \in C$ such that $c \boxdot \min(C)$ or $\min(C) \boxdot c$. Note that this implies that for any C-classified sal Λ, $sc(\lambda) \neq \emptyset$ for any $\lambda \in \Lambda$ under the mild condition that $\exists \lambda \in \Lambda : C_\Lambda(\lambda) = \min(C)$, which we will assume in the following.

The generic definition of inconsistency we have introduced is 'tunable' as its instances can be 'adjusted' varying the incompatibility relation, and possibly also the underlying intolerance relation and C-classification, giving rise to a family of alternative (in)consistency notions. In particular, different argumentation semantics can be put in correspondence with different (in)consistency notions, as discussed next.

3. Consistency properties in argumentation semantics

As well-known, in abstract argumentation an argumentation semantics is a formal specification of a criterion to determine the possible outcomes of a situation of conflict, represented by a binary relation of attack (denoted as $\rightarrow$ in the following), between a set $\mathscr{A}$ of arguments, as expressed by the traditional notion of argumentation framework [1].

Definition 7 *An argumentation framework is a pair $AF = (\mathscr{A}, \rightarrow)$ where $\mathscr{A}$ is a set of arguments and $\rightarrow \subseteq \mathscr{A} \times \mathscr{A}$ is a binary relation of attack between them.*

In the *extension-based* approach to argumentation semantics the conflict outcomes are expressed as sets of arguments called *extensions* and, in this context, a basic consistency notion called *conflict-freeness* has been traditionally considered: a set of arguments is conflict-free if it does not include any pair of arguments α, β such that $(\alpha, \beta) \in \rightarrow$ (also denoted as $\alpha \in \beta^-$). In the *labelling-based* approach to argumentation semantics, the outcomes are expressed as arguments labellings, i.e. as assignments of labels, taken from a given set, to the set of arguments $\mathscr{A}$. Using the set of three labels Λ^{IOU} a correspondence can be drawn between extensions and labellings, while in general the labelling-based approach is more expressive than the extension-based approach. Combining the generalized notion of consistency with three-valued labellings enables to identify correspondences between different notions of consistency and different semantics. In particular, given an abstract argumentation framework, we naturally assume that the intolerance relation coincides with the attack relation, i.e. $\alpha \odot \beta$ iff $\alpha \in \beta^-$, and use the classification $C^3_{\Lambda\mathrm{IOU}}$ introduced above. Then, an analysis of labelling-based semantics in this perspective can be developed, as we do in the following, where we review the definitions of some fundamental labelling-based semantics [2] and analyze their generalized consistency properties.

The simplest semantics notion is the one of conflict-freeness, which is recalled in Definition 8.

Definition 8 *Let L be a labelling of an argumentation framework $AF = (\mathscr{A}, \rightarrow)$. L is* conflict-free *iff for each $\alpha \in \mathscr{A}$ it holds that:*

1. *if $L(\alpha) = \mathtt{in}$ then $\nexists \beta \in \alpha^- : L(\beta) = \mathtt{in}$*
2. *if $L(\alpha) = \mathtt{out}$ then $\exists \beta \in \alpha^- : L(\beta) = \mathtt{in}$*

Item 1 in Definition 8 corresponds exactly to the weakest form of consistency, i.e. to the incompatibility relation $\underline{\mathrm{inc}}_{C^3} = \{(\mathsf{pos}, \mathsf{pos})\}$.

Admissibility of a set of arguments was introduced in [1] with reference to the notion of defense, i.e. the ability of a conflict-free set to defend its members by counterattacking their attackers. The labelling-based counterpart of this idea is given in Definition 9.

Definition 9 *Let L be a labelling of an argumentation framework $AF = (\mathscr{A}, \rightarrow)$. L is* admissible *iff for each $\alpha \in \mathscr{A}$ it holds that:*

1. *if $L(\alpha) = \mathtt{in}$ then $\forall \beta \in \alpha^- : L(\beta) = \mathtt{out}$*
2. *if $L(\alpha) = \mathtt{out}$ then $\exists \beta \in \alpha^- : L(\beta) = \mathtt{in}$*

Item 1 in Definition 9 is a strengthening of item 1 of Definition 8, while item 2 is the same in both Definition 8 and 9. Interestingly, this strengthening corresponds to the choice of a stronger form of consistency: having an attacker labelled und is forbidden for an argument labelled in, while having an attacker labelled in is allowed for an argument labelled und. This coincides with adopting the following asymmetric incompatibility relation $\mathrm{inc}^a_{C^3} = \{(\mathsf{pos}, \mathsf{pos}), (\mathsf{mid}, \mathsf{pos})\}$.

Proposition 1 *The set of admissible labellings coincides with the set of conflict-free labellings which are $\rightarrow$-$\mathrm{inc}^a_{C^3}$-consistent.*

Proof: For a labelling L let us first assume that L is admissible. Then L is conflict-free and by item 1 of Definition 9 $\nexists \alpha, \beta \in \mathscr{A}$ such that $\beta \in \alpha^-$ (i.e. $\beta \odot \alpha$) and $(L(\beta), L(\alpha)) \in \mathrm{inc}^a_{C^3}$ (i.e. $L(\beta) \boxdot L(\alpha)$). Hence L is $\rightarrow$-$\mathrm{inc}^a_{C^3}$-consistent. Let now assume L is conflict-free and $\rightarrow$-$\mathrm{inc}^a_{C^3}$-consistent. To complete the proof we have to show that item 1 of Definition 9 holds: assume by contradiction that $\exists \alpha$ such that $L(\alpha) = \mathtt{in}$ and $\exists \beta \in \alpha^- : L(\beta) \neq \mathtt{out}$. It follows that $(L(\beta), L(\alpha)) \in \mathrm{inc}^a_{C^3}$ which contradicts the hypotesis that L is $\rightarrow$-$\mathrm{inc}^a_{C^3}$-consistent. □

Completeness of a set of arguments was introduced in [1] and is based on the idea that if an argument is defended by an admissible set of arguments, it should be accepted together with its defenders. The labelling-based counterpart of this idea is given in Definition 10.

Definition 10 *Let L be a labelling of an argumentation framework $AF = (\mathscr{A}, \rightarrow)$. L is* complete *if it is admissible and for each $\alpha \in \mathscr{A}$ it holds that if $L(\alpha) = \texttt{und}$ then $\nexists \beta \in \alpha^- : L(\beta) = \texttt{in}$ and $\exists \beta \in \alpha^- : L(\beta) = \texttt{und}$*

In words a complete labelling is an admissible labelling with the additional requirement that an argument which is labelled und must have an und-labelled attacker and no in-labelled attackers. This amounts to further strengthening the notion of consistency by adopting the incompatibility relation $\text{inc}^c_{C3} = \{(\mathsf{pos}, \mathsf{pos}), (\mathsf{pos}, \mathsf{mid}), (\mathsf{mid}, \mathsf{pos})\}$ together with enforcing the following reinstatement property.

Definition 11 *A labelling L satisfies the reinstatement property if $\forall \alpha \in \mathscr{A}$ it holds that if $\forall \beta \in \alpha^-$ $L(\beta) = \texttt{out}$ then $L(\beta) = \texttt{in}$*

Proposition 2 *The set of complete labellings coincides with the set of admissible labellings which are $\rightarrow$-inc^c_{C3}-consistent and satisfy the reinstatement property.*

Proof: For a labelling L let us first assume that L is complete, hence admissible. From Proposition 1 we have that $\nexists \alpha, \beta$ such that $\beta \in \alpha^-$ and $(L(\beta), L(\alpha)) \in \{(\mathsf{pos}, \mathsf{pos}), (\mathsf{mid}, \mathsf{pos})\}$. From Definition 10 we have also that if $L(\alpha) = \texttt{und}$ then $\nexists \beta \in \alpha^- : L(\beta) = \texttt{in}$, i.e. $\nexists \alpha, \beta$ such that $\beta \in \alpha^-$ and $(L(\beta), L(\alpha)) \in \{(\mathsf{pos}, \mathsf{mid})\}$. It follows that L is $\rightarrow$-inc^c_{C3}-consistent. Moreover, it is well known that complete labellings satisfy the reinstatement property [2]. Let us now assume L is admissible, $\rightarrow$-inc^c_{C3}-consistent and satisfies the reinstatement property. Given an argument α such that $L(\alpha) = \texttt{und}$ it follows (from consistency) that $\nexists \beta \in \alpha^- : L(\beta) = \texttt{in}$ and (from the reinstatement property) that $\exists \beta \in \alpha^- : L(\beta) \neq \texttt{out}$, hence $\exists \beta \in \alpha^- : L(\beta) = \texttt{und}$ and L is a complete labelling. □

Stability of a set of arguments can be characterized in several ways, its key feature being that no room is left for undecidedness (an argument is either accepted or attacked by an accepted argument) as indicated by Definition 12.

Definition 12 *Let L be a labelling of an argumentation framework $AF = (\mathscr{A}, \rightarrow)$. L is* stable *if it is complete and $\nexists \alpha \in \mathscr{A} : L(\alpha) = \texttt{und}$.*

This constraint can be put in correspondence with the adoption of the strongest notion of consistency, namely with the choice of the incompatibility relation $\overline{\text{inc}}_{C3} = \{(\mathsf{pos}, \mathsf{pos}), (\mathsf{pos}, \mathsf{mid}), (\mathsf{mid}, \mathsf{pos}), (\mathsf{mid}, \mathsf{mid})\}$.

Proposition 3 *The set of stable labellings coincides with the set of complete labellings which are $\rightarrow$-$\overline{\text{inc}}_{C3}$-consistent.*

Proof: For a labelling L let us first assume that L is stable. It follows that no argument is labelled und hence $\nexists \alpha, \beta$ such that $\beta \in \alpha^-$ and $(L(\beta), L(\alpha)) \in \{(\mathsf{pos}, \mathsf{mid}), (\mathsf{mid}, \mathsf{pos}), (\mathsf{mid}, \mathsf{mid})\}$ and from conflict-freeness we have also that $\nexists \alpha, \beta$ such that $\beta \in \alpha^-$ and $(L(\beta), L(\alpha)) = (\mathsf{pos}, \mathsf{pos})$. Therefore L is $\rightarrow$-$\overline{\text{inc}}_{C3}$-consistent. Assume now L is complete and $\rightarrow$-$\overline{\text{inc}}_{C3}$-consistent and suppose by contradiction that $\exists \alpha$ such that $L(\alpha) = \texttt{und}$. It follows that $\alpha^- \neq \emptyset$, otherwise by the reinstatement property it would hold that $L(\alpha) = \texttt{in}$. For every $\beta \in \alpha^-$ we have that $L(\beta) \notin \{\texttt{in}, \texttt{und}\}$ otherwise L would not be

$\rightarrow$-$\overline{\text{inc}}_{C^3}$-consistent. But then $\forall \beta \in \alpha^-$ we get $L(\beta) = \texttt{out}$ which, by the reinstatement property, contradicts $L(\alpha) = \texttt{und}$. □

To summarize, admissible labellings can be characterized in terms of strengthening consistency with respect to conflict-freeness without resorting to the traditional notion of defense, while further strengthenings of consistency, together with the reinstatement property, characterize complete and stable labellings.

In the next section we move beyond the evaluation of acceptability of arguments carried out on the basis of argumentation semantics and consider further evaluations that can be derived from it and raise the issue of preserving consistency across the derivation.

4. Consistency preservation in labelling derivation mechanisms

The outcomes prescribed by an argumentation semantics are typically used as the starting point for the derivation of further evaluations, for instance whether an argument is skeptically justified. It is then interesting to consider the question of whether and how the consistency properties of the original evaluation are preserved in the derived evaluation and of the requirements that can be posed on the derivation mechanism to ensure this preservation.

We focus here on what we call pure synthesis labelling derivation, namely a mechanism where a labelling of a set S is generated from a set of labellings of the same set S. To exemplify, the evaluation of the argument justification status according to a given semantics is derived from the set of the argument extensions/labellings prescribed by the same semantics.

The simplest notion of argument justification, which we will use as running example, is based on three possible states.

Definition 13 *Given a set $\mathscr{L}$ of Λ^{IOU}-labellings of a set of arguments $\mathscr{A}$, an argument $\alpha \in \mathscr{A}$ is:*

- *skeptically justified iff $\forall L \in \mathscr{L}$ $L(\alpha) = \texttt{in}$;*
- *credulously justified iff it is not skeptically justified*[2] *and $\exists L \in \mathscr{L} : L(\alpha) = \texttt{in}$;*
- *not justified iff $\nexists L \in \mathscr{L} : L(\alpha) = \texttt{in}$*

Considering a sal $\Lambda^{\text{AJ}} = \{\mathsf{SkJ}, \mathsf{CrJ}, \mathsf{NoJ}\}$, the evaluation of argument justification can be modelled as the generation of a Λ^{AJ}-labelling from a set of Λ^{IOU}-labellings. Concerning Λ^{AJ} it is intuitive to assume the classification $C^3_{\Lambda^{\text{AJ}}} = \{(\mathsf{SkJ}, \mathsf{pos}), (\mathsf{NoJ}, \mathsf{neg}), (\mathsf{CrJ}, \mathsf{mid})\}$.

At a general level, pure synthesis labelling derivations, like the one of argument justification, can be formalized through a simple synthesis function.

Definition 14 *Given two sets of labels Λ_1 and Λ_2, a simple synthesis function (ssf) from Λ_1 to Λ_2 is a mapping* $\mathsf{syn} : 2^{\Lambda_1} \setminus \{\emptyset\} \rightarrow \Lambda_2$.

The idea is that given a set of Λ_1-labellings of a set S a Λ_2-labelling of S can be derived by applying a ssf to the set of labels relevant to each element of S.

[2] Traditionally credulous justification id regarded as including skeptical justification, we enforce this distinction so that argument justification can be properly modelled as a labelling.

Definition 15 *Let S be a set, Λ_1 and Λ_2 sets of labels,* syn *a ssf from Λ_1 to Λ_2, and $\mathscr{L}_1$ a set of Λ_1-labellings of S. The Λ_2-labelling L_2 derived from $\mathscr{L}_1$ through* syn *is denoted as $DL^{\mathsf{syn}}_{\mathscr{L}_1}$ defined, for every $s \in S$ as:*

$$DL^{\mathsf{syn}}_{\mathscr{L}_1}(s) = \mathsf{syn}(\{L_1(s) \mid L_1 \in \mathscr{L}_1\})$$

To exemplify, it is easy to see that the argument justification evaluation described above corresponds to the use of a ssf $\mathsf{syn}_{\mathrm{AJ}}$ from Λ^{IOU} to Λ^{AJ} defined, for every $\Lambda \subseteq \Lambda^{\mathrm{IOU}}$ as follows:

- $\mathsf{syn}_{\mathrm{AJ}}(\Lambda) = \mathsf{SkJ}$ if $\Lambda = \{\mathtt{in}\}$;
- $\mathsf{syn}_{\mathrm{AJ}}(\Lambda) = \mathsf{CrJ}$ if $\Lambda \supsetneq \{\mathtt{in}\}$;
- $\mathsf{syn}_{\mathrm{AJ}}(\Lambda) = \mathsf{NoJ}$ otherwise.

Assuming that the labellings used for derivation satisfy some consistency properties, a preservation of these properties in the derived labelling appears to be desirable.

Definition 16 *Let C be a sac equipped with an incompatibility relation inc, and Λ_1 and Λ_2 be two C-classifed sets of labels. A ssf* syn *from Λ_1 to Λ_2 is* consistency preserving *iff for any set S equipped with an intolerance relation int and any int-inc-consistent set $\mathscr{L}_1$ of Λ_1-labellings of S it holds that the labelling $DL^{\mathsf{syn}}_{\mathscr{L}_1}$ is int-inc-consistent.*

This in turn raises the issue of analyzing at a general level some properties of the ssf that can ensure consistency preservation.

To start, we introduce a notion of well-behaved ssf which intuitively means that the function is monotonic with respect to some positiveness ordering of sets of labels, introduced in next definition.

Definition 17 *Given a sal Λ, and $\Lambda_1, \Lambda_2 \subseteq \Lambda$, we say that Λ_2 is at least as positive as Λ_1, denoted as $\Lambda_1 \preceq_P \Lambda_2$, iff $\forall \lambda \in \Lambda_1\ \exists \lambda' \in \Lambda_2$ such that $\lambda \preceq_\Lambda \lambda'$ and $\forall \lambda' \in \Lambda_2\ \exists \lambda \in \Lambda_1$ such that $\lambda \preceq_\Lambda \lambda'$.*

The idea of the $\preceq_P$ relation is that every element of Λ_1 can be mapped into an at least as positive element of Λ_2 and at the same time every element of Λ_2 can be mapped into a no more positive element of Λ_1. $\preceq_P$ is reflexive and transitive, i.e. a preorder. To exemplify, $\forall \emptyset \subsetneq \Lambda \subseteq \Lambda^{\mathrm{IOU}}$ it holds that $\Lambda \preceq_P \{\mathtt{in}\}$ and $\{\mathtt{out}\} \preceq_P \Lambda$. Also $\{\mathtt{in},\mathtt{out}\} \preceq_P \{\mathtt{in},\mathtt{und},\mathtt{out}\}$ and $\{\mathtt{in},\mathtt{und},\mathtt{out}\} \preceq_P \{\mathtt{in},\mathtt{out}\}$ while $\{\mathtt{in},\mathtt{out}\} \not\preceq_P \{\mathtt{und}\}$ and $\{\mathtt{und}\} \not\preceq_P \{\mathtt{in},\mathtt{out}\}$,

We can now introduce the notion of well-behaved ssf.

Definition 18 *A ssf* syn *is well-behaved iff whenever $\Lambda_1 \preceq_P \Lambda_2$* $\mathsf{syn}(\Lambda_1) \preceq \mathsf{syn}(\Lambda_2)$.

We then move to consider, given a set of labels Λ_1, whether a set of labels Λ_2 is a compatible dual of Λ_1, meaning that, given an int-inc-consistent set of labellings $\mathscr{L}_1$, if $\Lambda_1 = \{L_1(s) \mid L_1 \in \mathscr{L}_1\}$ for some element s, then it is possible that $\Lambda_2 = \{L_1(s') \mid L_1 \in \mathscr{L}_1\}$ for some s' such that $s \odot s'$.

Definition 19 *Given a sal Λ, and $\Lambda_1 \subseteq \Lambda$, we say that $\Lambda_2 \subseteq \Lambda$ is a compatible dual of Λ_1, denoted as $\Lambda_2 \in CD(\Lambda_1)$, iff $\forall \lambda \in \Lambda_1\ \exists \lambda' \in \Lambda_2$ such that $\lambda' \in sc(\lambda)$, and $\forall \lambda' \in \Lambda_2$ $\exists \lambda \in \Lambda_1$ such that $\lambda' \in sc(\lambda)$.*

Proposition 4 *Let C be a sac equipped with an incompatibility relation inc and Λ be a C-classifed set of labels. For any set S equipped with an intolerance relation int, any int-inc-consistent set $\mathscr{L}_1$ of Λ-labellings of S, and any $s_1, s_2 \in S$ such that $(s_1, s_2) \in int$, it holds that $\mathscr{L}_1^{\downarrow}(s_2) \in CD(\mathscr{L}_1^{\downarrow}(s_1))$ where for any set $\mathscr{L}$ of Λ-labellings of S and any $s \in S$, $\mathscr{L}^{\downarrow}(s) \triangleq \{L(s) \mid L \in \mathscr{L}\}$.*

Proof: Since every $L \in \mathscr{L}_1$ is int-inc-consistent it must be the case that $L(s_2) \in sc(L(s_1))$, hence $\forall \lambda \in \mathscr{L}_1^{\downarrow}(s_1)\ \exists \lambda' \in \mathscr{L}_1^{\downarrow}(s_2)$ such that $\lambda' \in sc(\lambda)$ and also $\forall \lambda' \in \mathscr{L}_1^{\downarrow}(s_2)\ \exists \lambda \in \mathscr{L}_1^{\downarrow}(s_1)$ such that $\lambda' \in sc(\lambda)$, hence $\mathscr{L}_1^{\downarrow}(s_2) \in CD(\mathscr{L}_1^{\downarrow}(s_1))$. □

Towards characterizing well-behaved ssfs which are consistency preserving we focus on the case where the set of labellings to be synthesized is finite, which is common in argumentation semantics and many other kinds of assessments. The case of infinite sets of labellings is left to future work.

To start, we need to consider a compatible dual of a finite set of labels which turns out to be not less positive than any other compatible dual.

Definition 20 *Given a sal Λ, and a finite $\Lambda_1 \subseteq \Lambda$ we define $MCD(\Lambda_1) \triangleq \bigcup_{\lambda \in \Lambda_1} \widehat{MC}(\lambda)$, where $\widehat{MC}(\lambda) \triangleq \{\lambda' \in sc(\lambda) \mid \nexists \lambda'' \in sc(\lambda) : \lambda' \prec \lambda''\}$*

Note that non emptyness of $MCD(\Lambda_1)$ follows from the non emptyness of $sc(\lambda)$ for every λ (Section 2) and from the finiteness of Λ_1 together with the total ordering of C. The following propositions, which assume again Λ_1 finite, provide two interesting properties of $MCD(\Lambda_1)$: it belongs to $CD(\Lambda_1)$ and is maximal with respect to $\preceq_P$.

Proposition 5 $MCD(\Lambda_1) \in CD(\Lambda_1)$.

Proof: From the definition it is immediate to see that for every $\lambda \in \Lambda_1\ \exists \lambda' \in MCD(\Lambda_1)$ such that $\lambda' \in sc(\lambda)$ and that for every $\lambda' \in MCD(\Lambda_1)\ \exists \lambda \in \Lambda_1$ such that $\lambda' \in sc(\lambda)$. □

Proposition 6 $\forall D \in CD(\Lambda_1)$ *it holds that* $D \preceq_P MCD(\Lambda_1)$.

Proof: For any $\lambda' \in D$, from Definition 19 it holds that $\exists \lambda \in \Lambda_1$ such that $\lambda' \in sc(\lambda)$. Then, by Definition 20 $\exists \lambda'' \in MCD(\Lambda_1)$ such that $\lambda'' \in sc(\lambda)$ and $\nexists \lambda''' \in sc(\lambda) : \lambda'' \prec \lambda'''$ which implies that $\lambda' \preceq \lambda''$. Consider now any $\lambda'' \in MCD(\Lambda_1)$: by Definition 20 it holds that $\exists \lambda \in \Lambda_1$ such that $\lambda'' \in sc(\lambda)$. Moreover by Definition 19 $\exists \lambda' \in D$ such that $\lambda' \in sc(\lambda)$. Now by Definition 20 we have again that $\nexists \lambda''' \in sc(\lambda) : \lambda'' \prec \lambda'''$ and hence $\lambda' \preceq \lambda''$. □

On this basis, we can now derive a necessary and sufficient condition for a well-behaved ssf to be consistency preserving for finite sets of labels.

Proposition 7 *A well-behaved ssf* syn *is consistency preserving if and only if for every finite set $\Lambda_1 \subseteq \Lambda$ it holds that* $\mathsf{syn}(MCD(\Lambda_1)) \in sc(\mathsf{syn}(\Lambda_1))$.

Proof: Let syn be a ssf satisfying the hypotheses and assume by contradiction that syn is not consistency preserving. This means that there are two elements $s_1, s_2 \in S$ such that $s_1 \odot s_2$ and a set $\mathscr{L}_1$ of Λ-labellings of S such that $DL_{\mathscr{L}_1}^{\mathsf{syn}}(s_1) \boxdot DL_{\mathscr{L}_1}^{\mathsf{syn}}(s_2)$.

Now $DL^{\mathsf{syn}}_{\mathscr{L}_1}(s_1) = \mathsf{syn}(\mathscr{L}_1^{\downarrow}(s_1))$ and similarly $DL^{\mathsf{syn}}_{\mathscr{L}_1}(s_2) = \mathsf{syn}(\mathscr{L}_1^{\downarrow}(s_2))$. Let $\Lambda_1 = \mathsf{syn}(\mathscr{L}_1^{\downarrow}(s_1))$. From Proposition 4 we have $\mathscr{L}_1^{\downarrow}(s_2) \in CD(\mathscr{L}_1)$ and hence from Proposition 6 $\mathscr{L}_1^{\downarrow}(s_2) \preceq_P MCD(\Lambda_1)$. Since syn is well-behaved $\mathsf{syn}(\mathscr{L}_1^{\downarrow}(s_2) \preceq \mathsf{syn}(MCD(\Lambda_1))$, but this, together with $\mathsf{syn}(MCD(\Lambda_1)) \in sc(\mathsf{syn}(\Lambda_1))$, contradicts $\mathsf{syn}(\mathscr{L}_1^{\downarrow}(s_1)) \boxdot \mathsf{syn}(\mathscr{L}_1^{\downarrow}(s_2))$. As to the other direction of the proof, assume now that syn is consistency preserving. Since by Proposition 5 for every set $\Lambda_1 \subseteq \Lambda$ it holds that $MCD(\Lambda_1) \in CD(\Lambda_1)$, we can identify a consistent set $\mathscr{L}_1$ of Λ-labellings such that $\mathscr{L}_1^{\downarrow}(s_1) = \Lambda_1$ and $\mathscr{L}_1^{\downarrow}(s_2) = MCD(\Lambda_1)$ with $s_1 \odot s_2$. Then by consistency preservation it must also hold that $\mathsf{syn}(MCD(\Lambda_1)) \in sc(\mathsf{syn}(\Lambda_1))$. □

As an example of application of the above concepts, we show that the function $\mathsf{syn}_{\mathrm{AJ}}$ is consistency preserving for the incompatibility relations $\underline{\mathrm{inc}}_{C^3}$, $\mathrm{inc}^a_{C^3}$, and $\mathrm{inc}^c_{C^3}$ while it is not for $\overline{\mathrm{inc}}_{C^3}$.

First we need to show that $\mathsf{syn}_{\mathrm{AJ}}$ is well-behaved.

Proposition 8 *The ssf* syn_{AJ} *is well-behaved.*

Proof: Since the strict order $\prec$ induced on Λ^{AJ} is total, it is sufficient to show that for any two non-empty sets $\Lambda_1, \Lambda_2 \subseteq \Lambda^{\mathrm{IOU}}$ whenever $\mathsf{syn}_{\mathrm{AJ}}(\Lambda_1) \prec \mathsf{syn}_{\mathrm{AJ}}(\Lambda_2)$ it does not hold that $\Lambda_2 \preceq_P \Lambda_1$. First, it is easy to see that $\{\mathtt{in}\} \not\preceq_P \Lambda_1$ for any $\Lambda_1 \subseteq \Lambda^{\mathrm{IOU}}$, with $\Lambda_1 \notin \{\emptyset, \{\mathtt{in}\}\}$ which covers all cases where $\mathsf{syn}_{\mathrm{AJ}}(\Lambda_1) \prec \mathsf{SkJ}$. Then, it is also easy to see that for any non-empty set Λ_1 such that $\mathtt{in} \notin \Lambda_1$ and any set Λ_2 such that $\{\mathtt{in}\} \subsetneq \Lambda_2$, $\Lambda_2 \not\preceq_P \Lambda_1$, covering all cases where $\mathsf{syn}_{\mathrm{AJ}}(\Lambda_1) \prec \mathsf{CrJ}$ and thus completing the proof. □

Proposition 9 *The ssf* syn_{AJ} *is consistency preserving for the incompatibility relations* $\underline{inc}_{C^3}$*,* $inc^a_{C^3}$*, and* $inc^c_{C^3}$ *while it is not for* $\overline{inc}_{C^3}$*.*

Proof: We need to show that for every non-empty set $\Lambda_1 \subseteq \Lambda^{\mathrm{IOU}}$ it holds that $\mathsf{syn}_{\mathrm{AJ}}(MCD(\Lambda_1)) \in sc(\mathsf{syn}_{\mathrm{AJ}}(\Lambda_1))$. For $\underline{\mathrm{inc}}_{C^3}$, $\mathrm{inc}^a_{C^3}$, and $\mathrm{inc}^c_{C^3}$ this is illustrated in Table 1, where the first column presents the various possible cases for Λ_1 with the relevant value of $\mathsf{syn}_{\mathrm{AJ}}(\Lambda_1)$ and the following columns (illustrating $\underline{\mathrm{inc}}_{C^3}$, $\mathrm{inc}^a_{C^3}$, and $\mathrm{inc}^c_{C^3}$ respectively) show the corresponding $MCD(\Lambda_1)$ and the relevant value of $\mathsf{syn}_{\mathrm{AJ}}(MCD(\Lambda_1))$. By inspection, it can be checked that, as desired, for every pair $(\mathsf{syn}_{\mathrm{AJ}}(\Lambda_1), \mathsf{syn}_{\mathrm{AJ}}(MCD(\Lambda_1)))$ obtained by taking the first element from a row of the first column, and the second element from any other cell (say the i-th with $i \in \{2,3,4\}$) of the same row it holds that $(\mathsf{syn}_{\mathrm{AJ}}(\Lambda_1), \mathsf{syn}_{\mathrm{AJ}}(MCD(\Lambda_1))) \notin \mathrm{inc}'$ where inc' is the incompatibility relation induced by the inc relation specified at the top of the column from which the second element of the pair was taken. For instance, considering the fifth row, with $\Lambda_1 = \{\mathtt{in}, \mathtt{out}\}$ and $(\mathsf{syn}_{\mathrm{AJ}}(\Lambda_1)) = \mathsf{CrJ}$ and its second cell where (according to $\underline{\mathrm{inc}}_{C^3}$) $MCD(\Lambda_1) = \{\mathtt{in}, \mathtt{und}\}$ we have $(\mathsf{syn}_{\mathrm{AJ}}(MCD(\Lambda_1))) = \mathsf{CrJ}$ and then $(\mathsf{CrJ}, \mathsf{CrJ}) \notin \mathrm{inc}'$ since $(\mathsf{mid}, \mathsf{mid}) \notin \underline{\mathrm{inc}}_{C^3}$.

Concerning $\overline{\mathrm{inc}}_{C^3}$ a counterexample is provided by $\Lambda_1 = \{\mathtt{in}, \mathtt{out}\}$ with $MCD(\Lambda_1) = \{\mathtt{in}, \mathtt{out}\}$ and $\mathsf{syn}_{\mathrm{AJ}}(\Lambda_1) = \mathsf{syn}_{\mathrm{AJ}}(MCD(\Lambda_1)) = \mathsf{CrJ}$ while $(\mathsf{mid}, \mathsf{mid}) \in \overline{\mathrm{inc}}_{C^3}$. □

The fact that $\mathsf{syn}_{\mathrm{AJ}}$ is not consistency preserving according to $\overline{\mathrm{inc}}_{C^3}$ is not surprising, given that $\overline{\mathrm{inc}}_{C^3}$ essentially reflects the fully bipolar nature of stable semantics, while $\mathsf{syn}_{\mathrm{AJ}}$ admits tripolar assessments.

Λ_1 $\mathsf{syn}_{\mathsf{AJ}}(\Lambda_1)$	$\underline{\mathrm{inc}}_{C^3}$	$\mathrm{inc}^a_{C^3}$	$\mathrm{inc}^c_{C^3}$
$\{\mathtt{in}\}$ SkJ	$\{\mathtt{und}\}$ NoJ	$\{\mathtt{und}\}$ NoJ	$\{\mathtt{out}\}$ NoJ
$\{\mathtt{out}\}$ NoJ	$\{\mathtt{in}\}$ SkJ	$\{\mathtt{in}\}$ SkJ	$\{\mathtt{in}\}$ SkJ
$\{\mathtt{und}\}$ NoJ	$\{\mathtt{in}\}$ SkJ	$\{\mathtt{und}\}$ NoJ	$\{\mathtt{und}\}$ NoJ
$\{\mathtt{in},\mathtt{out}\}$ CrJ	$\{\mathtt{in},\mathtt{und}\}$ CrJ	$\{\mathtt{in},\mathtt{und}\}$ CrJ	$\{\mathtt{in},\mathtt{out}\}$ CrJ
$\{\mathtt{in},\mathtt{und}\}$ CrJ	$\{\mathtt{in},\mathtt{und}\}$ CrJ	$\{\mathtt{und}\}$ NoJ	$\{\mathtt{und},\mathtt{out}\}$ NoJ
$\{\mathtt{und},\mathtt{out}\}$ NoJ	$\{\mathtt{in}\}$ SkJ	$\{\mathtt{in},\mathtt{und}\}$ CrJ	$\{\mathtt{in},\mathtt{und}\}$ CrJ
$\{\mathtt{in},\mathtt{und},\mathtt{out}\}$ CrJ	$\{\mathtt{in},\mathtt{und}\}$ CrJ	$\{\mathtt{in},\mathtt{und}\}$ CrJ	$\{\mathtt{in},\mathtt{und},\mathtt{out}\}$ CrJ

Table 1. Illustration of the proof of Proposition 9.

5. Discussion and conclusion

We have introduced a generalized notion of consistency and provided two initial examples of its possible uses in formal argumentation: revisiting some of Dung's traditional semantics from a perspective of progressive strengthening of consistency requirements and characterizing the consistency preservation of operators which produce assessments as a synthesis of sets of labellings, as is the case for the traditional notion of argument justification.

To our knowledge, providing a generalized form of the notion of consistency has not been previously considered in the formal argumentation literature, while other related and complementary research directions have been pursued. For instance, in [7] the idea of encompassing some inconsistency tolerance, through an *inconsistency budget* in the semantics of weighted argumentation systems is considered. This proposal does not address the issues we consider for traditional argumentation frameworks, while extending our approach to the case of weighted systems appears to be an important direction for future work. In [8] the notion of conflict-tolerant semantics is introduced, which is essentially based on lifting the requirement of conflict-freeness in semantics definition. In the context of our approach, this corresponds to making the intolerance relation empty, while keeping other constraints: again, we consider drawing correspondences between our approach and this proposal as interesting future work. In [9] the problem of measuring inconsistency in (abstract and structured) argumentation formalisms is addressed: this is an orthogonal research direction as we do not aim at quantifying inconsistency in a given setting, but rather at encompassing different notions of inconsistency. Bridging the two directions appears worth investigating.

Extending the analysis beyond tripolar classifications is another important future development. For example, more articulated notions of argument justification have been considered in the literature [10,11,12,13]. Dealing with consistency and its preservation in such a context might require considering different sets of assessment classes and defining a notion of refinement between them.

Addressing the evaluation of argument conclusions and their consistency is a further key step. In particular, it would be interesting to extend the notions presented in this paper to the formalism of multi-labelling systems [14], which can capture a variety of approaches to derive the assessment of conclusions from the assessment of arguments. This will require to tackle several additional aspects, like addressing the connections between intolerance relations involving entities at different levels and dealing with the various possible mechanism for synthesizing the labellings of conclusions after projecting argument labellings on them.

Finally, we suggest that, in the long term, the potential uses of the proposed approach go beyond the formal argumentation field. Consistency is a crucial aspect of most, if not all, reasoning formalisms, typically defined using their structural elements. Exposing the elementary concepts composing the notion of consistency brings, among others, the following advantages. Firstly, it may enable inter-formalism analyses, comparisons, and cross-fertilization. Further, it may provide a basis for developing novel theoretical and practical tools, like, for instance, methods to preserve consistency across different reasoning stages or general-purpose parametric consistency checkers.

References

[1] Dung PM. On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming, and n-Person Games. Artif Intell. 1995;77(2):321-57.

[2] Baroni P, Caminada M, Giacomin M. An introduction to argumentation semantics. Knowledge Engineering Review. 2011;26(4):365-410.

[3] Caminada M, Amgoud L. On the evaluation of argumentation formalisms. Artif Intell. 2007;171:286 310.

[4] Jakobovits H, Vermeir D. Robust Semantics for Argumentation Frameworks. J of Logic and Computation. 1999;9(2):215-61.

[5] Modgil S, Prakken H. A general account of argumentation with preferences. Artif Intell. 2013;195:361 397.

[6] Baroni P, Giacomin M, Liao B. Dealing with Generic Contrariness in Structured Argumentation. In: Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, IJCAI 2015; 2015. p. 2727-33.

[7] Dunne PE, Hunter A, McBurney P, Parsons S, Wooldridge MJ. Weighted argument systems: Basic definitions, algorithms, and complexity results. Artif Intell. 2011;175(2):457-86.

[8] Arieli O. Conflict-free and conflict-tolerant semantics for constrained argumentation frameworks. J Appl Log. 2015;13(4):582-604.

[9] Hunter A. Measuring Inconsistency in Argument Graphs. CoRR. 2017;abs/1708.02851. Available from: http://arxiv.org/abs/1708.02851.

[10] Caminada MWA, Wu Y. On the Justification Status of Arguments. In: Proc. of the 22nd Benelux Conference on Artificial Intelligence; 2010. .

[11] Wu Y, Caminada MWA. A Labelling-Based Justification Status of Arguments. Studies in Logic. 2010;3(4):12-29.

[12] Dvořák W. On the Complexity of Computing the Justification Status of an Argument. In: Modgil S, Oren N, Toni F, editors. Proc. of the 1st Int. Workshop on Theory and Applications of Formal Argumentation (TAFA 2011). No. 7132 in Lecture Notes in Computer Science. Springer; 2012. p. 32-49.

[13] Baroni P, Giacomin M, Guida G. Towards a Formalization of Skepticism in Extension-based Argumentation Semantics. In: Proc. of the 4th Workshop on Computational Models of Natural Argument (CMNA 2004); 2004. p. 47-52.

[14] Baroni P, Riveret R. Enhancing Statement Evaluation in Argumentation via Multi-labelling Systems. J Artif Intell Res. 2019;66:793-860.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220142

Argument Schemes for Factor Ascription

Trevor BENCH-CAPON and Katie ATKINSON
Department of Computer Science, University of Liverpool, UK

Abstract. Reasoning with legal cases by balancing *factors* (reasons to decide for and against the disputing parties) is a two stage process: *first* the factors must be ascribed and *then* these reasons for and against weighed to reach a decision. While the task of determining which set of reasons is stronger has received much attention, the task of factor ascription has not. Here we present a set of argument schemes for factor ascription, illustrated with a detailed example.

Keywords. argument schemes, explanation, factors, legal reasoning

1. Introduction

Reasoning with legal cases provides a paradigmatic example of reasoning involving the weighing of reasons for and against to come to a conclusion which will be consistent with decisions made in the past. It is a central feature of legal reasoning, and as such has been central to the study of AI and Law. The most successful approach has been to use *factors*, which were introduced in CATO [2] as a development from the dimensions introduced in HYPO [22]. Further developments are described in [5], and the most recent has been a summary of formalisations of this approach [18]. Such reasoning has received recent attention in [13] as a way of the explaining in the output of approaches using machine learning techniques to predict the outcomes of legal cases, such as [1] and [17]. The same techniques have also been used to explain classifications from machine learning systems in areas other than law [19].

Since [2], cases have been represented as sets of *factors*. Factors are stereotypical patterns of facts which offer a reason to decide for one or other of the parties to the dispute. Once cases are available in terms of factors, the reasoning becomes a matter of deciding which set of reasons is the stronger[1], given the need to be consistent with the preferences expressed in previous cases (the legal principle of *stare decisis*). Argumentation about a new case then typically involves the exchange of moves (shown in *italics* in the bullets below) structured in the form of a three-ply dialogue, as originated in HYPO:

- Plaintiff *cites* a favourable precedent case with factors in common with the current case;
- Defendant replies either by citing an unfavourable case with factors in common with the current case (*counter example*) or *distinguishes* the case, by pointing to factors favourable to the defendant in the current case but not in the precedent, or factors favourable to the plaintiff in the precedent but not the current case;

[1] Often the strength of a reason is seen in terms of the social values it promotes or demotes [10]

- Plaintiff rebuts by *distinguishing* the counter examples, or by down playing the distinctions by finding a factor to *substitute* for the missing favourable factor or *cancel* the unfavourable factor.

This three ply structure corresponds to the use of argument schemes as introduced by Walton [24], with the proposal of an argument scheme (*cite*) being answered with critical questions (*counter example* and *distinguish*), which are in turn responded to (*distinguish*, *substitute* and *cancel*). This style of argument has been formalised as a set of argument schemes in [20], and used to explain the output of machine learning systems in [19].

In order to reason with factors, however, factors must be ascribed to the cases. Reasoning with legal precedents is a two stage process, in which *first* factors are ascribed to cases and *then* weighed to see whether the reasons for the defendant or the plaintiff are the stronger (see e.g. [12] and [8]). Both these stages may be the subject of argument: it may not be clear whether or not a factor should be ascribed on the basis of the facts.

Moreover, precedents guide the ascription of factors. Whilst the systems which look back to CATO see precedents as determining preferences between sets of factors (e.g. [16]), some precedents are concerned with whether or not a factor should be ascribed: once the factors are ascribed, the balance between them is obvious. For example, in the domain used by CATO, US Trade Secrets – where cases cover the misappropriation (or not) of a trade secret – in the case of *Arco Industries Corp. v. Chemcast Corp (1980)* all the factors favoured the defendant, and so, once the factors had been ascribed, the case was entirely clear cut. The plaintiff had, however, argued that certain factors favourable to him *should* be ascribed, but his arguments were rejected: in particular the court rejected the argument that the defendant's product was identical to the plaintiff's, setting a precedent for how similar the products needed to be for the purposes of this factor. Similarly in *Technicon Data Systems Corp. v. Curtis 1000, Inc (1984)* , the case turned on whether or not the information was reverse engineerable. The court held that, given the time the defendant had expended on attempting to reverse engineer the information, the factor should not be ascribed. In doing so the Court set a precedent for how much effort precluded reverse engineerability. As a case where a disputed factor was held to be present, it was held in *Space Aero Products Co. v. R.E. Darling Co* (1965), that former employees who had acquired the information in the course of their employment were in a confidential relationship with the employer, even if they had signed no explicit non-disclosure agreement, again setting a precedent for future cases. Thus while some precedents guide preferences between factors, others, termed *ascription* precedents in [8], guide the ascription of factors.

Thus the explanation of an outcome in terms of the factors pro and con, and the preferences between them, may not be enough: an explanation of why the factors are considered present (or absent) may be what is needed.

In this paper we present a set of argument schemes to justify factor ascription. Our contribution is the articulation of four new schemes and their critical questions that demonstrate how factors can be ascribed, to enable the second stage of reasoning with legal cases to be undertaken. The new schemes complement the set presented previously in [20] to now provide full coverage of both stages of the process described earlier: this enables a *complete* explanation of *both* stages of the decision that was not previously available. Use and effectiveness of the schemes is demonstrated by walking through a detailed example.

Section 2 identifies different ways of ascribing factors, giving rise to different schemes to provide justification. Section 3 presents schemes for each of these types, with an extended example following in Section 4 and concluding remarks in Section 5.

2. Types of Factor Ascription

Factors are ascribed on the basis of facts. But the relationship between factors and facts is not always straightforward, since the cases exhibit an enormous variety of facts, which need to be mapped into a relatively small set of factors.

The most straightforward case is where the facts licence the ascription of the factor through an ordinary understanding of the words involved. Thus, in the often discussed wild animals property law cases introduced in [11] – where the cases concerned determining when an individual can be deemed to possess (or not) a wild animal being pursued – one factor was whether those involved were pursuing their livelihoods. In one case[2], Keeble rented a pond frequented by ducks which he regularly shot and sold at market. In another[3], Young was a professional fisherman looking for fish in his trawler. Both are clearly pursuing their livelihoods. In contrast, in *Pierson v Post* (1805)[4], Post was hunting a fox in pursuit of pleasure: fox hunting played no part in his professional activities. No special judgement, or knowledge of past cases, is needed to ascribe the factor. In these cases there are no issues as to extent: either the plaintiff was pursuing his livelihood or he was not.

But things are not always so simple: often the ascription of the factor does require knowledge of past cases. In HYPO [3], in the domain of US Trade Secrets Law, facts are used to position the case on a set of *dimensions*. One such dimension is *SecurityMeasures*, which relates to the steps taken by plaintiffs to protect their information. The measures taken in any given case will be very different. Moreover, the rigour of the measures will vary greatly: at one end of the dimension no measures will have been taken, which will be a reason to find for the defendant. At the other end, the very rigorous measures taken will be a reason to find for the plaintiff. Thus the dimension gives rise to two factors, one for each side, depending on the nature of the measures taken in the particular case. Between them there *may* be a neutral range in which neither side is favoured and no factor is applicable. At one end we have the pro-defendant factor *NoSecurityMeasures*, and at the other end the pro-plaintiff factor, *AdequateSecurityMeasures*. At some point on the dimension the measures will be sufficient to be no longer a reason to find for the defendant. At some, possibly later, point the measures will be sufficiently strong to be a reason to find for the plaintiff. These points are not a matter of ordinary language, but are determined in the context of actual cases, and dependent on past decisions. These points were discussed by Rigoni [21], who termed them *switching points*. *SecurityMeasures* has two factors: a pro–defendant factor, followed by a neutral range, followed by a pro-plaintiff factor. Other dimensions may differ. Some map to only one factor with the rest of the range neutral. Others may map to more than one factor for a given side: in Trade Secrets *DisclosuresToOutsiders* has a neutral range followed by a pro-plaintiff factor representing a significant number of disclosures, followed by a stronger pro-plaintiff

[2] *Keeble v Hickeringill* (1707) 103 ER 1127

[3] *Young v Hitchens* (1844) 115 ER 228

[4] *Pierson v Post* (1805) 3 Cai. R. 175, 2 Am. Dec. 264

factor if the information has been put into the public domain. For all factors of this type we have a dimension with ranges in which the factors that are applicable are demarcated by switching points constrained by precedents.

A third type of ascription arises when a pair of dimensions need to be considered together, because one may trade off against the other so that we need to strike a balance between them. For example, as discussed in [9], in the US Fourth Amendment which protects against unreasonable search, the privacy of the citizen must be balanced against the exigency of the need to enforce the law. If the life of the President is thought to be under threat, privacy will be respected less than if we are dealing with a minor offence. Neither exigency nor privacy can provide a reason to decide the case by themselves: they must be considered together. Thus the factor is something like *SufficientRespectforPrivacyGivenExigency*, and we can picture the dimensions as the x- and y-axes and precedents determining a line separating the area where the factor applies from where it does not, as discussed in [7] and [6]. Again, this line will be constrained by previous decisions as to the applicability of the factor. A diagram for the worked example is given in Figure 1.

The fourth type of justification we will consider is analogy. Here the problem is not whether the extent of some aspect of a case is sufficient, and so it *could* be a case for ordinary interpretation, but the facts do not allow the ascription of the factor on an ordinary interpretation, and so it is argued that the factor should be ascribed on the basis of an *analogy* between the situations. Analogy has often been seen as determining a relation between whole cases, as in [25] and [23]. We, however, argue that the analogy is rather between particular aspects of the case. In Steven's prime example in [23] the (hypothetical) case turns on whether a kindergarten teacher should be considered sufficiently analogous to the mother in the case of *Dillon v Legg*[5] to receive compensation for witnessing an injury to a child. In *Dillon* it was held that one factor was whether the victim and the emotional sufferer had "a close relationship"[6]. The case turns on the point because the other pro-claimant factors are in place, and so all that is at issue is whether being a kindergarten teacher enables the factor *CloseRelationship*, introduced to describe the mother-son relation, to be applied. In ordinary language most people would not say that a kindergarten teacher had a close relationship of this sort to the child, but Stevens argues that there is a possible analogy to be drawn on the basis that both love the child. In our opinion this analogy is too tenuous, but a similar analogy might succeed in the case of a more intimate non-blood relationship such as a wet nurse. Walton's key example is the well known *Popov v Hayashi* case[7] [4], in which there was a dispute over the ownership of a potentially valuable baseball, hit by Barry Bonds to set a home run record. Here the analogy is not with a particular case, but with the body of cases including *Pierson*, *Keeble* and *Young* discussed above, by drawing an analogy to a baseball hit out of the ball park and the wild animals of those cases [14]. Here what is at stake is the ascription of the factors relating to the quarry: once these factors have been ascribed, *Popov* can then be argued about with the other wild animals cases as precedents.

[5] *Dillon v Legg* 68 Cal. 2d 728 (1968).

[6] The Court had been deliberately vague, saying that it "cannot now predetermine the defendant's obligation in every situation by a fixed category; no immutable rule can establish the extent of that obligation for every circumstance in the future." It is quite normal for a Court to use a term which is clearly satisfied in the current case, but leave more precise lines to be drawn later.

[7] *Popov v Hayashi* WL 31833731 Ca. Sup. Ct. (2002). The case was the subject of a 2004 comic documentary film, *Up For Grabs*, https://www.imdb.com/title/tt0420356/.

3. Argumentation Schemes For Factor Ascription

In this section we will provide an argumentation scheme for the ascription of each of the types of factor discussed in the previous section.

3.1. Ordinary Meaning Scheme

The most straightforward scheme is where the facts of the case justify the ascription of a factor, on a ordinary interpretation of the terms involved.

Ordinary Meaning Scheme

Facts Premise: Facts $a_1...a_n$ are true in Case C_1

Usage Premise: As F is ordinarily understood, $a_1...a_n$ are sufficient for Factor F to be considered present in C_1

Conclusion: F is present in C_1

The following is a set of critical questions to enable the scheme's components to be questioned:

MCQ1: Does $a_1...a_n$ really justify the ascription of F? There might be some additional fact which is needed. For example we might require that the activity be sufficiently remunerative if it to be considered the person's livelihood.

MCQ2: Does some other fact, b, provide an exception which prevents the ascription of F? There might be some unusual feature in the situation which should prevent ascription. For example suppose a duck hunter takes a pot shot at a fox. Though out pursuing his livelihood, hunting that particular quarry is not really part of that pursuit.

MCQ3: Do other facts $b_1...b_n$ justify the ascription of factor F_2, which is incompatible with F? For example if the person concerned was trespassing, then they might be considered to be engaged in an activity such as poaching, which would not be considered earning a livelihood.

3.2. Switching Point Scheme

The next scheme is based on Rigoni's notion of a *switching point* [21]. If we consider a dimension with a factor favouring the plaintiff at one end and a factor favouring the defendant at the other, there will be points (possibly the same) at which one factor ceases to apply and the other factor begins to apply. These are the *switching points*. Thus given a precedent more favourable on the dimension than the new case, we can say that the factor applies to the new case. Similarly, if the new case is less favourable, we can argue that the factor does not apply. We can use this notion as the basis of an argumentation scheme:

Switching Point Scheme

Precedent Premise: P_1 is a precedent with location L_1 on dimension D at which factor F is present.

Case Premise: C_1 is a case with L_2 on dimension D

Party Premise: F favours the plaintiff (defendant)

Value Premise: L_2 is more (less) favourable to the plaintiff (defendant) than L_1

Conclusion: F applies (does not apply) to C_1

We can question an instantiation of ths scheme with the following critical questions:

SCQ1: *Is L_2 so much more favorable that a different factor applies?* For example in the US Trade Secret domain of [2] there are two pro-defendant factors on the disclosures dimension, *DisclosedToOutsiders* and the stronger *DisclosedInPublicForum*.

SCQ2: When arguing that the factor does not apply because L_2 is less favourable: *Is L_2 sufficiently close to L_1 that the same factor applies?* It is possible that P_1 does not precisely identify the switching point, and that C_1 may become a new precedent for the factor, giving a more generous switching point.

SCQ3: *Is there another precedent, P2, which can ground an instantiation of the switching point scheme to give an argument that the factor does not (does) apply?*. It may be that some additional information is needed to say which precedent should apply.

3.3. Trade Off Scheme

The next scheme concerns trade-offs between two dimensions, as described in [6]. For example in the US Fourth Amendment domain there is a trade-off between being able to enforce the law and respect for privacy [9]. The factor involves balancing these two concerns and is something like “Sufficient respect for privacy while enabling enforcement”. The idea in [6] is that a line, e.g $a.D_1 + b.D_2 + c = 0$, can be fitted to the precedents[8], separating the pro-plaintiff and pro-defendant regions of the case space. In the equation a and b are the coefficients of the variables D_1 and D_2, representing the values on the two dimensions. This determines the gradient of the line which indicates how much more D_1 is need to compensate for less D_2, and c is a constant showing where the line crosses the axes given these coefficients.

Trade Off Scheme

Precedents Premise: $P_1 \ldots P_n$ are precedent cases in which factor F is present.

Locations Premise: Precedent $P_i \in \{P_1 .., P_n\}$ have locations $D1_i$ and $D2_i$ for dimensions $D1$ and $D2$,

Case Premise: C_1 is a case with L_1 on dimension D_1 and L_2 on dimension D_2

Line Premise: All $a.D1_i + b.D2_i + c \geq 0$

Point Premise: $a.L_1 + b.L_2 + c \geq (<)\ 0$

Conclusion: F applies (does not apply) to C_1

For this scheme we have the following key critical questions;

TCQ1: Is there a counter example, a precedent, P_{n+1}, such that $a.D1_{n+1} + b.D2_{n+1}$ $+ c < (\geq)\ 0$?. There might be a precedent which does not fit the line.

TCQ2: Can the line be drawn less (more) tightly? If the precedents are not precisely on the line the constant c could be adjusted to lower (raise) the line to allow (disallow) more cases to qualify unless this created a counter example.

[8] Of course more complicated curves can be used, but a straight line is the simplest.

3.4. Analogy Scheme

There are a number of schemes for analogy in the literature. Different schemes are given in [26], [25] and [23]. Here we give one tailored to our need to analogise between aspects of cases rather than cases as a whole.

Analogy Scheme
Base Premise: A situation S_1 is described in precedent P_1.
Derived Premise: Factor F is plausibly ascribed to P_1 on the basis of S_1.
Case Premise: Case C_1 contains situation S_2
Similarity Premise: As it relates to F, situation S_2 is similar to situation S_1.
Conclusion: Factor F is plausibly ascribed to C_1.

The following set of critical questions is based on the account given in [26] for the basic scheme for argument from analogy.

ACQ1: Are there respects in which P_1 and C_1 are different that would tend to undermine the force of the similarity with respect to F? For example, the kindergarten teacher has a transient relationship, whereas the maternal relation is permanent.

ACQ2: Is the similarity sufficient for F to be ascribed? The love felt by a kindergarten teacher might not be considered to have the same quality as mother-love.

ACQ3: Is there some other precedent P_2 that is also similar to C_1, but in which F was not ascribed? Suppose there was a precedent with a nanny, where the relationship was not considered sufficiently close.

In the next section we will give an extended example to show these schemes in action.

4. Change of Domicile Example

Our example will be based on the example used in [15]. The idea is that a person has applied for a change of fiscal domicile, for tax purposes. A decision will be taken in the light of the particular facts of the case, which will be very varied as people have many different reasons for changing country, and many different ways of arranging their lives as they transition from one place to another. We will suppose the following facts are relevant. Some will be *dimensional*, the particular fact representing a point on a dimension, while others will be Boolean.

- **Absence**: The length of absence (dimensional).
- **IncomeSource**: The percentage of income earned abroad (dimensional).
- **Spouse** Whether there are family connections with the new country. For example the spouse may be a national of that country (Boolean).
- **Age**. The age of the person concerned (dimensional).
- **Dwelling**: Whether links with the old country had been maintained. For example a house may still be owned there (Boolean).

From these we can form the following factors. The dimensional facts will give rise to *switching point* and *trade off* factors. Boolean factors may be argued for either on the basis of a literal interpretation or an analogy. The conflicting principles are that tax should be paid where income is earned, but that tax should be paid where benefits are

Table 1. Facts of Example Cases

Case	Absence	IncomeSource	Spouse	Age	Dwelling
LowPay	48	20	No	35	Timeshare
HighPay	6	100	No	17	No
Married	36	60	Spouse	26	No
Owner	60	20	No	48	House
NewCase1	54	20	Spouse	66	Caravan
NewCase2	36	60	Partner	18	No

Table 2. Factors Present In Precedent Cases

Case	F1 suff	F2 Insuff	F3 Family	F4 Working	F5 Minor	Links
LowPay		X		X		
HighPay	X				X	
Married	X		X	X		
Owner	X			X		X

received, and so the current domicile should receive tax until the connection has been severed by prolonged absence and abandoning other connections.

F1 **SufficientAbsence**. This will mean that the absence is sufficient with respect to the amount of income earned abroad. This is a *trade off* factor. In general, the higher the percentage earned abroad the shorter the absence required. The factor will favour *change*.

F2 **InufficientAbsence**. If the absence is not sufficient, this can mean that a pro *noChange* factor is present.

F3 **Family**. There are close family ties with the new country. This favours *change*. Here there is no corresponding *noChange* factor: lacking family ties is no reason to decide for *noChange*.

F4 **WorkingAge**. If a person is of working age, this favours no change.

F5 **Minor**.If a person is a minor, this favours *change*, since minors are held to have no control of their domicile. Note, however, that being retired is neutral: thus, some points on the age dimension are neutral.

F6 **Links**. This is a Boolean factor favouring *noChange*. It applies if the claimant has maintained links such as property in the old country.

We now consider a set of precedent cases, with facts as shown in Table 1. The factors ascribed to the four precedent cases are shown in Table 2. We are now presented with a new case, *NewCase1*, with facts as shown in Table 1.

We now consider how these factors should be ascribed to *NewCase1*. First we will consider the income/absence trade off.

4.1. Trade Off Between Absence and Income Percentage

NewCase1 has the same amount of income as the unfavourable *LowPay* and the favourable *Owner*. But the length of absence is midway between the two. So argument is required. Suppose first that the advocate of no change wishes to argue that F1, *SufficientAbsence* does not apply. He can draw a line separating the favourable from the

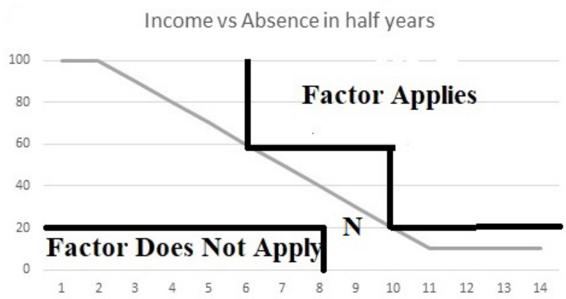


Figure 1. Trade off between absence and income

unfavourable precedents as shown in Figure 1. The line shown is $3 \times income + 5 \times absence - 360 = 0$. We can now instantiate the **TradeOff** scheme.

> *Precedents Premise*: *Married* and *Owner* are precedent cases in which factor *SufficientAbsence* is present.
> *Locations Premise*: These precedents have locations (36,60) and (60,20) on the absence and income dimensions.
> *Case Premise*: *NewCase1* has location (54,20)
> *Line Premise*: For *Married* and *Owner* $3 \times income + 5 \times absence - 360 \geq 0$
> *Point Premise*: $3 \times 20 + 5 \times 54 - 360 = -30$ and $-30 < 0$
> *Conclusion*: *SufficientAbsence* does not apply to *NewCase1*

We can now consider how we might challenge this argument. In *HighPay*, we have a negative value for $3 \times 100 + 5 \times 6 - 360$, and yet it was held that *SufficientAbsence* applied to *HighPay*. This counter example to the Line Premise allows us to pose TCQ1. The proponent would likely argue that the relationship is not linear for extreme values, as shown in Figure 1, so that 100% income will mean that *any* absence is sufficient. The court would have to decide whether this rebuttal was effective, or whether *NewCase1* should also be included in *SufficientAbsence*.

The line in Figure 1 has been drawn very tightly on the precedents, and so there is scope for challenging this argument with TCQ2. If the constant factor is reduced to 330, *NewCase1* would now lie on the line, and the two precedents would remain included, while the counterexample *LowPay* would remain excluded. This would have the additional strength of including *HighPay* with no need to make it a special case, and so represents a powerful challenge. A rebuttal would have to argue in terms of the values promoted: that the new line does not give enough weight to the claims of the current domicile.

4.2. Switching Point on Age Dimension

The factors for which this is the argument scheme are *WorkingAge* and *Minor*. Let us first consider *NewCase1*. Here a proponent of no change would instantiate the scheme using *Owner*.

> *Precedent Premise*: *Owner* is a precedent with location 48 on dimension *Age* at which factor *WorkingAge* is present.

> *Case Premise* :*NewCase1* is a case with 66 on dimension *Age*.
> *Party Premise*: *WorkingAge* favours no change.
> *Value Premise*: 66 is more favourable to no change than 48
> *Conclusion*: *WorkingAge* applies to *NewCase1*.

But we can pose critical questions against this, knowing that beyond *WorkingAge* on the *Age* dimension is a neutral area for the retired. So one can pose SCQ1, arguing that the difference between 66 and 48 is sufficiently great that we have entered the neutral range. We have no precedents setting the switching point here: if the court decides that a person of 66 should be considered retired so that *WorkingAge* does not apply, *NewCase1* will become a precedent putting an upper bound on *WorkingAge*.

The other two critical questions do not apply to *NewCase1*, so now consider *NewCase2*. Here *HighPay* can be used as a precedent to argue that *Minor* does not apply.

> *Precedent Premise*: *HighPay* is a precedent with location 17 on dimension *Age* at which factor *Minor* is present.
> *Case Premise* :*NewCase2* is a case with 18 on dimension *Age*.
> *Party Premise*: *Minor* favours change.
> *Value Premise*: 18 is less favourable to change than 17
> *Conclusion*: *Minor* does not apply to *NewCase2*.

Here we can pose SCQ2. We can argue that 18 is sufficiently close to 17 that *Minor* can also be taken to apply the *NewCase2*, raising the lower bound on the switching point for this factor. This could be supported by also posing SCQ3, using *Married* to argue that *WorkingAge* does not apply to *NewCase2*. The decision on the court could be made on the basis of the value concerned, namely the autonomy of the person with respect to place of residence of a person of that age, or by considering other legislation in the two jurisdictions concerning the age of majority.

4.3. *Ordinary Meaning Scheme*

We now turn to the Ordinary Meaning Scheme. Consider here *NewCase1*, where the applicant has a spouse of the other nationality.

> *Facts Premise*: Applicant has a spouse of the appropriate nationality in *NewCase1*,
> *Usage Premise*: As ordinarily understood, spouse justifies the ascription of *Family*, which requires a close family tie.
> *Conclusion*: *Family* is present in *NewCase1*,

Since spouse is a paradigmatic close family tie, the only possibility of arguing against this is to find an exception, and pose MCQ2. For example if the couple were legally separated this would provide a possible reason to withhold the factor.

For the other critical questions, consider *NewCase2*. Here we have a partner, but no legal marriage, yet a partnership might well be considered effectively the same as a formal marriage. However, it is possible that additional conditions could be suggested using MCQ1. For example it could be argued that to be treated equally with marriage there should be some evidence of permanence, such as the relationship having existed for a significant period of time. For MCQ3 it might be possible to point to another concept such as *cohabitee*, which is treated differently from a spouse with regard to, e.g. inheritance and welfare benefits. Here it could be argued that a partner is more readily seen under this concept than as a spouse.

4.4. Analogy Scheme

We will discuss the analogy scheme and its relation to the *Links* factor, specifically whether a dwelling had been maintained in the country of origin, with Owner as the *precedent*. While the caravan in *NewCase1* would not be interpreted as a dwelling in the ordinary meaning of the word, it could be argued that there is a sufficient analogy to allow the *Links* factor to apply.

> *Base Premise*: A house in the country of origin is present in *Owner*.
>
> *Derived Premise*: *Links* is plausibly ascribed to *Owner* on the basis of the retained house.
>
> *Case Premise*: In *NewCase1* a caravan in the country of origin has been retained.
>
> *Similarity Premise*: As it relates to *Links*, a caravan is similar to a house since they can both be used to live in.
>
> *Conclusion*: *Links* is plausibly ascribed *NewCase1*.

This analogy can, of course, be questioned. Using ACQ1, it could be argued that there are significant differences between a house and a caravan, such as mobility and that the latter is used typically as a second home for short stays. ACQ2 is similar, but whereas ACQ1 could be answered by suggesting that the other caravans were in use as permanent dwellings, ACQ2 suggests that the similarity is still insufficient. For example that given the disparity in cost between retaining a house and a caravan, the caravan did not demonstrate a big enough commitment to qualify for ascription of the *Links* factor. Finally, for ACQ3 we can point to *LowPay*, where it was held that a Timeshare was not enough to ascribe the *Links* factor. This would allow the argument that the Timeshare is a better analogy for a caravan than a house.

5. Concluding Remarks

A full explanation of a decision in a legal case requires not only an indication of why one set of factors was preferred over another, as in [20], but also why these reasons were considered applicable to the case, and why others were not. To enable this kind of explanation we have provided a set of argumentation schemes for factor ascription. We have identified different kinds of factors, which require different kinds of justification. One key aspect is whether the factor is derived from a dimension or a Boolean attribute. Another is whether aspects of the case trade off against one another. We have identified four schemes, and their associated critical questions to enable critical discussion in the manner of [24], which can be deployed as a three ply dialogue as advocated in HYPO [22]. When coupled with the schemes of [20], a complete explanation of the reasoning in legal cases is enabled. We propose that these schemes could be used as in [19] to give a full explanation of the output of a machine learning prediction system, both in law and in other domains.

We have illustrated the schemes with an extended example, based on one in [15], but in future work we will analyse an actual body of case law using these schemes. One possibility would be US Trade Secrets, allowing comparison with a range of other work [5]. We anticipate that this may require us to expand our initial set of critical questions.

References

[1] Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93, 2016.

[2] Vincent Aleven. *Teaching case-based argumentation through a model and examples*. PhD thesis, University of Pittsburgh, 1997.

[3] Kevin D Ashley. *Modeling legal arguments: Reasoning with cases and hypotheticals*. MIT press, Cambridge, Mass., 1990.

[4] Katie Atkinson, editor. *Special Issue on Modelling Popov v Hayashi*, volume 20:1 of *AI and Law*. 2012.

[5] Trevor Bench-Capon. HYPO's legacy: Introduction to the virtual special issue. *AI and Law*, 25(2):205–250, 2017.

[6] Trevor Bench-Capon. Using issues to explain legal decisions. In *Proceedings of XAILA 2021*. arXiv preprint arXiv:2106.14688, 2021.

[7] Trevor Bench-Capon and Katie Atkinson. Dimensions and values for legal CBR. In *Proceedings of JURIX 2017*, pages 27–32, 2017.

[8] Trevor Bench-Capon and Katie Atkinson. Precedential constraint: The role of issues. In *Proceedings of the 18th International Conference on Artificial Intelligence and Law*, pages 12–21. ACM Press, 2021.

[9] Trevor Bench-Capon and Henry Prakken. Using argument schemes for hypothetical reasoning in law. *AI and Law*, 18(2):153–174, 2010.

[10] Trevor Bench-Capon and Giovanni Sartor. Theory based explanation of case law domains. In *Proceedings of the 8th International Conference on Artificial Intelligence and Law*, pages 12–21, 2001.

[11] Donald H Berman and Carole L Hafner. Representing teleological structure in case-based legal reasoning: The missing link. In *Proceedings of the 4th International Conference on Artificial Intelligence and Law*, pages 50–59, 1993.

[12] L Karl Branting. Explanation in hybrid, two-stage models of legal prediction. In *XAILA 2020: EXplainable and Responsible AI in Law 2020*, pages 1–10. CEUR 2891, 2020.

[13] L Karl Branting, Craig Pfeifer, Bradford Brown, Lisa Ferro, John Aberdeen, Brandy Weiss, Mark Pfaff, and Bill Liao. Scalable and explainable legal prediction. *AI and Law*, 29(2):213–238, 2021.

[14] Paul Finkelman. Fugitive baseballs and abandoned property: Who owns the home run ball. *Cardozo Law Review*, 23:1609, 2001.

[15] John Horty. Reasoning with dimensions and magnitudes. *AI and Law*, 27(3):309–345, 2019.

[16] John Horty and Trevor Bench-Capon. A factor-based definition of precedential constraint. *AI and Law*, 20(2):181–214, 2012.

[17] Masha Medvedeva, Xiao Xu, Martijn Wieling, and Miche Vols. Juri says: An automatic judgement prediction system for the European Court of Human Rights. In *Proceedings of JURIX 2020*, pages 277–280, 2020.

[18] Henry Prakken. A formal analysis of some factor-and precedent-based accounts of precedential constraint. *AI and Law*, 29(4):559–585, 2021.

[19] Henry Prakken and Ratsma Rosa. A top-level model of case-based argumentation for explanation: formalisation and experiments. *Argument and Computation*, 13(2):159–194, 2022.

[20] Henry Prakken, Adam Wyner, Trevor Bench-Capon, and Katie Atkinson. A formalization of argumentation schemes for legal case-based reasoning in ASPIC+. *Journal of Logic and Computation*, 25(5):1141–1166, 2015.

[21] Adam Rigoni. Representing dimensions within the reason model of precedent. *AI and Law*, 26(1):1–22, 2018.

[22] Edwina L Rissland and Kevin D Ashley. A case-based system for Trade Secrets law. In *Proceedings of the 1st International Conference on Artificial Intelligence and Law*, pages 60–66, 1987.

[23] Katharina Stevens. Reasoning by precedent—between rules and analogies. *Legal Theory*, pages 1–39, 2018.

[24] Douglas Walton. *Argumentation schemes for presumptive reasoning*. Lawrence Erlbaum Ass., 1996.

[25] Douglas Walton. Similarity, precedent and argument from analogy. *AI and Law*, 18(3):217–246, 2010.

[26] Douglas Walton, Christopher Reed, and Fabrizio Macagno. *Argumentation schemes*. CUP, 2008.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220143

Serialisable Semantics for Abstract Argumentation

Lars BENGEL and Matthias THIMM
Artificial Intelligence Group, University of Hagen, Germany

Abstract. We investigate the recently proposed notion of *serialisability* of semantics for abstract argumentation frameworks. This notion describes semantics where the construction of extensions can be serialised through iterative addition of minimal non-empty admissible sets. We investigate general relationships between serialisability and other principles from the literature. We also investigate the novel *unchallenged semantics* as a new instance of a serialisable semantics and, in particular, analyse it in terms of satisfied principles and computational complexity.

Keywords. abstract argumentation, serialisability, principles, computational complexity

1. Introduction

Abstract argumentation frameworks [1] are a simple, yet powerful formalism for representing argumentative scenarios and investigating matters regarding the acceptability of arguments. They consist simply of a set of arguments and an attack relation between arguments and can thus be represented as a directed graph. Abstract argumentation semantics [2] are used to interpret abstract argumentation frameworks by appropriately constraining the space of possible outcomes of the underlying argumentation. In particular, extension-based semantics define when a set of arguments (called extension) represents a plausible constellation of arguments that makes “sense” given the attacks in a framework. While we also consider extension-based semantics in this paper, it is noteworthy to mention that there are also other approaches for semantics such as the labelling-based approach [3], ranking and gradual semantics [4], and probabilistic approaches [5].

In [6] it has been shown that many of the mainstream extension-based semantics can be *serialised*, meaning that there is non-deterministic construction principle that allows to iteratively construct extensions by selecting minimal non-empty admissible sets—called *initial sets* [7]—and moving to the reduct [8]. Individual semantics can be distinguished by the way they select initial sets (via a so-called *selection function*) and how they terminate the construction (via a so-called *termination function*). For example, preferred semantics can be serialised by selecting initial sets arbitrarily until no further initial sets can be found [6]. Satisfaction of this principle of serialisability by a semantics allows a deeper inspection of the reasons why certain arguments are contained in an extension and therefore facilitates the explanatory power of an argumentation semantics [9,10]. In this paper, we aim at a better understanding of the principle of serialisability, in particular with respect to its connections to other principles for argumentation semantics

[11,12]. As it turns out, serialisability is an independent principle that is neither implied by nor implies other similar properties such as directionality.

As it has already been mentioned above, a serialisable semantics is characterised by a selection and termination function. This parametrisation of a semantics allows the easy development of further semantics simply by defining those two components. In [6], a specific candidate for such a semantics has already been suggested, which we coin here the *unchallenged semantics*. This semantics is defined by exhaustively adding *unattacked* and *unchallenged* initial sets (the formal definitions of these terms will be introduced in Section 3) and it has some interesting connections to preferred and ideal semantics. We investigate unchallenged semantics in more depth, in particular wrt. its compliance to principles from [11,12,13] and in terms of computational complexity. As with regard to the latter, unchallenged semantics turns out to be highly intractable, with credulous and skeptical reasoning shown to be Σ_2^P- and Π_2^P-complete, respectively.

To summarise, the contributions of this paper are as follows.

1. We recall the principle of serialisability and analyse its relationship with other principles (Section 3).
2. We investigate unchallenged semantics as a new instance of a serialisable semantics wrt. to its compliance to principles (Section 4).
3. We analyse unchallenged semantics wrt. computational complexity (Section 5).

Section 2 presents the necessary background on abstract argumentation and Section 6 concludes. Proofs of technical results are omitted due to space restrictions but can be found in an online appendix.[1]

2. Preliminaries

Let $\mathfrak{A}$ denote a universal set of arguments. An *abstract argumentation framework* AF is a tuple $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ where $\mathsf{A} \subseteq \mathfrak{A}$ is a finite set of arguments and R is a relation $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$ [1]. Let $\mathfrak{AF}$ denote the set of all abstract argumentation frameworks. For two arguments $a,b \in \mathsf{A}$, the relation $a\mathsf{R}b$ means that argument a attacks argument b. For $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ and $\mathsf{AF}' = (\mathsf{A}',\mathsf{R}')$ we write $\mathsf{AF}' \sqsubseteq \mathsf{AF}$ iff $\mathsf{A}' \subseteq \mathsf{A}$ and $\mathsf{R}' = \mathsf{R} \cap (\mathsf{A}' \times \mathsf{A}')$. For a set $X \subseteq \mathsf{A}$, we denote by $\mathsf{AF}|_X = (X, \mathsf{R} \cap (X \times X))$ the projection of AF on X. For a set $S \subseteq \mathsf{A}$ we define

$$S^+_{\mathsf{AF}} = \{a \in \mathsf{A} \mid \exists b \in S : b\mathsf{R}a\} \qquad\qquad S^-_{\mathsf{AF}} = \{a \in \mathsf{A} \mid \exists b \in S : a\mathsf{R}b\}$$

If S is a singleton set, we omit brackets for readability, i. e., we write a^-_{AF} (a^+_{AF}) instead of $\{a\}^-_{\mathsf{AF}}$ ($\{a\}^+_{\mathsf{AF}}$). For two sets S and S' we write $S\mathsf{R}S'$ iff $S' \cap S^+_{\mathsf{AF}} \neq \emptyset$. We say that a set $S \subseteq \mathsf{A}$ is *conflict-free* if for all $a,b \in S$ it is not the case that $a\mathsf{R}b$. A set S *defends* an argument $b \in \mathsf{A}$ if for all a with $a\mathsf{R}b$ there is $c \in S$ with $c\mathsf{R}a$. A conflict-free set S is called *admissible* if S defends all $a \in S$. Let $\mathsf{adm}(\mathsf{AF})$ denote the set of admissible sets of AF.

Different semantics can be phrased by imposing constraints on admissible sets [2]. In particular, an admissible set E

- is a *complete* (co) extension iff for all $a \in \mathsf{A}$, if E defends a then $a \in E$,
- is a *grounded* (gr) extension iff E is complete and minimally so,

[1] http://mthimm.de/misc/lbmt_uncsem_proofs.pdf

- is a *stable* (st) extension iff $E \cup E^{+}_{\mathsf{AF}} = \mathsf{A}$,
- is a *preferred* (pr) extension iff E is maximal.
- is a *semi-stable* (sst) extension iff $E \cup E^{+}_{\mathsf{AF}}$ is maximal,
- is an *ideal* (id) extension iff E is the maximal admissible set with $E \subseteq E'$ for each preferred extension E'.
- is a *strongly admissible* (sa) extension iff $E = \emptyset$ or each $a \in E$ is defended by some strongly admissible $E' \subseteq E \setminus \{a\}$.

All statements on minimality/maximality are meant to be with respect to set inclusion. For $\sigma \in \{\mathsf{co}, \mathsf{gr}, \mathsf{st}, \mathsf{pr}, \mathsf{sst}, \mathsf{id}, \mathsf{sa}\}$ let $\sigma(\mathsf{AF})$ denote the set of σ-extensions of AF.

3. Initial Sets and Serialisability

Non-empty minimal admissible sets have been coined *initial sets* by Xu and Cayrol [7].

Definition 1. For $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, a set $S \subseteq \mathsf{A}$ with $S \neq \emptyset$ is called an *initial set* if S is admissible and there is no admissible $S' \subsetneq S$ with $S' \neq \emptyset$. Let $\mathsf{IS}(\mathsf{AF})$ denote the set of initial sets of AF.

Initial sets are not supposed to be used to solve the whole argumentation represented in an argumentation framework, but rather a single atomic conflict within the framework. We can also differentiate between three types of initial sets [6].

Definition 2. For $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $S \in \mathsf{IS}(\mathsf{AF})$, we say that

1. S is *unattacked* iff $S^{-} = \emptyset$,
2. S is *unchallenged* iff $S^{-} \neq \emptyset$ and there is no $S' \in \mathsf{IS}(\mathsf{AF})$ with $S'\mathsf{R}S$,
3. S is *challenged* iff there is $S' \in \mathsf{IS}(\mathsf{AF})$ with $S'\mathsf{R}S$.

In the following, we will denote with $\mathsf{IS}^{\not\leftarrow}(\mathsf{AF})$, $\mathsf{IS}^{\not\leftrightarrow}(\mathsf{AF})$, and $\mathsf{IS}^{\leftrightarrow}(\mathsf{AF})$ the set of unattacked, unchallenged, and challenged initial sets, respectively.

In [6] the notion of *serialisability* has been introduced as a new approach for constructing admissible sets (and extensions of a variety of semantics) iteratively using initial sets. This approach relies also on the notion of the reduct [8].

Definition 3. For $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $S \subseteq \mathsf{A}$, the S-reduct AF^{S} is defined via $\mathsf{AF}^{S} = \mathsf{AF}|_{A \setminus (S \cup S^{+})}$.

The idea behind the approach of [6] to construct admissible sets is quite simple: We solve an atomic conflict in AF by selecting an initial set S. Afterwards, we move to the reduct AF^{S} which may reveal further conflicts and therefore new initial sets. This process is continued until some termination criterion is satisfied. In order to formalise this idea, we need a way to select initial sets in each step and also a criterion for determining if the construction of an admissible set is finished. The following concepts have been defined for this purpose.

Definition 4. A *state* T is a tuple $T = (\mathsf{AF}, S)$ with $\mathsf{AF} \in \mathfrak{AF}$ and $S \subseteq \mathfrak{A}$.

Definition 5. A *selection function* α is any function $\alpha : 2^{2^{\mathfrak{A}}} \times 2^{2^{\mathfrak{A}}} \times 2^{2^{\mathfrak{A}}} \to 2^{2^{\mathfrak{A}}}$ with $\alpha(X, Y, Z) \subseteq X \cup Y \cup Z$ for all $X, Y, Z \subseteq 2^{\mathfrak{A}}$.

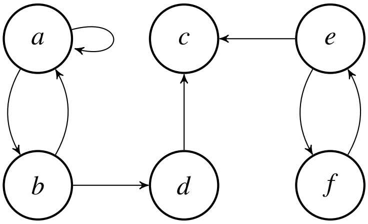


Figure 1. The argumentation framework AF_1 from Example 1.

We will apply a selection function α in the form $\alpha(\mathsf{IS}^{\not\leftarrow}(\mathsf{AF}),\mathsf{IS}^{\not\leftrightarrow}(\mathsf{AF}),\mathsf{IS}^{\leftrightarrow}(\mathsf{AF}))$ (for some AF), so α selects a subset of the initial sets as eligible to be selected in the construction process. We explicitly differentiate the different types of initial sets as parameters here as a technical convenience.

Definition 6. A *termination function* β is any function $\beta : \mathfrak{AF} \times 2^{\mathfrak{A}} \to \{0,1\}$.

A termination function β is used to indicate when a construction of an admissible set is finished (this will be the case if $\beta(\mathsf{AF},S) = 1$).

For some selection function α, consider the following transition rule:

$$(\mathsf{AF},S) \xrightarrow{S' \in \alpha(\mathsf{IS}^{\not\leftarrow}(\mathsf{AF}),\mathsf{IS}^{\not\leftrightarrow}(\mathsf{AF}),\mathsf{IS}^{\leftrightarrow}(\mathsf{AF}))} (\mathsf{AF}^{S'},S \cup S')$$

If (AF',S') can be reached from (AF,S) via a finite number of steps (this includes no steps at all) with the above rule we write $(\mathsf{AF},S) \leadsto^{\alpha} (\mathsf{AF}',S')$. If, in addition, the state (AF',S') also satisfies the termination criterion of β, i. e., $\beta(\mathsf{AF}',S) = 1$, then we write $(\mathsf{AF},S) \leadsto^{\alpha,\beta} (\mathsf{AF}',S')$.

Given concrete instances of α and β, let $\mathscr{E}^{\alpha,\beta}(\mathsf{AF})$ be the set of all S with $(\mathsf{AF},\emptyset) \leadsto^{\alpha,\beta} (\mathsf{AF}',S)$ (for some AF').

Definition 7. A semantics σ is *serialisable* if there exists a selection function α and a termination function β with $\sigma(\mathsf{AF}) = \mathscr{E}^{\alpha,\beta}(\mathsf{AF})$ for all AF. Then σ is also called the α,β-semantics.

In [6] it has already been shown that all of the standard admissible-based semantics adm, co, gr, pr and st as well as sa are serialisable. On the other hand, the semi-stable, ideal and eager semantics are not serialisable.

Example 1. As shown in [6], the preferred semantics can be serialised by the selection function $\alpha_{ad}(X,Y,Z) = X \cup Y \cup Z$ and the termination function

$$\beta_{pr}(\mathsf{AF},S) = \begin{cases} 1 \text{ if } \mathsf{IS}(\mathsf{AF}) = \emptyset \\ 0 \text{ otherwise} \end{cases}$$

Consider the argumentation framework AF_1 in Figure 1. The initial sets of AF_1 are $\{b\}$, $\{e\}$ and $\{f\}$. In order to obtain the preferred extensions we start with the state $(\mathsf{AF}_1,\emptyset)$. According to the α_{ad} all three initial sets can be selected. Assume we select $\{b\}$ first, then we apply the transition rule as

$$(\mathsf{AF}_1,\emptyset) \xrightarrow{\{b\}} (\mathsf{AF}_1^{\{b\}},\{b\}).$$

In this reduct $\mathsf{AF}_1^{\{b\}}$, we have two initial sets, namely $\{e\}$ and $\{f\}$. If we select $\{f\}$, the next transition would be

$$(\mathsf{AF}_1^{\{b\}},\{b\}) \xrightarrow{\{f\}} (\mathsf{AF}_1^{\{b,f\}},\{b,f\}).$$

This leaves us with one more possible transition via

$$(\mathsf{AF}_1^{\{b,f\}},\{b,f\}) \xrightarrow{\{c\}} ((\emptyset,\emptyset),\{b,c,f\}).$$

Now, trivially, the termination function is true, since there is no initial set for the empty framework and $\{b,c,f\}$ is a preferred extension of AF_1. Similarly, we could have selected $\{e\}$ in the state $(\mathsf{AF}_1^{\{b\}},\{b\})$. In that case, we also obtain an empty argumentation framework and the set $\{b,e\}$, which is the only other preferred extension of AF_1.

The principle of serialisability allows to define a semantics simply by specifying a selection function for initial sets and a termination function. In Section 4 we will define a completely new semantics using this approach and investigate its properties. However, in the remainder of this section we will analyse the principle of serialisability a bit deeper.

3.1. Relationship to Other Principles

In the following, we will look further at the serialisability principle and investigate its relationship with other principles from the literature [11]. First, we recall some basic definitions. Let $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ be an argumentation framework. A set of arguments $U \subseteq \mathsf{A}$ is called *unattacked* if and only if $\nexists a \in (\mathsf{A} \setminus U) : a\mathsf{R}U$. The set of unattacked sets of AF is denoted as $\mathfrak{US}(\mathsf{AF})$. Furthermore, a set $S \subseteq \mathsf{A}$ is a *strongly connected component* of AF, if there is a directed path between any pair $a,b \in S$ in AF and there is no $S' \supset S$ with that property. Let $\mathsf{SCCs}_{\mathsf{AF}}$ be the set of strongly connected components of AF. For a set $S \subseteq \mathsf{A}$, we define $\mathrm{op}_{\mathsf{AF}}(S) = \{a \in \mathsf{A} \mid a \notin S \wedge a\mathsf{R}S\}$. In order to define the principle of *SCC-Recursiveness* [14], we need some additional concepts.

Definition 8. Given an argumentation framework $\mathsf{AF} = (\mathsf{A},\mathsf{R})$, a set $E \subseteq \mathsf{A}$ and a strongly connected component $S \in \mathsf{SCCs}_{\mathsf{AF}}$, we define:

- $D_{\mathsf{AF}}(S,E) = \{a \in S \mid (\mathrm{op}_{\mathsf{AF}}(S))\mathsf{R}a\}$,
- $P_{\mathsf{AF}}(S,E) = \{a \in S \mid (E \cap \mathrm{op}_{\mathsf{AF}}(S)) \not\mathsf{R} a \wedge \exists b \in (\mathrm{op}_{\mathsf{AF}}(S) \cap a^-_{\mathsf{AF}}) : E \not\mathsf{R} b\}$,
- $U_{\mathsf{AF}}(S,E) = S \setminus (D_{\mathsf{AF}}(S,E) \cup P_{\mathsf{AF}}(S,E))$.

Definition 9. Let $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ be an argumentation framework and $C \subseteq \mathsf{A}$ is a set of arguments.

1. A function $\mathscr{BF}(\mathsf{AF},C)$ is called *base function*, if, given an argumentation framework $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ such that $|\mathsf{SCCs}(\mathsf{AF})| = 1$ and a set $C \subseteq \mathsf{A}$, $\mathscr{BF}(\mathsf{AF},C) \subseteq 2^{\mathsf{A}}$.
2. Given a base function $\mathscr{BF}(\mathsf{AF},C)$ we define the function $\mathscr{GF}_{\mathscr{BF}}(\mathsf{AF},C) \subseteq 2^{\mathsf{A}}$ as follows: for any $E \subseteq \mathsf{A}, E \in \mathscr{GF}(\mathsf{AF},\mathscr{C})$ if and only if
   - in case $|\mathsf{SCCs}_{\mathsf{AF}}| = 1$, $E \in \mathscr{BF}(\mathsf{AF},C)$,
   - otherwise, $\forall S \in \mathsf{SCCs}_{\mathsf{AF}} : E \cap S \in \mathscr{GF}_{\mathscr{BF}}(\mathsf{AF}|_{S \setminus D_{\mathsf{AF}}(S,E)}, U_{\mathsf{AF}}(S,E) \cap C)$.

Definition 10. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an argumentation framework and $E \subseteq \mathsf{A}$ is a set of arguments. We say that an argument $a \in \mathsf{A}$ is *strongly defended* by E (denoted as $sd(a,E)$) iff $\forall b \in \mathsf{A} : b\mathsf{R}a \implies \exists c \in E \setminus \{a\} : c\mathsf{R}b$ and $sd(c, E \setminus \{a\})$.

Finally, we recall the definitions of the different principles from the literature, that we considered in our analysis.

Definition 11. A semantics σ satisfies the principle of:

- *conflict-freeness [11]*, iff for every AF AF, every $E \in \sigma(\mathsf{AF})$ is conflict-free with respect to the attack relation.
- *admissibility [11]*, iff for every AF AF, every $E \in \sigma(\mathsf{AF})$ is conflict-free and defends itself in AF.
- *strong admissibility [11]*, iff for every AF AF, for every $E \in \sigma(\mathsf{AF})$ it holds that $a \in E$ implies that E strongly defends a.
- *reinstatement [11]*, iff for every AF $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $E \in \sigma(\mathsf{AF})$ we have: if E defends some $a \in \mathsf{A}$ then $a \in E$.
- *naivety [11]*, iff for every AF $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $E \in \sigma(\mathsf{AF})$ we have: E conflict-free and maximal among $cf(\mathsf{AF})$.
- *allowing abstention [15]*, iff for every AF AF and for every $a \in \mathsf{A}$, if there exist two extensions $E_1, E_2 \in \sigma(\mathsf{AF})$ such that $a \in E_1$ and $a \in E_2^+$ then there exists an extension $E_3 \in \sigma(\mathsf{AF})$ such that $a \notin (E_3 \cup E_3^+)$.
- *I-maximality [11]*, iff for every AF AF and $E_1, E_2 \in \sigma(\mathsf{AF}), E_1 \subseteq E_2 \implies E_1 = E_2$.
- *SCC-recursiveness [14]*, iff there is a base function $\mathscr{BF}_\sigma$ such that for every AF $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ we have that $\sigma(\mathsf{AF}) = \mathscr{GF}_{\mathscr{BF}_\sigma}(\mathsf{AF}, \mathsf{A})$.
- *directionality [11]*, iff for every AF $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $\forall U \in \mathfrak{US}(\mathsf{AF})$ we have that $\sigma(\mathsf{AF}, U) = \sigma(\mathsf{AF}|_U)$ with $\sigma(\mathsf{AF}, U) = \{E \cap U \mid E \in \sigma(\mathsf{AF})\}$.
- *modularization [16]*, iff for every AF AF we have: $E_1 \in \sigma(\mathsf{AF})$ and $E_2 \in \sigma(\mathsf{AF}^{E_1})$ implies $E_1 \cup E_2 \in \sigma(\mathsf{AF})$.
- *reduct-admissibility [13]*, iff for every AF AF and $E \in \sigma(\mathsf{AF})$, we have that $\forall a \in E$: if b attacks a then $b \notin \bigcup \sigma(\mathsf{AF}^E)$.
- *semi-qualified-admissibility [13]*, iff for every AF AF and $E \in \sigma(\mathsf{AF})$, we have that $\forall a \in E$, if b attacks a and $b \in \bigcup \sigma(\mathsf{AF})$ then $\exists c \in E$ s.t. c attacks b.

The principle of serialisability is intrinsically linked with admissibility since the building blocks of constructed extensions are the initial sets of an argumentation framework. By design, every extension constructed by the transition system for some α and β satisfies admissibility and thus also conflict-freeness. In other words, admissibility and conflict-freeness are necessary criteria for serialisability. Interestingly, the recently introduced principle of modularization [16] is also implied by serialisability.

Two of the more prominent principles from the literature are directionality and SCC-recursiveness. Like serialisability, the SCC-recursiveness principle can also be used to characterise existing semantics or define new semantics [14]. That raises the question if there exists a connection between these principles.

Interestingly, the principles of directionality and serialisability are independent of each other. The same holds true for SCC-recursiveness. While the above mentioned serialisable semantics are all SCC-recursive, the unchallenged semantics, which is investigated further in the following section, is not SCC-recursive. The relevant results are summarised in the following theorem.

Theorem 1. *Let σ be any semantics.*

- *If σ satisfies serialisability then it satisfies conflict-freeness.*
- *If σ satisfies serialisability then it satisfies admissibility.*
- *If σ satisfies serialisability then it satisfies modularization.*
- *Directionality does not imply serialisability and vice versa.*
- *SCC-recursiveness does not imply serialisability and vice versa.*

For all other mentioned principles, we could not find any relationships to serialisability. We will now take a closer look on the principle of directionality.

3.2. A Closer Look on Directionality

We now specify some additional property called $\alpha\beta$*-closure* that allows us to relate serialisability and directionality. This property captures whether or not every path of the transition system for α_σ and β_σ of the semantics σ eventually terminates for all argumentation frameworks $\mathsf{AF} \in \mathfrak{AF}$, i. e., every path leads to some σ-extension of AF.

Definition 12. Let σ be serialisable with α_σ and β_σ. We say that σ is $\alpha\beta$*-closed* for all argumentation frameworks $\mathsf{AF} \in \mathfrak{AF}$ if and only if, for every state (AF', S') with $(\mathsf{AF}, \emptyset) \leadsto^{\alpha_\sigma} (\mathsf{AF}', S')$ we have that, there exists some $\mathsf{AF}'' \in \mathfrak{AF}$ and some $S'' \subseteq \mathfrak{A}$ such that $(\mathsf{AF}', S') \leadsto^{\alpha_\sigma, \beta_\sigma} (\mathsf{AF}'', S'')$.

The property of $\alpha\beta$-closure is satisfied by most of the existing serialisable semantics. Only the transition system for the stable semantics does not terminate for all paths. Due to space limitations we do not recall the corresponding selection and termination functions but we refer to [6].

Theorem 2. *The* adm*,* co*,* gr*,* pr *and* sa *semantics are $\alpha\beta$-closed, while the* st *semantics is not, wrt. the selection and termination functions defined in [6].*

The fact that stable semantics is not closed wrt. its transition system is no coincidence since it is also the only semantics of the above that is not directional. In fact, if a semantics σ is serialisable and also $\alpha\beta$-closed, then it follows that σ must also be directional.

Theorem 3. *If a semantics σ is serialisable via α_σ and β_σ and is $\alpha_\sigma\beta_\sigma$-closed, then σ satisfies directionality.*

4. Unchallenged Semantics

The notion of serialisability allows to define completely new semantics by defining only a selection and a termination function. One aspect behind the initial sets is that they represent sets of arguments that solve a local conflict. We also have the differentiation between unattacked, unchallenged, and challenged initial sets, essentially distinguishing how convincing these sets solve their local conflict. In general, the grounded semantics can be considered to represent a minimal consensus, i. e., a set of arguments that everyone can agree on. The serialised characterisation of the grounded semantics shows us that this

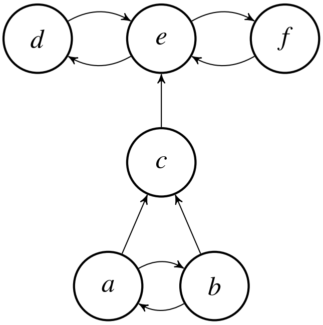


Figure 2. The argumentation framework AF_2 from Example 2.

is achieved by only considering unattacked initial sets in the selection function σ_{gr}. This is formalised by the selection function $\alpha_{gr}(X,Y,Z) = X$ and the termination function

$$\beta_{gr}(\mathsf{AF},S) = \begin{cases} 1 \text{ if } \mathsf{IS}^{\not\leftarrow}(\mathsf{AF}) = \emptyset \\ 0 \text{ otherwise} \end{cases}$$

However, from the perspective of local conflicts, an unchallenged initial set S also resolves its conflict while being uncontested by any acceptable argument. Therefore, it is reasonable to accept the arguments in an unchallenged initial set S as part of a consensus, since there exists no competing acceptable solution to the conflict S is concerned with. A natural approach to address this concern would now be to consider a semantics which allows for unattacked as well as unchallenged initial sets to be selected until no further unattacked or unchallenged initial sets exist. This means, we do not allow challenged initial sets to be included since there is at least one other set of arguments that solves the same local conflict, i. e., there is no consensual solution to this conflict. This approach has already been suggested in [6] but we will now investigate it in-depth. The approach can be implemented by the selection function α_{uc} defined via

$$\alpha_{uc}(X,Y,Z) = X \cup Y$$

and the termination function β_{uc} defined via

$$\beta_{uc}(\mathsf{AF},S) = \begin{cases} 1 \text{ if } \mathsf{IS}^{\not\leftarrow}(\mathsf{AF}) \cup \mathsf{IS}^{\not\leftrightarrow}(\mathsf{AF}) = \emptyset \\ 0 \text{ otherwise} \end{cases}$$

Essentially, this approach amounts to exhaustively adding unattacked and unchallenged initial sets. In light of this aspect, we also call the α_{uc},β_{uc}-semantics the *unchallenged semantics* (uc) where $\mathsf{uc}(\mathsf{AF}) = \{E \mid (\mathsf{AF},\emptyset) \rightsquigarrow^{\alpha_{uc},\beta_{uc}} (\mathsf{AF}',E)\}$ denotes the set of *unchallenged extensions*.

Example 2. Consider AF_2 depicted in Figure 2. There are four preferred extensions E_1, E_2, E_3, and E_4 in AF_2 defined via

$$E_1 = \{a,e\} \qquad E_2 = \{a,d,f\} \qquad E_3 = \{b,e\} \qquad E_4 = \{b,d,f\}$$

while the grounded and ideal extensions are empty. However, there is one unchallenged extension $E_5 = \{d,f\}$. The reason for that is that both $\{d\}$ and $\{f\}$ are unchallenged initial sets in AF_2 (and once one is selected the other becomes an unattacked initial set of the respective reduct and can be selected as well).

The unchallenged semantics is more skeptical than the preferred semantics but less skeptical than the ideal semantics as has already been observed in [6].

Theorem 4. *For every $E \in \mathit{uc}(\mathsf{AF})$:*

1. *$E \subseteq E'$ for some preferred extension E' and*
2. *$E_{id} \subseteq E$ for the ideal extension E_{id}.*

Also clear is the following observation:

Proposition 1. *For every AF, $\mathit{uc}(\mathsf{AF}) \neq \emptyset$.*

In Definition 12 we introduced the property of $\alpha\beta$-closure for serialisable semantics. This property is also satisfied by the unchallenged semantics.

Theorem 5. *Unchallenged semantics is $\alpha_{uc}\beta_{uc}$-closed.*

In light of Theorem 3 this directly implies that the unchallenged semantics is directional. In addition to the above characterisation via the selection and termination functions, the unchallenged semantics can also be characterised in a different manner. The following theorem gives a recursive definition of the unchallenged semantics based on the notion of the reduct, but without use of the transition rule.

Theorem 6. *Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an abstract argumentation framework and $E \subseteq \mathsf{A}$. E is an unchallenged extension if and only if either*

- *$E = \emptyset$ and $\mathit{IS}^{\not\leftarrow} \cup \mathit{IS}^{\not\leftrightarrow}(\mathsf{AF}) = \emptyset$ or*
- *$E = E_1 \cup E_2$, $E_1 \in \mathit{IS}^{\not\leftarrow} \cup \mathit{IS}^{\not\leftrightarrow}(\mathsf{AF})$ and E_2 is an unchallenged extension in AF^{E_1}.*

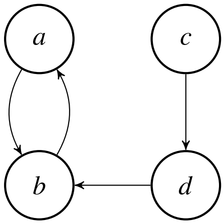


Figure 3. The argumentation framework AF_3 from Example 3.

Example 3. Consider the argumentation framework AF_3 in Figure 3. The initial sets of AF_3 are $\{c\}$ and $\{a\}$. Here, $\{c\}$ is unattacked and $\{a\}$ is considered unchallenged. Therefore, both sets are valid in terms of the selection function α_{uc}. Assume we select the set $\{c\}$, we transition to the framework $\mathsf{AF}_3^{\{c\}} = (\{a,b\}, \{(a,b),(b,a)\})$. In this argumentation framework we have two initial sets $\{a\}$ and $\{b\}$. Both of which are challenged by each other. This means, the termination function β_{uc} is satisfied and therefore $\{c\}$ is an unchallenged extension of AF_3.

On the other hand, if we select $\{a\}$ as the first transition, we arrive at the argumentation framework $\mathsf{AF}_3^{\{a\}} = (\{c,d\}, \{(c,d)\})$. Here, $\{c\}$ is the only initial set and it is

unattacked, just like it was in AF_3 itself. After the transition step, we obtain $\mathsf{AF}_3^{\{a,c\}} = (\emptyset,\emptyset)$, which means we are in a terminal state since we have that $\beta_{uc}(\mathsf{AF}_3^{\{a,c\}},\{a,c\}) = 1$.
All in all, $\{c\}$ and $\{a,c\}$ are the unchallenged extensions of AF_3.

In the following, we further investigate the compliance of the unchallenged semantics with principles from the literature. The unchallenged semantics satisfies conflict-freeness and admissibility by design. It also satisfies the more recently introduced principle of modularization as well as the reinstatement principle. Furthermore, the unchallenged semantics also satisfies the more complex principle of directionality.

Theorem 7. *The unchallenged semantics satisfies the following principles: Conflict-Freeness, Admissbility, Reduct Admissibility, Semi-Qualified Admissibility, Reinstatement, Directionality, Modularization and Serialisability.*

On the other hand, the unchallenged semantics does not satisfy strong admissibility. Like most admissible-based semantics, it does not satisfy the "allowing abstention" principle. As we have seen in Example 3 the unchallenged semantics does not satisfy I-maximality.

Interestingly, the SCC-recursiveness property is also not satisfied by this semantics. The reason for that stems from the inclusion of unchallenged initial sets. This allows for situations like in Example 3 where an unchallenged initial set can become challenged in some reduct of AF, but it can still be part of the extension if selected in an earlier transition step. Therefore, the unchallenged semantics serves as an example to show that not all serialisable semantics must necessarily be SCC-recursive.

Theorem 8. *The unchallenged semantics does not satisfy the following principles: Strong Admissibility, Naivety, Allowing Abstention, I-Maximality and SCC-Recursiveness.*

5. Computational Complexity

We assume familiarity with basic concepts of computational complexity and basic complexity classes such as P, NP, coNP, see [17] for an introduction. We also require knowledge of the classes Σ_2^P, Π_2^P, and $\mathsf{P}_{\|}^{\mathsf{NP}}$. The class $\Sigma_2^P = \mathsf{NP}^{\mathsf{NP}}$ is the class of decision problems that can be solved in polynomial time by a non-deterministic algorithm that has access to an NP-oracle, i. e., in every step of the algorithm it can immediately obtain the answer to an NP-complete problem. The class $\Pi_2^P = \mathsf{co}\Sigma_2^P = \mathsf{noNP}^{\mathsf{NP}}$ is the complement of Σ_2^P. The class $\mathsf{P}_{\|}^{\mathsf{NP}}$ [18] is the class of decision problems that can be solved by a deterministic polynomial-time algorithm that can make polynomially many *non-adaptive* (or *parallel*) queries to an NP-oracle. Note that $\mathsf{P}_{\|}^{\mathsf{NP}}$ is sometimes denoted by Θ_2^P and is equal to $\mathsf{P}^{\mathsf{NP}[log]}$, i. e., the class of decision problems solvable by a deterministic polynomial-time algorithm that can make logarithmically many *adaptive* NP-oracle calls [17].

We consider the following computational tasks, cf. [19]:

$\mathit{Ver}_{\mathsf{uc}}$ Given $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ and $E \subseteq \mathsf{A}$,
decide whether $E \in \mathsf{uc}(\mathsf{AF})$.
$\mathit{Exists}_{\mathsf{uc}}^{\neg\emptyset}$ Given $\mathsf{AF} = (\mathsf{A},\mathsf{R})$,

decide whether there is an $E \in \mathsf{uc}(\mathsf{AF})$ with $E \neq \emptyset$.

Skept$_{\mathsf{uc}}$ Given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $a \in \mathsf{A}$,
decide whether for all $E \in \mathsf{uc}(\mathsf{AF})$, $a \in E$.

Cred$_{\mathsf{uc}}$ Given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $a \in \mathsf{A}$,
decide whether there is $E \in \mathsf{uc}(\mathsf{AF})$ with $a \in E$.

Note that we do not consider the problem *Exists*$_{\mathsf{uc}}$, which asks whether some unchallenged extension exists, since this problem is trivial due to Proposition 1.

The results of our analysis are as follows.

Theorem 9.

1. *Ver*$_{uc}$ *is in* Σ_2^P *and* $\mathsf{P}_{||}^{\mathsf{NP}}$*-hard.*
2. *Exists*$_{uc}^{\neg\emptyset}$ *is* $\mathsf{P}_{||}^{\mathsf{NP}}$*-complete.*
3. *Skept*$_{uc}$ *is* Π_2^P*-complete.*
4. *Cred*$_{uc}$ *is* Σ_2^P*-complete.*

As can be seen, the exact computational complexity of the verification task is still an open problem (which is a bit surprising since we have exact characterisations for the more "complex" problems). However, all results are in line with our previous observation that unchallenged semantics is somehow "in-between" ideal and preferred semantics, cf. Theorem 4. While most tasks related to ideal semantics are $\mathsf{P}_{||}^{\mathsf{NP}}$-complete [20], skeptical reasoning with preferred semantics is Π_2^P-complete [21]. But in difference to preferred semantics, both skeptical and credulous reasoning is on the second level of the polynomial hierarchy for unchallenged semantics. As before, the proof of Theorem 9 can be found in the online appendix.[2] While the proofs of items 1 and 2 from Theorem 9 follow quite easily from existing results, in particular from [6], the hardness proofs of items 3 and 4 require quite a different reduction technique as, e. g., the Π_2^P-hardness proof for skeptical reasoning with preferred semantics [21].

6. Summary and Conclusion

We investigated the principle of *serialisability* in-depth, in particular wrt. its relationships to other principles from the literature [11,12,13]. While serialisability implies conflict-freeness, admissibility, and modularization, it is independent of similar principles like directionality and SCC-recursiveness. However, if a serialisable semantics is $\alpha\beta$-closed, it is also directional. We also analysed unchallenged semantics, a specific instance of a serialisable semantics, in terms of satisfied principles and computational complexity. This semantics is $\alpha_{uc}\beta_{uc}$-closed and thus directional. It also satisfies reinstatement, but interestingly it is not SCC-recursive, in contrast to all other serialisable semantics. We have also implemented a general serialisable reasoner as well as reasoners for all existing serialisable semantics[3].

In future work, we intent to further investigate serialisability. That includes defining and analysing completely new semantics with more sophisticated selection and termina-

[2] http://mthimm.de/misc/lbmt_uncsem_proofs.pdf

[3] Link to implementation: https://tinyurl.com/serialisableReasoner

tion functions. We will also consider applying the concept of serialisability to other types of semantics such as naive- or weak-admissible-based semantics. Regarding the unchallenged semantics, the question of whether there exists a non-recursive characterisation is also subject to future work.

Acknowledgements The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (grant 375588274).

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995;77(2):321-58.

[2] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018. p. 159-236.

[3] Caminada MWA, Gabbay DM. A Logical Account of Formal Argumentation. Studia Logica. 2009;93(2–3):109-45.

[4] Bonzon E, Delobelle J, Konieczny S, Maudet N. A Comparative Study of Ranking-based Semantics for Abstract Argumentation. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16); 2016. p. 914-20.

[5] Hunter A, Thimm M. Probabilistic Reasoning with Abstract Argumentation Frameworks. Journal of Artificial Intelligence Research. 2017 August;59:565-611.

[6] Thimm M. Revisiting initial sets in abstract argumentation. Argument & Computation. 2022 July.

[7] Xu Y, Cayrol C. Initial Sets in Abstract Argumentation Frameworks. In: Proceedings of the 1st Chinese Conference on Logic and Argumentation (CLAR'16). vol. 1811; 2016. p. 72-85.

[8] Baumann R, Brewka G, Ulbricht M. Revisiting the foundations of abstract argumentation–semantics based on weak admissibility and weak defense. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. p. 2742-9.

[9] Amgoud L, Prade H. Using arguments for making and explaining decisions. Artificial Intelligence. 2009 March;173(3-4):413-36.

[10] Rago A, Cocarascu O, Bechlivanidis C, Lagnado DA, Toni F. Argumentative explanations for interactive recommendations. Artificial Intelligence. 2021;296:103506.

[11] Baroni P, Giacomin M. On principle-based evaluation of extension-based argumentation semantics. In: Artificial Intelligence. vol. 171. Elsevier; 2007. p. 675-700.

[12] van der Torre L, Vesic S. The Principle-Based Approach to Abstract Argumentation Semantics. In: Handbook of formal argumentation, Vol. 1. College Publications; 2018. p. 2735-78.

[13] Dauphin J, Rienstra T, van der Torre L. A Principle-Based Analysis of Weakly Admissible Semantics. In: Proceedings of COMMA 2020; 2020. p. 167-78.

[14] Baroni P, Giacomin M, Guida G. SCC-recursiveness: a general schema for argumentation semantics. Artificial Intelligence. 2005;168(1–2):162-210.

[15] Baroni P, Caminada M, Giacomin M. An introduction to argumentation semantics. The knowledge engineering review. 2011;26(4):365-410.

[16] Baumann R, Brewka G, Ulbricht M. Comparing Weak Admissibility Semantics to their Dung-style Counterparts–Reduct, Modularization, and Strong Equivalence in Abstract Argumentation. In: Proceedings of KR 2020; 2020. p. 79-88.

[17] Papadimitriou C. Computational Complexity. Addison-Wesley; 1994.

[18] Eiter T, Gottlob G. The Complexity Class Theta2p: Recent Results and Applications in AI and Modal Logic. In: Proceedings of FCT'97. Berlin, Heidelberg; 1997. p. 1-18.

[19] Dvorák W, Dunne PE. Computational problems in formal argumentation and their complexity. In: Handbook of formal argumentation; 2018. p. 631-87.

[20] Dunne PE. The computational complexity of ideal semantics. Artificial Intelligence. 2009 December;173(18):1559-91.

[21] Dunne PE, Bench-Capon TJM. Coherence in finite argument systems. Artificial Intelligence. 2002;141(1/2):187-203. Available from: https://doi.org/10.1016/S0004-3702(02)00261-8.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220144

Abstract Argumentation with Conditional Preferences

Michael BERNREITER [1], Wolfgang DVOŘÁK and Stefan WOLTRAN
Institute of Logic and Computation, TU Wien, Austria

Abstract. In this paper, we study conditional preferences in abstract argumentation by introducing a new generalization of Dung-style argumentation frameworks (AFs) called Conditional Preference-based AFs (CPAFs). Each subset of arguments in a CPAF can be associated with its own preference relation. This generalizes existing approaches for preference-handling in abstract argumentation, and allows us to reason about conditional preferences in a general way. We conduct a principle-based analysis of CPAFs and compare them to related generalizations of AFs. Specifically, we highlight similarities and differences to Modgil's Extended AFs and show that our formalism can capture Value-based AFs.

Keywords. Abstract argumentation, conditional preferences, principles.

1. Introduction

Preferences in argumentation have been studied from various points of view, be it in terms of argument strength [1,2,3,4,5] or preferences between values [6,7]. Despite this, *conditional preferences* have received only limited attention in the field of argumentation. Dung et al. investigated conditional preferences in the setting of structured argumentation [8]. There, argumentation frameworks (AFs) are built from defeasible knowledge bases containing preference rules of the form $a_1,\dots,a_n \rightarrow d_0 \succ d_1$, where d_0 and d_1 are defeasible rules. However, no work that deals with conditional preferences on the abstract level is known to us. This is in contrast to unconditional preferences, which are studied both in structured [9,10,11] and abstract [5,7] argumentation in the literature.

Conditional preferences can appear in many situations and formalisms. Dung et al. [8] demonstrate this with the help of an example, which we now adapt:[2]

Example 1. *Sherlock Holmes is investigating a murder. There are two suspects, Person 1 and Person 2. After analyzing the crime scene, Sherlock is sure:*

- I_1*: Person 1 or Person 2 is the culprit, but not both.*

Moreover, Sherlock adheres to the following rules:

- R_1*: If Person i has a motive but Person j, with $j \neq i$, does not, then this supports the case that Person i is the culprit.*

[1] Corresponding Author: Michael Bernreiter; E-mail: michael.bernreiter@tuwien.ac.at.

[2] We specify the example in natural language. See [8] for how Dung's original example can be modeled as a defeasible knowledge base with conditional preferences. Our example can be formalized similarly.

- *R_2: If Person i has an alibi but Person j, with $j \neq i$, does not, then this supports the case that Person j is the culprit.*
- *R_3: Alibis have more importance than motives.*

After interrogating the suspects, Sherlock concludes that:

- *C_1: Person 1 has a motive but Person 2 does not.*
- *C_2: Person 1 has an alibi but Person 2 does not.*

If C_1 is trusted, but C_2 is not, then this supports that Person 1 is the culprit. If C_2 is trusted then this supports that Person 2 is the culprit, regardless of our stance on C_1.

This example demonstrates the importance of conditional preferences in common reasoning tasks. We believe it is valuable to capture conditional preferences in argumentation not only on the structured level as Dung et al. [8] did, but also on the abstract level. Doing so will generalize existing formalisms for unconditional preferences in abstract argumentation and provide a more direct target formalism for structured approaches.

To this end, we introduce Conditional Preference-based AFs (CPAFs), where each subset of arguments S can be associated with its own preference relation $\succ_S$. Preferences are then resolved via so-called preference-reductions [5], which modify the attack relation based on the given preferences. As a consequence, S must be justified in view of its own preferences, i.e., S must be an extension in view of $\succ_S$.

We show that CPAFs generalize Preference-based AFs (PAFs), and demonstrate that they are capable of dealing with conditional preferences in a general manner. Moreover, we conduct a principle-based analysis of CPAF-semantics and show that especially complete and stable semantics preserve desirable properties of regular PAFs. Lastly, we compare CPAFs to related formalisms. Specifically, we show that CPAFs can capture other generalizations of AFs such as Value-based AFs (VAFs) [6,7] in a straightforward way, and compare CPAFs to Extended Argumentation Frameworks (EAFs) [12,13,14] in order to highlight similarities and differences.

2. Preliminaries

We first define (abstract) argumentation frameworks [15].

Definition 1. *An argumentation framework (AF) is a tuple $F = (A, R)$ where A is a finite set of arguments and $R \subseteq A \times A$ is an attack relation between arguments. Let $S \subseteq A$. We say S* attacks *b (in F) if $(a,b) \in R$ for some $a \in S$; $S_F^+ = \{b \in A \mid \exists a \in S : (a,b) \in R\}$ denotes the set of arguments attacked by S. An argument $a \in A$ is* defended *(in F) by S if $b \in S_F^+$ for each b with $(b,a) \in R$.*

Semantics for AFs are defined as functions σ which assign to each AF $F = (A, R)$ a set $\sigma(F) \subseteq 2^A$ of extensions [16]. We consider for σ the functions *cf* (conflict-free), *adm* (admissible), *com* (complete), *stb* (stable), *grd* (grounded), and *prf* (preferred).

Definition 2. *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is conflict-free (in F), written as $S \in cf(F)$, if there are no $a, b \in S$, such that $(a,b) \in R$. For $S \in cf(F)$ it holds that*

- *$S \in adm(F)$ if each $a \in S$ is defended by S in F;*
- *$S \in com(F)$ if $S \in adm(F)$ and each $a \in A$ defended by S in F is contained in S;*

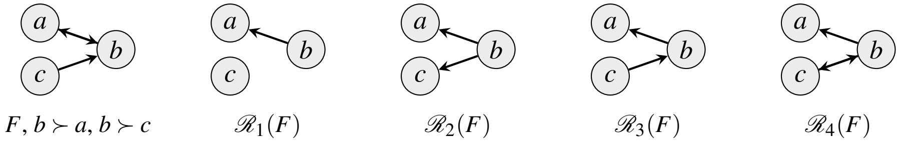


Figure 1. PAF F and its preference reducts from Example 2.

- *$S \in stb(F)$ if each $a \in A \setminus S$ is attacked by S in F;*
- *$S \in grd(F)$ if $S \in com(F)$ and there is no $T \in com(F)$ with $T \subset S$;*
- *$S \in prf(F)$ if $S \in adm(F)$ and there is no $T \in adm(F)$ with $S \subset T$;*

Preference-based AFs enrich regular AFs with preferences between arguments.

Definition 3. *A preference-based AF (PAF) is a triple $F = (A, R, \succ)$ where (A, R) is an AF and $\succ$ is an irreflexive and asymmetric binary relation over A.*

If a and b are arguments and $a \succ b$ holds then we say that a is stronger than b. An established method of resolving preferences in PAFs are so-called preference reductions, of which there exist four in the literature [5]. If in a PAF $(A, R, \succ)$ there is an attack $(a, b) \in R$ and a preference $b \succ a$ then (a, b) is called a critical attack. In other words, critical attacks are from weak to strong arguments. The preference-reductions resolve preferences by dealing with these critical attacks, e.g., by removing or reverting them.

Definition 4. *Given a PAF $F = (A, R, \succ)$, a corresponding AF $\mathscr{R}_i(F) = (A, R')$ is constructed via Reduction i, where $i \in \{1,2,3,4\}$, as follows:*

- *$i = 1$: $\forall a, b \in A : \ (a,b) \in R' \Leftrightarrow (a,b) \in R, b \not\succ a$*
- *$i = 2$: $\forall a, b \in A : \ (a,b) \in R' \Leftrightarrow ((a,b) \in R, b \not\succ a)$ or $((b,a) \in R, (a,b) \notin R, a \succ b)$*
- *$i = 3$: $\forall a, b \in A : \ (a,b) \in R' \Leftrightarrow ((a,b) \in R, b \not\succ a)$ or $((a,b) \in R, (b,a) \notin R)$*
- *$i = 4$: $\forall a, b \in A : \ (a,b) \in R' \Leftrightarrow ((a,b) \in R, b \not\succ a)$ or $((b,a) \in R, (a,b) \notin R, a \succ b)$ or $((a,b) \in R, (b,a) \notin R)$*

The preference-based variant of a semantics σ relative to Reduction i is defined as $\sigma_p^i(F) = \sigma(\mathscr{R}_i(F))$.

Intuitively, Reduction 1 removes critical attacks while Reduction 2 reverts them. Reduction 3 removes critical attacks, but only if the stronger argument also attacks the weaker one. Reduction 4 can be seen as a combination of Reduction 2 and 3. Note that on symmetric attacks, all four reductions function in the same way. The following example demonstrates the reductions and PAF-semantics.

Example 2. *Consider the PAF $F = (\{a,b,c\}, \{(a,b),(b,a),(c,b)\}, \succ)$ with $b \succ a$ and $b \succ c$. Figure 1 depicts F as well as $\mathscr{R}_i(F)$, $i \in \{1,2,3,4\}$. It can be checked that, for Reduction 1, $adm_p^1(F) = adm(\mathscr{R}_i(F)) = \{\emptyset, \{b\}, \{c\}, \{b,c\}\}$ and therefore $com_p^1(F) = prf_p^1(F) = stb_p^1(F) = \{\{b,c\}\}$. If we use Reduction 2 for example we get different extensions, namely $adm_p^2(F) = \{\emptyset, \{b\}\}$ and $com_p^2(F) = prf_p^2(F) = stb_p^2(F) = \{\{b\}\}$.*

A principle-based analysis of the four preference reductions was conducted for complete, grounded, preferred, and stable semantics [4,5]. To this end, the following six PAF-properties were laid out and investigated.

	$\mathscr{R}_1$	$\mathscr{R}_2$	$\mathscr{R}_3$	$\mathscr{R}_4$
$P1$	×	$CGPS$	$CGPS$	$CGPS$
$P2$	×	×	CS	×
$P3$	×	×	CS	×
$P4$	×	×	CGS	×
$P5$	×	×	CG	×
$P6$	G	G	$CGPS$	G

Table 1. Satisfaction of various PAF-principles. C stands for complete, G for grounded, P for preferred, and S for stable. × indicates that none of those four semantics satisfy this principle.

Definition 5. *Let σ_p^i be a PAF-semantics. Let $\succ' \subseteq (A \times A)$ be irreflexive and asymmetric.*

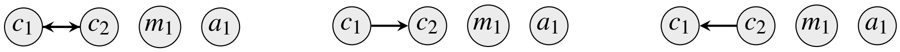
- *σ_p^i satisfies $P1$ (conflict-freeness) iff for all PAFs $F = (A, R, \succ)$ there is no $S \in \sigma_p^i(F)$ such that $\{a, b\} \subseteq S$ and $(a, b) \in R$.*
- *σ_p^i satisfies $P2$ (preference selects extensions 1) iff $\sigma_p^i(A, R, \succ \cup \succ') \subseteq \sigma_p^i(A, R, \succ)$ holds for all PAFs $(A, R, \succ)$ and all $\succ'$.*
- *σ_p^i satisfies $P3$ (preference selects extensions 2) iff $\sigma_p^i(A, R, \succ) \subseteq \sigma_p^i(A, R, \emptyset)$ holds for all PAFs $(A, R, \succ)$.*
- *σ_p^i satisfies $P4$ (extension refinement) iff for all $S' \in \sigma_p^i(A, R, \succ \cup \succ')$ there exists some $S \in \sigma_p^i(A, R, \succ)$ such that $S \subseteq S'$.*
- *σ_p^i satisfies $P5$ (extension growth) iff $\bigcap(\sigma_p^i(A, R, \succ)) \subseteq \bigcap(\sigma_p^i(A, R, \succ \cup \succ'))$ holds for all PAFs $(A, R, \succ)$ and all $\succ'$.*
- *σ_p^i satisfies $P6$ (number of extensions) iff $|\sigma_p^i(A, R, \succ \cup \succ')| \leq |\sigma_p^i(A, R, \succ)|$ holds for all PAFs $(A, R, \succ)$ and all $\succ'$.*

Intuitively, $P1$ states that if there is an attack between two arguments, then there is no extension containing both of them. $P2$ expresses that adding more preferences to a PAF can exclude extensions, but not introduce them. $P3$ is a special case of $P2$. $P4$ states that adding preferences means extensions will be supersets of extensions in the original PAF. $P5$ says that adding preferences will preserve skeptically accepted arguments, and might cause new arguments to be skeptically accepted. $P6$ expresses that the number of extensions will not grow if new preferences are added.

Table 1 shows which semantics satisfy which principle. In addition to these fundamental principles [4], four more principles were introduced later [5], but we do not consider them at this point and leave them for future work.

3. Conditional Preference-Based Argumentation Frameworks

As argued in the introduction, our aim is to provide a framework for reasoning about conditional preferences in abstract argumentation. This means that arguments themselves must be capable of expressing preferences, and that those argument-bound preferences are relevant only if the corresponding arguments are themselves accepted. How this is implemented must be considered carefully, as Example 1 demonstrates. There, the fact that Person 1 has a motive (let us refer to this as m_1) and the fact that Person 1 has an alibi (a_1) result in opposing preferences. When accepting both m_1 and a_1 it seems natural to combine these opposing preferences, i.e., to cancel them. But this does not allow us to express that alibis are more important than motives, as required in Example 1. Therefore,



F, $c_1 \succ_S c_2$ if $m_1 \in S$, $a_1 \notin S$ and $c_2 \succ_S c_1$ if $a_1 \in S$

$\mathscr{R}_i^S(F)$, $m_1 \in S$, $a_1 \notin S$

$\mathscr{R}_i^S(F)$, $a_1 \in S$

Figure 2. CPAF F and its preference-reducts from Example 3.

we need to define our formalism in a general way such that the joint acceptance of arguments must not necessarily result in the combination of their associated preferences. We solve this by mapping each subset S of arguments to a separate preference relation $\succ_S$.

Definition 6. *A Conditional PAF (CPAF) is a triple $F = (A, R, cond)$, where (A, R) is an AF and $cond\colon 2^A \to 2^{(A\times A)}$ is a function that maps each set of arguments $S \subseteq A$ to an irreflexive and asymmetric binary relation $\succ_S$ over A.*

Note that we set no restriction on how exactly conditional preferences are represented. This is deliberate, as we wish to stay as general as possible. In practice, succinct representations could be achieved, e.g., by expressing the *cond*-function via rules of the form $F \to x \succ y$ where F is a propositional formula over the arguments.

Just as in regular PAFs, preferences in CPAFs are resolved with the help of the four preference-reductions (cf. Definition 4). A set of arguments S is an extension of some CPAF if it is an extension relative to its associated preference relation $cond(S)$.

Definition 7. *Let $F = (A, R, cond)$ be a CPAF and let $S \subseteq A$. The S-reduct of F with respect to a preference reduction $i \in \{1,2,3,4\}$ is defined as $\mathscr{R}_i^S(F) = \mathscr{R}_i(A, R, cond(S))$. Given an AF semantics σ, $S \in \sigma_{cp}^i(F)$ iff $S \in \sigma(\mathscr{R}_i^S(F))$.*

Using CPAFs we can easily formalize our Sherlock Holmes example.

Example 3. *We continue Example 1 and introduce two arguments c_1 and c_2 expressing that Person 1 (resp. Person 2) is the culprit. Moreover, we introduce m_1 and a_1 to express that Person 1 has a motive (resp. alibi) but Person 2 does not. c_1 and c_2 attack each other while m_1 and a_1 have no incoming or outgoing attacks, but rather express preferences. Formally, we model this via the CPAF $F = (\{c_1, c_2, m_1, a_1\}, \{(c_1, c_2), (c_2, c_1)\}, cond)$ with cond such that $c_1 \succ_S c_2$ if $m_1 \in S$ but $a_1 \notin S$, $c_2 \succ_S c_1$ if $a_1 \in S$, and $cond(S) = \emptyset$ for all other $S \subseteq A$. Figure 2 depicts F and the S-reducts of F. Note that m_1 and a_1 are unattacked in all S-reducts of F. Therefore, both arguments must be part of any σ_{cp}^i-extension for $\sigma \in \{grd, com, prf, stb\}$ and we can conclude that $\sigma_{cp}^i(F) = \{\{m_1, a_1, c_2\}\}$.*

Note that the preferred semantics defined above do not maximize over all admissible sets of a CPAF, but rather over all admissible sets in the given S-reduct. This means that if there is a set S that is admissible in the S-reduct of F, but there is also some $T \supset S$ that is admissible in the S-reduct of F, then S is not preferred in F. But this T does not have to be admissible in F, since it might not be admissible in the T-reduct of F. Thus, the following alternative semantics may be considered more natural:

Definition 8. *Let $F = (A, R, cond)$ be a CPAF and let $S \subseteq A$. Then $S \in prf\text{-}glb_{cp}^i(F)$ iff $S \in adm_{cp}^i(F)$ and there is no T such that $S \subset T$ and $T \in adm_{cp}^i(F)$.*

Intuitively, $\mathit{prf}\text{-}\mathit{glb}^i_{cp}$ maximizes *globally* over all admissible sets of a CPAF, while prf^i_{cp} maximizes *locally* over the admissible sets of the given S-reduct.

Example 4. *Let F be the CPAF from Example 3 and recall that $\mathit{prf}^i_{cp}(F) = \{\{m_1, a_1, c_2\}\}$. Observe that $\{m_1, c_1\}$ is not preferred in the $\{m_1, c_1\}$-reduct of F, but it is a subset-maximal admissible set in F. Thus, $\mathit{prf}\text{-}\mathit{glb}^i_{cp}(F) = \{\{m_1, a_1, c_2\}, \{m_1, c_1\}\}$.*

The difference between the two variants is not only philosophical, but impacts fundamental properties for maximization-based semantics such as I-maximality. A semantics σ^i_{cp} is I-maximal if, for all CPAFs F and all $S,T \in \sigma^i_{cp}(F)$, $S \subseteq T$ implies $S = T$.

Proposition 1. *$\mathit{prf}\text{-}\mathit{glb}^i_{cp}$ is I-maximal, but prf^i_{cp} is not, where $i \in \{1,2,3,4\}$.*

Proof. I-maximality of $\mathit{prf}\text{-}\mathit{glb}^i_{cp}$ follows from Definition 8. For our counter examples we consider the preference-reductions separately. Reduction 1: consider the CPAF $F = (\{a,b\}, \{(a,b)\}, \mathit{cond}\}$ with cond such that $b \succ_{\{a,b\}} a$. Then $\{a\} \in \mathit{prf}^1_{cp}(F)$ and $\{a,b\} \in \mathit{prf}^1_{cp}(F)$. Reductions 2 and 4: consider the CPAF $F' = (\{a,b,c\}, \{(a,b),(b,c),(c,a)\}, \mathit{cond}\}$ with cond such that $a \succ_{\{a\}} c$. Then $\emptyset \in \mathit{prf}^i_{cp}(F')$ and $\{a\} \in \mathit{prf}^i_{cp}(F')$. Reduction 3: consider the CPAF $F'' = (\{a,b,c\}, \{(a,b),(b,a),(b,c),(c,a)\}, \mathit{cond}\}$ with cond such that $a \succ_{\emptyset} b$. Then $\emptyset \in \mathit{prf}^3_{cp}(F'')$ and $\{b\} \in \mathit{prf}^3_{cp}(F'')$. □

One may be tempted to deduce from the above proposition that $\mathit{prf}\text{-}\mathit{glb}^i_{cp}$ is more suitable as a default preferred semantics than prf^i_{cp}. However, we will see in Section 5.1 that prf^i_{cp} allows us to capture the problems of subjective/objective acceptance in VAFs in a natural way. In our subsequent analysis of CPAFs we consider both prf^i_{cp} and $\mathit{prf}\text{-}\mathit{glb}^i_{cp}$. Like preferred semantics, stable semantics satisfy I-maximality on regular AFs. Interestingly, on CPAFs, this depends on the preference-reduction.

Proposition 2. *stb^1_{cp} is not I-maximal, but stb^j_{cp} is, where $j \in \{2,3,4\}$.*

Proof. For stb^1_{cp} we can use the same counter-example as for prf^1_{cp} (cf. Proposition 1). For stb^j_{cp} with $j \in \{2,3,4\}$ we proceed by contradiction: assume there is a CPAF $F = (A, R, \mathit{cond})$ with $S,T \in \mathit{stb}^j_{cp}(F)$ such that $S \subset T$. Then there is $x \in T$ such that $x \notin S$. Since $S \in \mathit{stb}^j_{cp}(F)$ there is $y \in S$ such that $(y,x) \in \mathscr{R}^S_j(F)$. Reductions 2, 3, and 4 do not remove conflicts between arguments, and thus either $(y,x) \in R$ or $(x,y) \in R$. Therefore, $(y,x) \in \mathscr{R}^T_j(F)$ or $(x,y) \in \mathscr{R}^T_j(F)$. But $y \in S$ implies $y \in T$, i.e., $T \notin \mathit{cf}^j_{cp}(F)$. □

Another interesting point is that grounded extensions are not necessarily unique in CPAFs: consider $F = (\{a,b\}, \{(a,b)\}, \mathit{cond})$ with cond such that $b \succ_{\{b\}} a$. Then $\{a\} \in \mathit{grd}^2_{cp}(F)$ and $\{b\} \in \mathit{grd}^2_{cp}(F)$. We stress that each grounded extension S is still unique in the S-reduct of the given CPAF and thus unique with respect to its own preferences.

Lastly, by the following proposition we express that all CPAF-semantics considered here generalize their corresponding PAF-semantics, i.e., that CPAFs generalize PAFs.

Proposition 3. *Let $F = (A, R, \mathit{cond})$ be a CPAF such that the preference function cond maps every set of arguments to the same binary relation, i.e., there is some $\succ$ such that $\mathit{cond}(S) = \succ$ for all $S \subseteq A$. Let $\sigma \in \{\mathit{cf}, \mathit{adm}, \mathit{stb}, \mathit{com}, \mathit{prf}, \mathit{grd}\}$. Then $\sigma^i_{cp}(F) = \sigma^i_p(A, R, \succ)$. Furthermore, $\mathit{prf}\text{-}\mathit{glb}^i_{cp}(F) = \mathit{prf}^i_p(A, R, \succ)$.*

4. Principle-Based Analysis

In this section, we generalize the principles of Kaci et al. for PAFs (cf. Definition 5) to account for conditional preferences. We then investigate by which semantics these principles are satisfied, and show that there are differences to the case of regular PAFs.

Definition 9. *Let σ_{cp}^{i} be a CPAF-semantics. In the following, given a CPAF $(A,R,cond)$, we denote by $cond'$ an arbitrary function such that $cond(S) \subseteq cond'(S)$ for all $S \subseteq A$. Furthermore, $cond_{\emptyset}(S) = \emptyset$ for all $S \subseteq A$.*

- *σ_{cp}^{i} satisfies $P1^*$ (conflict-freeness) iff for all CPAFs $F = (A,R,cond)$ there is no $S \in \sigma_{cp}^{i}(F)$ such that $\{a,b\} \subseteq S$ and $(a,b) \in R$.*
- *σ_{cp}^{i} satisfies $P2^*$ (preference selects extensions) iff for all CPAFs $(A,R,cond)$ it holds that $\sigma_{cp}^{i}(A,R,cond') \subseteq \sigma_{cp}^{i}(A,R,cond)$.*
- *σ_{cp}^{i} satisfies $P3^*$ (preference selects extensions 2) iff for all CPAFs $(A,R,cond)$ it holds that $\sigma_{cp}^{i}(A,R,cond) \subseteq \sigma_{cp}^{i}(A,R,cond_{\emptyset})$.*
- *σ_{cp}^{i} satisfies $P4^*$ (extension refinement) iff for all $S' \in \sigma_{cp}^{i}(A,R,cond')$ there exists some $S \in \sigma_{cp}^{i}(A,R,cond)$ such that $S \subseteq S'$.*
- *σ_{cp}^{i} satisfies $P5^*$ (extension growth) iff for all CPAFs $(A,R,cond)$ it holds that $\bigcap(\sigma_{cp}^{i}(A,R,cond)) \subseteq \bigcap(\sigma_{cp}^{i}(A,R,cond'))$.*
- *σ_{cp}^{i} satisfies $P6^*$ (number of extensions) iff for all CPAFs $(A,R,cond)$ it holds that $|\sigma_{cp}^{i}(A,R,cond')| \leq |\sigma_{cp}^{i}(A,R,cond)|$.*

The following lemma establishes some relationships between the CPAF-principles and is a generalization of known relationships between PAF-principles [5].

Lemma 4. *If σ_{cp}^{i} satisfies $P2^*$ then it also satisfies $P3^*$, $P4^*$, and $P6^*$. If σ_{cp}^{i} always returns at least one extension, and if it satisfies $P2^*$, then it also satisfies $P5^*$.*

Observe that, since CPAFs are a generalization of PAFs (cf. Proposition 3), a CPAF-semantics σ_{cp}^{i} can not satisfy Pj^* if the corresponding PAF-semantics σ_{p}^{i} does not satisfy Pj. Moreover, it is obvious that $P1^*$ is still satisfied under Reductions 2, 3, and 4, as conflicts are not removed by these reductions even if we consider conditional preferences. We can also show that satisfaction of $P2$ carries over from PAFs to CPAFs.

Lemma 5. *If some σ_{p}^{i} satisfies P2 then σ_{cp}^{i} satisfies $P2^*$.*

Proof. Assume σ_{cp}^{i} does not satisfy $P2^*$. Then there is a CPAF $F = (A,R,cond)$ and $cond'$ with $cond(S) \subseteq cond'(S)$ for all $S \subseteq A$ such that $\sigma_{cp}^{i}(A,R,cond') \not\subseteq \sigma_{cp}^{i}(A,R,cond)$. Thus, there is $E \subseteq A$ such that $E \in \sigma_{cp}^{i}(A,R,cond')$ but $E \notin \sigma_{cp}^{i}(A,R,cond)$. Then $E \in \sigma(\mathscr{R}_i(A,R,cond'(E)))$ but $E \notin \sigma(\mathscr{R}_i(A,R,cond(E)))$, i.e., σ_{p}^{i} does not satisfy $P2$. □

Lemma 5 implies that complete and stable semantics satisfy $P2^*$ on CPAFs under Reduction 3. By Lemma 4 these semantics also satisfy $P3^*$, $P4^*$, and $P6^*$. However, we can not use Lemma 4 to show that complete semantics satisfy $P5^*$, since conditional preferences allow for frameworks without complete extensions. Indeed, we can find a counter-example in this case. Counter-examples for the satisfaction of various principles can also be found for grounded semantics and both variants of the preferred semantics.

	$\mathscr{R}_1$	$\mathscr{R}_2$	$\mathscr{R}_3$	$\mathscr{R}_4$
$P1^*$	×	*CGPS*	*CGPS*	*CGPS*
$P2^*$	×	×	*CS*	×
$P3^*$	×	×	*CS*	×
$P4^*$	×	×	*CS*	×
$P5^*$	×	×	×	×
$P6^*$	×	×	*CS*	×

Table 2. Satisfaction of CPAF-principles. *C* stands for complete, *G* for grounded, *P* for preferred (local and global maximization), and *S* for stable. × indicates that none of those semantics satisfy this principle.

Lemma 6. *com^3_{cp} does not satisfy $P5^*$. grd^i_{cp}, with $i \in \{1,2,3,4\}$, does not satisfy any of $P4^*$, $P5^*$, or $P6^*$. Moreover, prf^3_{cp} and $prf\text{-}glb^3_{cp}$ do not satisfy $P6^*$.*

Proof. For complete semantics, consider $A = \{a,b\}$, $R = \{(a,b),(b,a)\}$, $cond$ such that $a \succ_{\emptyset} b$ and $a \succ_{\{b\}} b$, as well as $cond'$ such that $a \succ'_{\emptyset} b$, $a \succ'_{\{b\}} b$, and $b \succ'_{\{a\}} a$. Then $com^3_{cp}(A,R,cond) = \{\{a\}\}$ while $com^3_{cp}(A,R,cond') = \emptyset$.

For grounded semantics, consider $A = \{a,b\}$, $R = \{(a,b),(b,a)\}$, $cond$ such that $a \succ_{\emptyset} b$ and $a \succ_{\{a\}} b$, as well as $cond'$ such that $a \succ'_{\emptyset} b$, $a \succ'_{\{a\}} b$, and $b \succ'_{\{b\}} a$. Then $grd^i_{cp}(A,R,cond) = \{\{a\}\}$ while $grd^i_{cp}(A,R,cond') = \{\{a\},\{b\}\}$.

For preferred semantics, consider $A = \{a,b,c\}$, $R = \{(a,c),(c,a),(b,c),(c,b),(c,c)\}$, $cond$ such that $cond(S) = \emptyset$ for all $S \subseteq A$, and $cond'$ such that $c \succ'_{\{b\}} a$, $c \succ'_{\{a\}} b$, $c \succ'_{\{a,b\}} a$, and $c \succ'_{\{a,b\}} b$. Then $prf^3_{cp}(A,R,cond) = prf\text{-}glb^3_{cp}(A,R,cond) = \{\{a,b\}\}$ while $prf^3_{cp}(A,R,cond') = prf\text{-}glb^3_{cp}(A,R,cond') = \{\{a\},\{b\}\}$. □

The above results constitute an exhaustive investigation of the six CPAF-principles for all semantics considered in this paper. Thus, we can conclude:

Theorem 7. *The satisfaction of CPAF-principles depicted in Table 2 holds.*

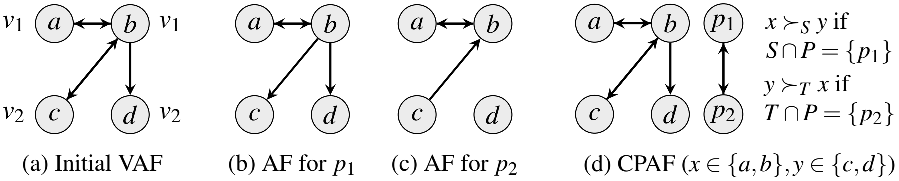
To summarize, complete and stable semantics preserve the satisfaction of PAF-principles in almost all cases. Grounded semantics no longer satisfy any of the principles on CPAFs except $P1^*$ (conflict-freeness) since grounded extensions are not unique on CPAFs, and since there are even CPAFs without a grounded extension (cf. Lemma 6). Unlike on PAFs, complete semantics do not satisfy $P5^*$ (extension growth) under Reduction 3. Furthermore, neither variant of preferred semantics satisfies $P6^*$ (number of extensions) under Reduction 3.

5. Related Formalisms

We now investigate the connection between CPAFs and two related formalisms. First, we show that Value-based Argumentation Frameworks (VAFs) [6,7] can be captured by CPAFs in a straightforward way. Secondly, we consider Extended Argumentation Frameworks (EAFs) [12,13,14] and highlight similarities and differences to CPAFs.

5.1. Capturing Value-Based Argumentation

VAFs, similarly to CPAFs, are capable of dealing with multiple preference relations. But, in contrast to CPAFs, these preferences are not over individual arguments but over values



(a) Initial VAF (b) AF for p_1 (c) AF for p_2 (d) CPAF ($x \in \{a,b\}, y \in \{c,d\}$)

Figure 3. A VAF with two audiences p_1 ($v_1 \succ v_2$) and p_2 ($v_2 \succ v_1$) translated to a CPAF.

associated with arguments. Which values are preferred depends on the audience. A set of arguments may then be accepted in view of one audience, but not in view of another.

More formally, a VAF is a quintuple (A,R,V,val,P) such that (A,R) is an AF, V is a set of values, $val\colon A \to V$ is a mapping from arguments to values, and P is a finite set of audiences. Each audience $p \in P$ is associated with a preference relation $\succ_p$ over values, and $F_P = (A,R,V,val,\succ_p)$ is called an audience-specific VAF (AVAF). The extensions of VAFs are determined for each audience separately. Specifically, an argument x successfully attacks y in F_p iff $(x,y) \in R$ and $val(y) \not\succ_p val(x)$. Conflict-freeness and admissibility are then defined over these successful attacks. In essence, this boils down to using Reduction 1 on F_p, i.e., deleting attacks that contradict the preference ordering.

For example, Figure 3a shows a VAF with two values v_1 and v_2. Let us say there are two audiences in this VAF, p_1 with the preference $v_1 \succ v_2$ and p_2 with $v_2 \succ v_1$. The AFs associated with p_1 and p_2, i.e., the AFs containing only the successful attacks in the AVAFs of p_1 and p_2, are depicted in Figures 3b and 3c.

The reasoning tasks typically associated with VAFs are those of subjective and objective acceptance. Let $F = (A,R,V,val,P)$ be a VAF and $x \in A$. Then x is subjectively accepted in F iff there is $p \in P$ such that x is in a preferred extension of the AVAF $(A,R,V,val,\succ_p)$. Similarly, x is objectively accepted in F iff for all $p \in P$ we have that x is in all preferred extensions of the AVAF $(A,R,V,val,\succ_p)$.

We now provide a translation where an arbitrary VAF $F = (A,R,V,val,P)$ is transformed into a CPAF $Tr(F) = (A',R',cond)$ such that the subjectively and objectively accepted arguments in F correspond to the credulously and skeptically preferred arguments[3] in $Tr(F)$ respectively. Firstly, each audience in the initial VAF becomes an argument in our CPAF, i.e., $A' = A \cup P$. Secondly, the attacks of the VAF are preserved and symmetric attacks are added between all audience-arguments, i.e., $R' = R \cup \{(p,p'),(p',p) \mid p,p' \in P\}$. Lastly, the preferences in our CPAF correspond to the preferences of each audience and are controlled by the newly introduced audience-arguments, i.e., $cond$ is defined such that for $S \subseteq A'$ we have $a \succ_S b$ iff there is $p \in P$ with $S \cap P = \{p\}$ and $val(a) \succ_p val(b)$. See Figure 3 for an example of this transformation.

Observe that the successful attacks in some AVAF $F_p = (A,R,V,val,\succ_p)$ are also attacks in $\mathscr{R}_1^{S\cup\{p\}}(Tr(F))$, where $S \subseteq A$, and vice versa. Thus, the admissible sets in the initial VAF F stand in direct relationship to the admissible sets in our constructed CPAF.

Lemma 8. *Let $F = (A,R,V,val,P)$ be a VAF, $S \subseteq A$, and $p \in P$. Then S is admissible in the AVAF $F_p = (A,R,V,val,\succ_p)$ iff $S \cup \{p\} \in adm^1_{cp}(Tr(F))$.*

[3] As for regular AFs, we say that an argument x is credulously (resp. skeptically) preferred in a CPAF w.r.t. Reduction i iff $x \in S$ for some (resp. for all) $S \in prf^i_{cp}(F)$.

Note that all audience-arguments in $Tr(F)$ attack each other, i.e., an admissible set in $Tr(F)$ contains at most one audience-argument. In fact, each audience-argument defends itself, and thus every preferred extension in $Tr(F)$ must contain exactly one audience-argument $p \in P$ if we appeal to the prf^1_{cp}-semantics. Therefore, the direct correspondence between admissible sets observed in Lemma 8 carries over to preferred extensions.

Theorem 9. *Given a VAF $F = (A, R, V, val, P)$, $x \in A$ is subjectively (resp. objectively) accepted in F iff x is credulously (resp. skeptically) preferred in $Tr(F)$ w.r.t. Reduction* 1.

Our translation highlights the versatility of our formalism. On the one hand, conditional preferences can be tied to dedicated arguments (in this case the audience-arguments). On the other hand, these dedicated arguments themselves may be part of the argumentation process. Note that we used CPAFs with Reduction 1 since preferences in VAFs are usually handled by deleting attacks. However, our approach also allows for the use of other preference-reductions in VAF-settings.

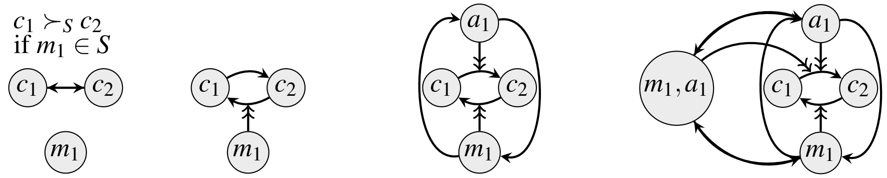
5.2. Relationship to Extended Argumentation Frameworks

EAFs allow arguments to express preferences between other arguments by permitting attacks themselves to be attacked. While EAFs are related to our CPAFs conceptually, we will see that there are crucial differences in how exactly preferences are handled.

Formally, an EAF is a triple (A, R, D) such that (A, R) is an AF, $D \subseteq A \times R$, and if $(a, (b, c)), (a', (c, b)) \in D$ then $(a, a'), (a', a) \in R$. The definition of admissibility in EAFs is quite involved and requires so-called reinstatement sets. Essentially, a set of arguments S is admissible in an EAF if all arguments $x \in S$ are defended from other arguments $y \in A \setminus S$, and if all attacks (z, y) used for defending x are in turn defended from attacks on attacks $(w, (z, y))$ and thus reinstated. It is possible that a chain of such reinstatements is required which is formalized with the aforementioned reinstatement sets. Formally defining these concepts is not necessary for our purposes, but the corresponding definitions can be found in [12]. Observe that the notion of attacks on attacks in EAFs is similar to Reduction 1 in the sense that attacks between arguments can be unsuccessful, but they are never reversed. Therefore, we will compare EAFs to CPAFs with Reduction 1.

Recall our Sherlock Holmes example from the introduction (Example 1) that we modeled as a CPAF (Example 3). Let us first consider a slimmed-down variation without an argument stating that Person 1 has an alibi. We can model this as an EAF with three arguments c_1 (Person 1 is the culprit), c_2 (Person 2 is the culprit), and m_1 (Person 1 has a motive) in which m_1 attacks the attack from c_2 to c_1. The corresponding EAF is depicted in Figure 4b. Compare this to the formalization via a CPAF in Figure 4a. Note that $\{c_1\}$ is admissible in the EAF but $\{c_2\}$ is not since (c_2, c_1) is used to defend against (c_1, c_2) but not reinstated against $(m_1, (c_2, c_1))$. In the CPAF, $\{c_2\}$ is admissible (but not stable).

This simple example highlights a fundamental difference in how preferences are viewed in the two formalisms. In CPAFs, preferences are relevant exactly if the argument that expresses them (e.g. m_1) are part of the set under inspection. In EAFs, preference are relevant even if the argument that expresses them is not accepted. Modgil [12] states that admissibility for EAFs was defined in this way because it was deemed important to satisfy Dung's Fundamental Lemma [15], which says that if S is admissible and x is acceptable w.r.t. S then $S \cup \{x\}$ is admissible. This Fundamental Lemma is not satisfied in our CPAFs. However, in our opinion, this is no drawback but rather a necessary property



(a) Simple CPAF (b) Simple EAF (c) Conflicting preferences (d) Combining preferences

Figure 4. The Sherlock Holmes example modeled via EAFs and a simple CPAF.

of formalisms that can deal with conditional preferences in a flexible way. For example, in Figure 4a it is clear that $\{c_2\}$ should be admissible since, when considering only admissibility, we are not forced to include the unattacked m_1, i.e., we do not have to accept that Person 1 has a motive. The inclusion of unattacked arguments in CPAFs is handled via more restrictive approaches such as stable or preferred semantics, as usual.

Another difference between CPAFs and EAFs becomes clear when considering the entire Sherlock Holmes example. Recall our formalization for CPAFs (cf. Figure 2). In order to express our preference in case Person 1 has an alibi we extend our EAF from Figure 4b by adding an attack from a_1 to the attack (c_1, c_2), as shown in Figure 4c. Note that a_1 and m_1 must attack each other in this EAF by definition since they express conflicting preferences. But this formalization is unsatisfactory since it should be possible for Person 1 to have both a motive and an alibi. The fact that the preference of one argument may change in view of another argument must be modeled indirectly in EAFs. For example, we can introduce an additional argument to express that Person 1 has both a motive and an alibi. This is depicted in Figure 4d. Thus, we can see that CPAFs allow for more flexibility when combining preferences associated with several arguments.

To summarize, CPAFs are designed to express conditional preferences in abstract argumentation, whereas preferences in EAFs are unconditional in the sense that they may always influence the argumentation process, even if the argument associated with the preference is not accepted. Moreover, since our CPAFs can make use of all four preference reductions, they allow for more flexibility in how preferences are handled compared to EAFs, in which unsuccessful attacks are always deleted. However, the two formalisms are similar in that arguments are capable of reasoning about the argumentation process itself, i.e., they constitute a form of metalevel argumentation [17].

6. Conclusion

In this paper, we introduce Conditional Preference-based AFs (CPAFs) which generalize PAFs and allow to flexibly handle conditional preferences in abstract argumentation. We show that the satisfaction of I-maximality can depend on how maximization is dealt with (in case of preferred semantics) and on which preference-reduction is chosen (in case of stable semantics). We conduct a principle-based analysis for CPAFs and show that complete and stable semantics satisfy the same principles as on PAFs in most cases while grounded semantics no longer satisfy the majority of principles. Moreover, we compare CPAFs to related formalisms: on the one hand we show that CPAFs can be used to capture VAFs via a straightforward translation; on the other hand, we demonstrate that CPAFs exhibit significant differences to EAFs in terms of how preferences are handled.

For future work, we plan to introduce an alternative grounded semantics which enforces unique extensions, examine the computational complexity of CPAFs, and consider restricted (e.g. transitive or linear) preference orderings. Moreover, we intend to investigate the relationship between CPAFs and existing approaches in structured argumentation [8] in detail. Related to this last point, it may also be interesting to see whether conditional preferences can be adapted to other formalisms such as bipolar argumentation frameworks [18], in which both attack and support relations are present. As for preference representation, it could be investigated how existing formalisms designed to handle conditional preferences such as CP-nets [19] can be used in the context of CPAFs.

Acknowledgments We thank the anonymous reviewers for their valuable feedback. This work was funded by the Austrian Science Fund (FWF) under grant Y698 and by the Vienna Science and Technology Fund (WWTF) under grant ICT19-065.

References

[1] Amgoud L, Cayrol C. On the Acceptability of Arguments in Preference-based Argumentation. In: Proc. UAI'98. Morgan Kaufmann; 1998. p. 1-7.

[2] Amgoud L, Cayrol C. A Reasoning Model Based on the Production of Acceptable Arguments. Ann Math Artif Intell. 2002;34(1-3):197-215.

[3] Amgoud L, Vesic S. Rich preference-based argumentation frameworks. Int J Approx Reason. 2014;55(2):585-606.

[4] Kaci S, van der Torre LWN, Villata S. Preference in Abstract Argumentation. In: Proc. COMMA'18. vol. 305 of FAIA. IOS Press; 2018. p. 405-12.

[5] Kaci S, van der Torre LWN, Vesic S, Villata S. Preference in Abstract Argumentation. In: Handbook of Formal Argumentation, Volume 2. College Publications; 2021. p. 211-48.

[6] Bench-Capon TJM, Doutre S, Dunne PE. Audiences in argumentation frameworks. Artif Intell. 2007;171(1):42-71.

[7] Atkinson K, Bench-Capon TJM. Value-based Argumentation. IfCoLog Journal of Logic and its Applications. 2021;8(6):1543-88.

[8] Dung PM, Thang PM, Son TC. On Structured Argumentation with Conditional Preferences. In: Proc. AAAI'19. AAAI Press; 2019. p. 2792-800.

[9] Modgil S, Prakken H. Reasoning about Preferences in Structured Extended Argumentation Frameworks. In: Proc. COMMA'10. vol. 216 of FAIA. IOS Press; 2010. p. 347-58.

[10] Modgil S, Prakken H. A general account of argumentation with preferences. Artif Intell. 2013;195:361-97.

[11] Modgil S, Prakken H. Abstract Rule-Based Argumentation. FLAP. 2017;4(8).

[12] Modgil S. Reasoning about Preferences in Argumentation Frameworks. Artif Intell. 2009;173(9-10):901-34.

[13] Dunne PE, Modgil S, Bench-Capon TJM. Computation in Extended Argumentation Frameworks. In: Proc. ECAI'10. vol. 215 of FAIA. IOS Press; 2010. p. 119-24.

[14] Baroni P, Cerutti F, Giacomin M, Guida G. Encompassing Attacks to Attacks in Abstract Argumentation Frameworks. In: Proc. ECSQARU'09. vol. 5590 of LNCS. Springer; 2009. p. 83-94.

[15] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artif Intell. 1995;77(2):321-58.

[16] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: Handbook of Formal Argumentation. College Publications; 2018. p. 159-236.

[17] Modgil S, Bench-Capon TJM. Metalevel argumentation. J Log Comput. 2011;21(6):959-1003.

[18] Amgoud L, Cayrol C, Lagasquie-Schiex M, Livet P. On bipolarity in argumentation frameworks. Int J Intell Syst. 2008;23(10):1062-93.

[19] Boutilier C, Brafman RI, Domshlak C, Hoos HH, Poole D. CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. J Artif Intell Res. 2004;21:135-91.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220145

A Ranking Semantics for Abstract Argumentation Based on Serialisability

Lydia BLÜMEL and Matthias THIMM
Artificial Intelligence Group, University of Hagen, Germany

Abstract. We revisit the foundations of ranking semantics for abstract argumentation frameworks by observing that most existing approaches are incompatible with classical extension-based semantics. In particular, most ranking semantics violate the principle of admissibility, meaning that admissible arguments are not necessarily better ranked than inadmissible arguments. We propose new postulates for capturing said compatibility with classical extension-based semantics and present a new ranking semantics that complies with these postulates. This ranking semantics is based on the recently proposed notion of *serialisability* that allows to rank arguments according to the number of conflicts needed to be solved in order to include that argument in an admissible set.

Keywords. abstract argumentation, ranking semantics, serialisability

1. Introduction

Abstract argumentation frameworks [1] represent argumentative scenarios via directed graphs, where vertices represent arguments and a directed edge from an argument a to an argument b denotes an “attack” from a to b. This simple representation formalism is already powerful enough to analyse and discuss many facets of argumentative reasoning such as argumentation-based dialogue [2], strategic argumentation [3], and dynamics of belief [4], see also [5,6]. Abstract argumentation frameworks are interpreted through formal semantics that assess which arguments can be deemed “acceptable”. The classical approach to formal semantics is by means of *extensions* [1,7], i. e., sets of arguments that form a plausible point of view on the outcome of the argumentation modelled by an abstract argumentation framework. Concrete extension-based semantics specify additional constraints that should be satisfied by their extensions, which capture, e. g., aspects such as conflict-freeness (no argument in an extension should attack another argument in the extension) or admissibility (arguments should be defended by the extension against attacks from outside). Another formal framework for the interpretation of abstract argumentation frameworks is given by ranking-based [8,9,10,11] or graded semantics [12,13,14,15,16,17]. There, *argument strength* is assessed by either *qualitative* (for ranking-based semantics) or *quantitative* (for graded semantics) rankings of arguments. For reasons of simplicity, we will use the term *ranking semantics* to capture both technical frameworks in the following.

Most ranking semantics such as [12,13,8,15] assess argument strength by weighing numbers of attackers and defenders and lengths of paths in the argumentation framework, in some form or the other. As it has already been observed by Bonzon and colleagues [10], there are some fundamental differences in the way ranking semantics assess the acceptability (or better: strength) of arguments, compared to the way this is done by extension-based semantics. As a result, they proposed some hybrid approaches that combine both views by pairing a concrete extension-based semantics with a concrete ranking semantics. In this paper, we pursue another direction, namely the development of a family of ranking semantics that is *compatible* with extension-based semantics in the sense that they *refine* the acceptability assessment of extension-based semantics. As a motivation for this endeavour we start from the general principle of admissibility, a notion that is central to almost all extension-based semantics—with exceptions, of course [18,19]—and demands that acceptable arguments should be defended by acceptable arguments. This core principle is violated by most of the existing ranking semantics in the sense that admissible arguments are not necessarily ranked higher than inadmissible arguments (see Section 3 for details). We consequently propose novel postulates for ranking semantics that capture the intuition behind our aim of developing ranking semantics that are compatible with classical extension-based semantics. We then present and analyse a new ranking semantics that complies with this interpretation. This ranking semantics is based on the notion of *serialisability* [20], which is a principle satisfied by all semantics from [1] and allows the step-wise construction of extensions via iterative selection of non-empty minimal admissible sets—also called *initial sets* [21]—and consideration of the resulting reducts [19]. We will use the minimal number of steps required to include an argument in such a construction as an assessment of the acceptability of an argument. This basically amounts to the number of conflicts between arguments that have to be resolved in order to accept an argument.

To summarise, the contributions of this paper are as follows.

1. We revisit and re-assess the foundations of ranking semantics by introducing and analysing postulates aiming at compatibility with extension-based semantics (Section 3).
2. We discuss a novel ranking semantics based on serialisability (Section 4).

Section 2 presents the background on abstract argumentation and Section 5 concludes this paper. Proofs of technical results can be found in an online appendix.[1]

2. Preliminaries

We present basic background on abstract argumentation and extension-based semantics in Section 2.1 and ranking semantics in Section 2.2.

2.1. Abstract Argumentation

Let $\mathfrak{A}$ denote a universal set of arguments. An *abstract argumentation framework* AF is a tuple $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ where $\mathsf{A} \subseteq \mathfrak{A}$ is a finite set of arguments and R is a relation $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$ [1]. Let $\mathfrak{AF}$ denote the set of all abstract argumentation frameworks. For two arguments $a, b \in \mathsf{A}$ the relation $a\mathsf{R}b$ means that argument a attacks argument b. For $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and

[1] http://mthimm.de/misc/lbmt_rankser_proofs.pdf

$\mathsf{AF}' = (\mathsf{A}', \mathsf{R}')$ we write $\mathsf{AF}' \sqsubseteq \mathsf{AF}$ iff $\mathsf{A}' \subseteq \mathsf{A}$ and $\mathsf{R}' = \mathsf{R} \cap (\mathsf{A}' \times \mathsf{A}')$. For a set $X \subseteq \mathsf{A}$, we denote by $\mathsf{AF}|_X = (X, \mathsf{R} \cap (X \times X))$ the projection of AF on X. For a set $S \subseteq \mathsf{A}$ we define

$$S^+ = \{a \in \mathsf{A} \mid \exists b \in S : b\mathsf{R}a\} \qquad\qquad S^- = \{a \in \mathsf{A} \mid \exists b \in S : a\mathsf{R}b\}$$

If S is a singleton set, we omit brackets for readability, i. e., we write a^- (a^+) instead of $\{a\}^-$ ($\{a\}^+$). For two sets S and S' we write $S\mathsf{R}S'$ iff $S' \cap S^+ \neq \emptyset$.

Two abstract argumentation frameworks $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $\mathsf{AF}' = (\mathsf{A}', \mathsf{R}')$ are *isomorphic*, written $\mathsf{AF} \equiv \mathsf{AF}'$, if there is a bijective function $\gamma : \mathsf{A} \to \mathsf{A}'$ such that $a\mathsf{R}b$ iff $\gamma(a)\mathsf{R}'\gamma(b)$ for all $a, b \in \mathsf{A}$ (γ is then called an *isomorphism*).

A set $S \subseteq \mathsf{A}$ is *conflict-free* if $S \cap S^+ = \emptyset$. S is a *naive* (na) extension if it is maximal wrt. set inclusion among the conflict-free sets of AF. A set S *defends* an argument $b \in \mathsf{A}$ if $b^- \subseteq S^+$. A conflict-free set S is called *admissible* if S defends all $a \in S$. Let $\mathsf{adm}(\mathsf{AF})$ denote the set of admissible sets of AF. Different *extension-based semantics* can be phrased by imposing constraints on admissible sets [7]. In particular, an admissible set E

- is a *complete* (co) extension iff for all $a \in \mathsf{A}$, if E defends a then $a \in E$,
- is a *grounded* (gr) extension iff E is complete and minimal,
- is a *stable* (st) extension iff $E \cup E^+ = \mathsf{A}$,
- is a *preferred* (pr) extension iff E is maximal.

All statements on minimality/maximality are meant to be with respect to set inclusion. For $\sigma \in \{\mathsf{co}, \mathsf{gr}, \mathsf{st}, \mathsf{pr}\}$ let $\sigma(\mathsf{AF})$ denote the set of σ-extensions of AF. The acceptance of an argument a wrt. a given semantics σ distinguishes three levels:

- a is *skeptically accepted* wrt. σ iff $a \in E$ for all $E \in \sigma(\mathsf{AF})$,
- a is *credulously accepted* wrt. σ iff there is $E \in \sigma(\mathsf{AF})$ with $a \in E$,
- a is *rejected* wrt. σ iff $a \notin E$ for all $E \in \sigma(\mathsf{AF})$.

2.2. *Ranking Semantics*

Directly comparing individual arguments with each other yields another class of argumentation semantics. Ranking semantics evaluate the acceptability (or better: strength) of single arguments instead of sets of arguments, their output is a (partial) preorder on the arguments of a given AF.

Definition 1. A ranking semantics is a mapping $\tau : \mathfrak{AF} \to 2^{\mathfrak{A} \times \mathfrak{A}}$ which assigns to each $\mathsf{AF} = (\mathsf{A}, \mathsf{R}) \in \mathfrak{AF}$ a partial preorder $\succeq_{\tau(\mathsf{AF})}$ on A, i. e., $\succeq_{\tau(\mathsf{AF})}$ is transitive and reflexive.

If the AF we refer to is clear from the context, the shorthand $\succeq_\tau$ is used instead. The stronger an argument, the greater its rank among the other arguments, i. e., a is at least as strong as b is represented by $a \succeq_\tau b$. We use the standard shorthands $a \succ_\tau b$ to say that a is strictly stronger than b ($a \succeq_\tau b \wedge b \not\succeq_\tau a$) and $a \simeq_\tau b$ when both arguments are equally strong ($a \succeq_\tau b \wedge b \succeq_\tau a$). An example of a ranking semantics is the categoriser [22,15].

Definition 2. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an AF. The *categoriser semantics* cat assigns to AF the ranking $\succeq_{\mathsf{cat}}$ defined by $a \succeq_{\mathsf{cat}(\mathsf{AF})} b$ iff $cat(\mathsf{AF})(a) \geq cat(\mathsf{AF})(b)$ where

$$cat(\mathsf{AF})(a) = \begin{cases} 1 & \text{if a}^- = \emptyset \\ \frac{1}{1 + \sum\limits_{b \in a^-} cat(\mathsf{AF})(b)} & \text{otherwise} \end{cases}$$

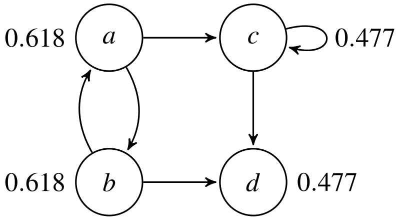


Figure 1. Example with Categoriser values

The above definition yields a system of equations, which can be uniquely solved [15] to obtain the ranks of the individual arguments.

Example 1. The arguments in the AF from Figure 1 are ranked $a \simeq_{\mathsf{cat}} b \succ_{\mathsf{cat}} c \simeq_{\mathsf{cat}} d$ according to the resp. values of the categoriser function (depicted next to the arguments). Since a and b only attack each other, their rank is the same i.e. higher than c and d which are both additionally attacked by c (resulting in the same values for this pair, too).

Following the tradition of the principle-based analysis for extension-based semantics, desirable properties for ranking semantics have been formulated to compare these different approaches. The principles considered in this paper are a selection from [9,17]. Before stating them, we need further notation. A path P of length $l_P = n$ between two arguments a, b is a sequence of arguments $P(a,b) = (a, a_1, ..., a_{n-1}, b)$ with $a_i \mathsf{R} a_{i+1}\ \forall i$ (with $a = a_0, b = a_n$). $cc(\mathsf{AF})$ is the set of all connected components of AF, i. e. all maximal subgraphs $\mathsf{AF}' = (\mathsf{A}', \mathsf{R}')$ such that for every two arguments $a, b \in \mathsf{A}'$ an undirected path $P_u(a,b) = (a = a_0, a_1, ..., a_n = b) \subseteq \mathsf{AF}'$ with $a_i \mathsf{R} a_{i+1}$ or $a_{i+1} \mathsf{R} a_i\ \forall i$ exists.

Definition 3. A ranking semantics τ satisfies the respective principle iff for any $\mathsf{AF} = (\mathsf{A}, \mathsf{R}) \in \mathfrak{AF}$ and any $a, b \in \mathsf{A}$:

Abstraction If for any $\mathsf{AF}' = (\mathsf{A}', \mathsf{R}')$ with $\mathsf{AF} \equiv \mathsf{AF}'$ and for every isomorphism $\gamma\colon \mathsf{A} \to \mathsf{A}'$: $a \succeq_{\tau(\mathsf{AF})} b$ iff $\gamma(a) \succeq_{\tau(\mathsf{AF}')} \gamma(b)$. (*The ranking on arguments should be defined only on the basis of the attacks between them.*)

Independence If for every $\mathsf{AF}' = (\mathsf{A}', \mathsf{R}') \in cc(\mathsf{AF})$ and for all $a, b \in \mathsf{A}'$: $a \succeq_{\tau(\mathsf{AF}')} b$ iff $a \succeq_{\tau(\mathsf{AF})} b$. (*The ranking between two arguments a and b should be independent of any argument that is neither connected to a nor to b.*)

Void precedence If $a^- = \emptyset$ and $b^- \neq \emptyset$ then $a \succ_{\tau(\mathsf{AF})} b$. (*A non-attacked argument should be ranked strictly higher than any attacked argument.*)

Self-contradiction If not $a\mathsf{R}a$ but $b\mathsf{R}b$ then $a \succ_{\tau(\mathsf{AF})} b$. (*A self-attacking argument should be ranked strictly lower than any non self-attacking argument.*)

Cardinality precedence If $|a^-| < |b^-|$ then $a \succ_{\tau(\mathsf{AF})} b$. (*The greater the number of direct attackers for an argument, the weaker the level of acceptability of this argument.*)

Quality precedence If there is $c \in b^-$ such that for all $d \in a^-$, $c \succeq_{\tau(\mathsf{AF})} d$ but not $d \succeq_{\tau(\mathsf{AF})} c$, then $a \succ_{\tau(\mathsf{AF})} b$. (*The greater the acceptability of one direct attacker for an argument, the weaker the level of acceptability of this argument.*)

Counter-Transitivity If some injective $f : a^- \to b^-$ exists such that $f(x) \succeq_\tau x\ \forall x \in a^-$ then $a \succeq_{\tau(\mathsf{AF})} b$. (*If the direct attackers of b are at least as numerous and acceptable as those of a, then a is at least as acceptable as b.*)

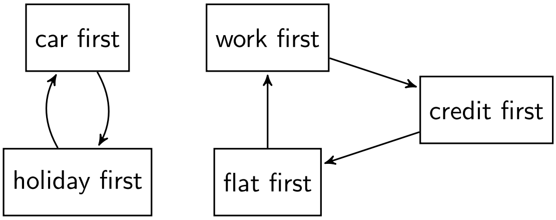


Figure 2. Two simple choice problems

Strict Counter-Transitivity If some injective $f : a^- \to b^-$ exists such that $f(x) \succeq_\tau x\ \forall x \in a^-$ and either $|a^-| < |b^-|$ or there exists some $x \in a^-$ with $f(x) \succ_\tau (x)$ then $a \succ_\tau b$. (*If the direct attackers of b are strictly more numerous or acceptable than those of a, then a is strictly more acceptable than b.*)

Defense precedence If $|a^-| = |b^-|$ and $(a^-)^- \neq \emptyset$ but $(b^-)^- = \emptyset$ then $a \succ_{\tau(\mathsf{AF})} b$. (*For two arguments with the same number of direct attackers, a defended argument is ranked higher than a non-defended argument.*)

Distributed Defense precedence If $|a^-| = |b^-|$ and $|(a^-)^-| = |(b^-)^-|$, and if the defense of a is simple—every direct defender of a directly attacks exactly one direct attacker of a—and distributed—every direct attacker of a is attacked by at most one argument—and the defense of b is simple but not distributed, then $a \succ_{\tau(\mathsf{AF})} b$. (*The best defense is when each defender attacks a distinct attacker (*distributed defense*).*)

Total If $a \succeq_{\tau(\mathsf{AF})} b$ or $b \succeq_{\tau(\mathsf{AF})} a$. (*All pairs of arguments can be compared.*)

Non-attacked Equivalence If $a^- = \emptyset$ and $b^- = \emptyset$ then $a \succeq_{\tau(\mathsf{AF})} b$ and $b \succeq_{\tau(\mathsf{AF})} a$. (*All the non-attacked arguments have the same rank.*)

Attack vs Full Defense If AF is acyclic and every path $P(u,a)$ in AF from an unattacked u to a has $l_p = 0\ mod\ 2$ and there exists $u \in b^-$ unattacked then $a \succ_\tau b$. (*An argument without any unattacked indirect attackers should be ranked higher than an argument only attacked by one unattacked argument.*)

3. Rankings, Admissibility, and Reinstatement

Let us start with the motivation for our work in form of a practical example.

Example 2. The argumentation framework depicted in Figure 2 illustrates two everyday choice problems. While the average employee might have to choose between a new car or an overseas holiday, those who have hit rock bottom often find themselves in unsolvable dilemmas. Suppose a homeless person with no money decided to change her life and look for work. Any company would need her address to handle taxes. So she first has to find herself a place to live. But she cannot pay the deposit for renting a flat. So she needs a credit. However any credit institute would require her to have a job in the first place.

When we apply the categoriser semantics to the argumentation framework from the example, all five arguments receive the same value of approximately 0.618. But ranking the impossible choices of the second case just as high as the two options of the first, which can actually materialize, seems inaccurate. The same issue is present in Example 1, where the self-attacker c and the credulously acceptable d end up having the same value. In practice we would have to execute caution when interpreting this ranking. For example, one known strategy of conspiracy theories is to convince with the sheer amount of arguments in favor of their hypothesis. Under existing ranking semantics, a classically acceptable argument with lots of attackers could end up with a lower rank than, e. g., one of its attackers which is in turn attacked by an unattacked argument like a fact (objective measurement etc.). This potentially leads to a nonsensical argument being ranked higher than a scientifically supported position. We therefore suggest the acceptability of an argument should be represented in its rank somehow. Strictly speaking, if an argument has no chance of being classically accepted, we should expect it to rank lower than any argument which is actually acceptable. Since existing ranking semantics do not conform to this, a need to investigate new options for ranking semantics emerges. That is not to say acceptance has been completely ignored in ranking semantics so far. A few existing principles already incorporate some aspects of classic defense, e. g., *defense precedence* demands that a ranking-semantics prefers an argument with defenders over one with only attackers provided they have the same number of attackers. The principle we introduce here is a more general approach to integrate extension-based acceptability into ranking semantics. In this paper we limit our investigations to its implications under classic admissible semantics, though.

Definition 4. Let σ be an extension-based semantics. A ranking semantics τ satisfies *σ-compatibility* iff for any $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and any $a, b \in \mathsf{A}$, if a is credulously accepted under σ and b is rejected then $a \succ_{\tau(AF)} b$.

σ-compatibility ensures no non-acceptable argument can rank as high or higher than any of the acceptable arguments under σ. A special case of σ-compatibility is *na-compatibility* which results in self-attackers ranking strictly lower than the rest. This is actually equivalent to the existing principle of *self-contradiction*.

Proposition 1. *A ranking semantics τ satisfies self-contradiction iff it satisfies* na-compatibility.

Let us turn our attention to *adm-compatibility*. The classic admissible, complete and preferred semantics credulously accept the same arguments, so *adm-compatibility* covers all three. *adm-compatibility* is incompatible with a number of known principles for ranking semantics, notably with *strict counter-transitivity*. This also generalizes some observations from [11].

Proposition 2. *Let τ be a ranking semantics satisfying* adm-compatibility. *Then τ does not satisfy any of the following four principles:* strict counter-transitivity, counter-transitivity, cardinality precedence*, and* quality precedence.

We briefly demonstrate the contradiction with *strict counter-transitivity* using Example 2. The credulously accepted arguments under classic admissibility are car first and holiday first. Suppose some ranking semantics τ satisfies *adm-compatibility*, then both car first $\succ_{\tau(AF)}$ credit first and holiday first $\succ_{\tau(AF)}$ flat first hold. But under *strict counter-transitivity* car first $\succ_{\tau(AF)}$ credit first implies flat first $\succ_{\tau(AF)}$ holiday first, so τ can at most satisfy one of the two principles. Many existing ranking semantics such as the categoriser semantics from above, but also the burden- and discussion-based semantics

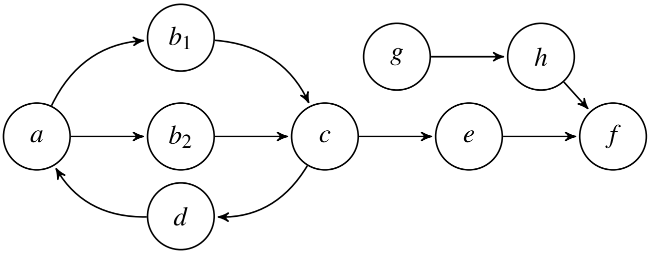


Figure 3. Weak and strong σ-support

[8] and the social abstract argumentation semantics [14] satisfy *strict counter-transitivity* [9]. Therefore none of them satisfies *adm-compatibility*.

For our next new properties we investigate the innate relational structure of extensions or, more precisely, the relations between acceptable arguments under a given extension-based semantics. A longstanding defense-related principle for extension-based semantics is *reinstatement*, the inclusion of defended arguments in an extension [23]. Now, an argument depending on other arguments to become defended should not be ranked higher than its defenders and strictly lower, if its defenders are acceptable independently from it. A way to express this is as follows.

Definition 5. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, $a, b \in \mathsf{A}$, and σ some extension-based semantics.

- *a weakly σ-supports b* if b is credulously accepted wrt. σ and for all $E \in \sigma(\mathsf{AF})$, if $b \in E$ then $a \in E$.
- *a strongly σ-supports b* if b is credulously accepted wrt. σ and for all $E \in \sigma(\mathsf{AF})$, if $b \in E$ then there is $E' \in \sigma(\mathsf{AF})$ with $E' \subseteq E$, $a \in E'$, and $b \notin E'$.

Informally, an argument a weakly σ-supports an argument b, if a is a part of any σ-extension containing b which intuitively amounts to a being an unavoidable side-effect of accepting b. Moreover, a strongly σ-supports b, if the presence of b in an extension (which necessarily also includes a) implies the existence of a smaller extension with a but without b. In that case a becomes a prerequisite for accepting b while b can be said to be irrelevant for accepting a. It is clear that strong σ-support implies weak σ-support.

Example 3. Let $\sigma = \mathsf{adm}$, then in the AF depicted in Figure 3 the arguments a and c weakly σ-support each other, since they take care of each others attackers in the 4-cycle. Neither of them can be accepted on its own, so a does not strongly σ-support c nor vice versa. Both of them strongly σ-support f because c is the only attacker of e and cannot be accepted without a. Note that although a and c strongly σ-support f, this does not imply they are sufficient for accepting f, g strongly σ-supports f as well. Take also note of d weakly σ-supporting b_1 but not the other way around. Since no admissible subset of b_1, d containing only d exists, d does not strongly σ-support b_1, though.

Using these two notions of σ-support, we can define the according principles for ranking semantics as follows.

Definition 6. Let τ be a ranking semantics.

- τ satisfies *weak σ-support* iff for every $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, $a, b \in \mathsf{A}$, if a weakly σ-supports b then $a \succeq_{\tau(AF)} b$.

- τ satisfies *strong σ-support* iff for every $\mathsf{AF} = (\mathsf{A},\mathsf{R})$, $a,b \in \mathsf{A}$, if a strongly σ-supports b then $a \succ_{\tau(AF)} b$.

The two principles are independent from each other, unlike the argument relations they are based upon.

Proposition 3. Strong σ-support *does not imply* weak σ-support *and* weak σ-support *does not imply* strong σ-support.

Reflecting on these two principles for ranking semantics leads again to some interesting observations. Weak σ-support firmly links the rank of an argument with the situations (extensions) in which it is accepted. If two arguments are always accepted together they are of equal rank.

Remark 1. Let τ be a ranking semantics satisfying *weak σ-support*, $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ an AF and $a,b \in \mathsf{A}$. If a weakly σ-supports b and b weakly σ-supports a then $a \simeq_{\tau(AF)} b$.

In case of a single-status semantics (or semantics providing a single extension for some framework) the above observation results in all accepted arguments sharing the same rank.

Corollary 1. *Let τ be a ranking semantics satisfying* weak σ-support, $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ *an AF. If* $|\sigma(\mathsf{AF})| = 1$ *then for all credulously accepted* $a,b \in \mathsf{A}$, $a \simeq_{\tau(AF)} b$.

For the same reason *weak σ-support* enforces the same rank on all skeptically accepted arguments wrt. any semantics.

Corollary 2. *Let τ be a ranking semantics satisfying* weak σ-support, $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ *an AF. For all skeptically accepted* $a,b \in \mathsf{A}$, $a \simeq_{\tau(AF)} b$.

While *weak σ-support* only blocks arguments from ranking better than those they depend on, *strong σ-support* discriminates between arguments in order to express one-sided dependencies as asymmetric rank differences. It has a kind of chain effect, an argument is not only ranked lower than the arguments it depends on but also lower than the arguments those arguments depend on in turn. When applied to classic admissibility, this property leads to a preference of short defense routes. Let us now investigate the relationships of *weak/strong adm-support*.

Proposition 4. *Let τ be a ranking semantics satisfying* strong adm-support. *Then τ does not satisfy any of the following four principles:* strict counter-transitivity, counter-transitivity, cardinality precedence *and* quality precedence.

Proposition 5. *Let τ be a ranking semantics satisfying* weak adm-support. *Then τ does not satisfy* strict counter-transitivity *or* cardinality precedence.

As expected, the two forms of *adm-support* do not go well with attacker-focused properties like *strict counter-transitivity* for which we already demonstrated their contradiction with *adm-compatibility*. Since, e. g., the categoriser semantics, burden- and discussion-based semantics, and social abstract argumentation semantics all satisfy *strict counter-transitivity* [9], none of them satisfies *weak/strong adm-support*.

4. A Ranking Semantics Based on Serialisability

In this section we will introduce a ranking semantics capable of expressing both the acceptability differences and the dependencies between arguments under the classic admissible semantics formalized in the previous section as *adm-compatibility* and *weak/strong adm-support*. In order to do this, we take the admissible extensions apart and analyze them for the relevant dependencies. The smallest units within an admissible set, which still maintain admissibility are the so-called *initial sets* introduced in [21].

Definition 7. For $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, a set $S \subseteq \mathsf{A}$ with $S \neq \emptyset$ is called an *initial set* if S is admissible and there is no admissible $S' \subsetneq S$ with $S' \neq \emptyset$. Let $\mathsf{IS}(\mathsf{AF})$ denote the set of initial sets of AF.

Example 4. The initial sets of the AF depicted in Figure 3 are $\{g\}, \{a,c\}, \{b_1,d\}$ and $\{b_2,d\}$.

Note that not all credulously accepted arguments wrt. admissibility are members of initial sets, e. g., f in Figure 3 is part of the admissible set $\{a,c,f,g\}$ but not contained in any initial set. Arguments like f are exactly those which depend on others for their defense while not being necessary for the defense of their defenders. In [20], a construction method for admissible sets is presented, which implements a form of step-by-step addition for including such arguments. This approach relies on the reduct [19] of an argument set in an argumentation framework.

Definition 8. For $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $S \subseteq \mathsf{A}$, the *reduct* of S wrt. AF is $\mathsf{AF}^S = \mathsf{AF}|_{\mathsf{A} \setminus (S \cup S^+)}$.

Using the reduct, the central idea of [20] can be formalised with the following notion of a serialisation sequence.

Definition 9. A *serialisation sequence* for $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ is a sequence $\mathscr{S} = (S_1, ..., S_n)$ with $S_1 \in \mathsf{IS}(\mathsf{AF})$ and for each $2 \leq i \leq n$ we have $S_i \in \mathsf{IS}(\mathsf{AF}^{S_1 \cup \ldots \cup S_{i-1}})$.

It has been shown that admissible sets can be characterized by serialisation [20]:

Proposition 6. *Let* $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ *be an AF and* $E \subseteq \mathsf{A}$. $E \in \mathit{adm}(\mathsf{AF})$ *if and only if there is a serialisation sequence* $(S_1, ..., S_n)$ *with* $E = S_1 \cup \ldots \cup S_n$.

Let us demonstrate this for some of the admissible sets of our previous example.

Example 5. Consider the admissible sets $S_1 = \{b_1, d\}$, $S_2 = \{b_1, b_2, d, g\}$, and $S_3 = \{a, c, g, f\}$ and corresponding serialisation sequences:

$$\mathscr{S}_1 = (\{b_1, d\}) \qquad \text{(for } S_1\text{)}$$

$$\mathscr{S}_2 = (\{b_1, d\}, \{b_2\}, \{g\}) \qquad \text{(for } S_2\text{)}$$

$$\mathscr{S}_3 = (\{a, c\}, \{g\}, \{f\}) \qquad \text{(for } S_3\text{)}$$

Serialisation sequences are not necessarily unique, but certain arguments can only be selected after they appear in some initial set. For example, $\{f\}$ only becomes an initial set after g (and $\{a,c\}$) are already part of the sequence. This dependency between sets in a serialisation sequence is similar to the *strong adm-support* introduced in Section 3.

Now that we have a tool for representing the structure of admissible sets, we can use it for defining a new argument ranking that is based on the length of shortest serialisation sequences.

Definition 10. For $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $a \in \mathsf{A}$ define the *serialisation index* $\mathsf{ser}_{\mathsf{AF}}(a)$ via

$$\mathsf{ser}_{\mathsf{AF}}(a) = \min\{n \mid (S_1, \ldots, S_n) \text{ is a serialisation sequence and } a \in S_n\}$$

with $\min \emptyset = \infty$.

Intuitively, the value of $\mathsf{ser}_{\mathsf{AF}}(a)$ represents the minimal number of conflicts, which have to be solved before an argument a can be accepted. In this context, $\mathsf{ser}_{\mathsf{AF}}(a) = 1$ means a can solve all relevant conflicts by "itself" or—to be correct—by being a member of an initial set itself. The serialisation-index $\mathsf{ser}_{\mathsf{AF}}(a) = \infty$ for non-acceptable arguments can be read as no serialisation sequence of any length will be sufficient for this argument. From the choice of a trivial value for all non-acceptable arguments, it already becomes clear that our ranking will only represent differences between acceptable arguments. To foster our understanding of these values let us compute the serialisation indices for our running example.

Example 6. For the arguments of the AF from Figure 3 we get $\mathsf{ser}_{\mathsf{AF}}(x) = 1$ for $x \in \{a, c, b_1, b_2, d, g\}$ a member of an initial set, $\mathsf{ser}_{\mathsf{AF}}(e) = 2$, since the two smallest admissible sets containing e are $\{b_1, d, e\}$ and $\{b_2, d, e\}$ which both can be serialised in $k = 2$ steps, $\mathsf{ser}_{\mathsf{AF}}(f) = 3$ because two initial sets, $\{g\}$ and $\{a, c\}$ are needed for the defense of f and have to be included first before $\{f\}$ becomes an initial set in $\mathsf{AF}^{\{a,c\} \cup \{g\}}$ and $\mathsf{ser}_{\mathsf{AF}}(h) = \infty$ for the non-acceptable argument h.

The ranking semantics naturally arising from the serialisation index is as follows.

Definition 11. For $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $a, b \in \mathsf{A}$, we say that a is *at least as preferred as* b (wrt. serialisability), written $a \succeq_{\mathsf{ser}} b$ iff $\mathsf{ser}_{\mathsf{AF}}(a) \leq \mathsf{ser}_{\mathsf{AF}}(b)$.

The lower the serialisation index, the higher the rank of an argument with the members of initial sets all being ranked equally at the top. Applying this ranking semantics to our running example yields $a \simeq_{\mathsf{ser}} b_1 \simeq_{\mathsf{ser}} b_2 \simeq_{\mathsf{ser}} c \simeq_{\mathsf{ser}} d \simeq_{\mathsf{ser}} g \succ_{\mathsf{ser}} e \succ_{\mathsf{ser}} f \succ_{\mathsf{ser}} h$. We will now prove that this ranking semantics indeed has the desired properties defined in Section 3 and begin by demonstrating that $\succeq_{\mathsf{ser}}$ produces the intended results for our motivating example.

Example 7. The options for the decision problems represented in Figure 2 are assigned the serialisation indices $\mathsf{ser}_{\mathsf{AF}}(\mathsf{holiday\ first}) = \mathsf{ser}_{\mathsf{AF}}(\mathsf{car\ first}) = 1$ and $\mathsf{ser}_{\mathsf{AF}}(\mathsf{flat\ first}) = \mathsf{ser}_{\mathsf{AF}}(\mathsf{work\ first}) = \mathsf{ser}_{\mathsf{AF}}(\mathsf{credit\ first}) = \infty$ respectively, resulting in a higher ranking for the viable options of the average employee.

Indeed, the serialisation ranking satisfies *adm-compatbility* per definition, since the serialisation index for non-acceptable arguments of ∞ cannot be reached by acceptable arguments. The conformity to *weak* and *strong adm-support* is not that trivial, but can also be shown.

Theorem 1. $\succeq_{\mathsf{ser}}$ *satisfies* adm-compatibility *and both* strong *and* weak adm-support.

The similarities of our new ranking semantics to classical extension-based semantics do not stop with the above result. Another important property of admissibility semantics is *directionality*, i. e., the admissible sets of an unattacked subset of an AF are also admissible in the AF as a whole [23]. The intuition behind this principle is that an argument a which has no directed path to an argument b should not have any impact on the acceptability of b. This idea makes sense for ranking semantics as well and an according principle for graded semantics was formulated in [24]. Here we generalize this directionality principle for ranking semantics.

Definition 12. A ranking-based semantics τ satisfies *directionality* iff for any $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ and any $a,b,x,y \in \mathsf{A}$ such that $(a,b) \notin \mathsf{R}$ and no directed path from b to neither x nor y exists, then $x \succ_{\tau(\mathsf{AF})} y$ if and only if $x \succ_{\tau((\mathsf{A},\mathsf{R}\cup\{(a,b)\}))} y$.

Proposition 7. $\succeq_{ser}$ *satisfies* directionality.

Regarding the principles from Definition 3, $\succeq_{\mathsf{ser}}$ satisfies the general ones such as *abstraction* and *independence*. Most of the other principles are not satisfied, in particular because of the incompatibilities we already showed in Propositions 2, 4, and 5. Further principles are not satisfied because they demand rank differences under certain structural conditions, like *distributed defense precedence*. Since all non-acceptable arguments have the same rank under $\succeq_{\mathsf{ser}}$, those principles are violated if their conditions can apply to pairs of non-acceptable arguments. For example, defense precedence is only uphold in case the stronger argument is acceptable. The following proposition summarizes our findings.

Proposition 8. $\succeq_{ser}$ *satisfies* abstraction*,* independence*,* totality*,* non-attacked equivalence*, and* attack vs full defense*. All other principles from Def. 3 are not satisfied.*

5. Summary and Conclusion

We revisited the foundation of ranking semantics for abstract argumentation and proposed a new interpretation of ranking semantics as *refinements* of classical extension-based semantics. For that aim, we presented the new postulates *σ-compatibility* as well as *weak* and *strong σ-support* and showed that these are generally incompatible with existing postulates for ranking semantics. We proposed a new ranking semantics based on the concept of serialisibility and showed that this new semantics behaves well wrt. these postulates.

Our contributions should be regarded as an additional aspect of interpreting ranking semantics and not as disregarding previous approaches. The central aspect of existing ranking semantics is that they aim at assessing *strength* of arguments, which is—as we have seen in this paper — not necessarily the same as acceptability. Here, we aimed at comparing acceptability (wrt. admissibility) of arguments. An interesting avenue for future work is also to investigate more general foundations for acceptability such as *weak admissibility* [19,23] or to exploit different notions of *defense* [25] for our formalisation of weak and strong σ-support.

Acknowledgements The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (grant 375588274).

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995;77(2):321-58.

[2] Black E, Maudet N, Parsons S. Argumentation-based Dialogue. In: Handbook of Formal Argumentation. vol. 2. College Publications; 2021. p. 511-76.

[3] Governatori G, Maher MJ, Olivieri F. Strategic Argumentation. In: Handbook of Formal Argumentation. vol. 2. College Publications; 2021. p. 577-662.

[4] Baumann R, Doutre S, Mailly JG, Wallner JP. Enforcement in Formal Argumentation. In: Handbook of Formal Argumentation. vol. 2. College Publications; 2021. p. 445-510.

[5] Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018.

[6] Gabbay D, Giacomin M, Simari GR, Thimm M, editors. Handbook of Formal Argumentation. vol. 2. College Publications; 2021.

[7] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: Handbook of Formal Argumentation. College Publications; 2018. p. 159-236.

[8] Amgoud L, Ben-Naim J. Ranking-based Semantics for Argumentation Frameworks. In: Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM'13); 2013. p. 134-47.

[9] Bonzon E, Delobelle J, Konieczny S, Maudet N. A Comparative Study of Ranking-based Semantics for Abstract Argumentation. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16); 2016. p. 914-20.

[10] Bonzon E, Delobelle J, Konieczny S, Maudet N. Combining extension-based semantics and ranking-based semantics for abstract argumentation. In: Sixteenth International Conference on Principles of Knowledge Representation and Reasoning; 2018. p. 118-27.

[11] Amgoud L, Ben-Naim J. Axiomatic foundations of acceptability semantics. In: International Conference on Principles of Knowledge Representation and Reasoning (KR 2016); 2016. p. 2-11.

[12] Cayrol C, Lagasquie-Schiex MC. Graduality in Argumentation. Journal of Artificial Intelligence Research. 2005;23:245-97.

[13] Matt PA, Toni F. A Game-Theoretic Perspective on the Notion of Argument Strength in Abstract Argumentation. In: Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA'2008); 2008. p. 285-97.

[14] Leite J, Martins J. Social Abstract Argumentation. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11); 2011. .

[15] Pu F, Luo J, Zhang Y, Luo G. Argument Ranking with Categoriser Function. In: Proceedings of the 7th International Conference on Knowledge Science, Engineering and Management (KSEM'14). Springer; 2014. p. 290-301.

[16] Grossi D, Modgil S. On the graded acceptability of arguments in abstract and instantiated argumentation. Artificial Intelligence. 2019;275:138-73.

[17] Thimm M, Cerutti F, Rienstra T. Probabilistic Graded Semantics. In: Proceedings of the Seventh International Conference on Computational Models of Argumentation (COMMA'18); 2018. p. 369-80.

[18] Baroni P, Giacomin M, Guida G. SCC-recursiveness: a general schema for argumentation semantics. Artificial Intelligence. 2005;168(1–2):162-210.

[19] Baumann R, Brewka G, Ulbricht M. Revisiting the Foundations of Abstract Argumentation - Semantics Based on Weak Admissibility and Weak Defense. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20); 2020. p. 2742-9.

[20] Thimm M. Revisiting initial sets in abstract argumentation. Argument & Computation. 2022 July.

[21] Xu Y, Cayrol C. Initial Sets in Abstract Argumentation Frameworks. In: Proceedings of the 1st Chinese Conference on Logic and Argumentation (CLAR'16); 2016. p. 1-20.

[22] Besnard P, Hunter A. A logic-based theory of deductive arguments. Artificial Intelligence. 2001;128(1-2):203-35.

[23] Baroni P, Giacomin M. On principle-based evaluation of extension-based argumentation semantics. Artificial Intelligence. 2007;171(10-15):675-700.

[24] Amgoud L, Ben-Naim J, Doder D, Vesic S. Acceptability Semantics for Weighted Argumentation Frameworks. In: Twenty-Sixth International Joint Conference on Artificial Intelligence; 2017. p. 56-62.

[25] Blümel L, Ulbricht M. Defining Defense and Defeat in Abstract Argumentation From Scratch - A Generalizing Approach. In: Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR'22); 2022. .

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220146

NEXAS: A Visual Tool for Navigating and Exploring Argumentation Solution Spaces

Raimund DACHSELT [a,d,e], Sarah Alice GAGGL [b], Markus KRÖTZSCH [c], Julián MÉNDEZ [a], Dominik RUSOVAC [b] and Mei YANG [a]

[a] *Interactive Media Lab Dresden, Technische Universität Dresden*
[b] *Logic Programming and Argumentation Group, Technische Universität Dresden*
[c] *Knowledge-Based Systems Group, Technische Universität Dresden*
[d] *Centre for Tactile Internet (CeTI), Technische Universität Dresden*
[e] *Cluster of Excellence Physics of Life (PoL), Technische Universität Dresden*

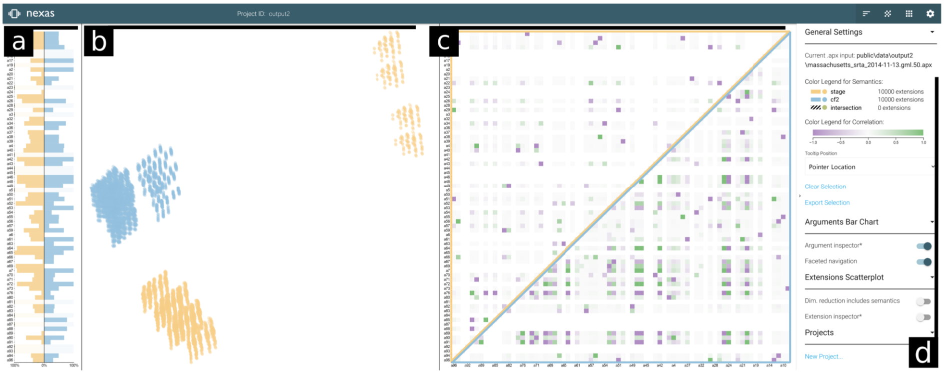
ORCiD ID: Raimund Dachselt https://orcid.org/0000-0002-2176-876X, Sarah Alice Gaggl https://orcid.org/0000-0003-2425-6089, Markus Krötzsch https://orcid.org/0000-0002-9172-2601, Julián Méndez https://orcid.org/0000-0003-1029-7656, Dominik Rusovac https://orcid.org/0000-0002-3172-5827

Abstract. Recent developments on solvers for abstract argumentation frameworks (AFs) made them capable to compute extensions for many semantics efficiently. However, for many input instances these solution spaces can become very large and incomprehensible. So far, for the further exploration and investigation of the AF solution space the user needs to use post-processing methods or handcrafted tools. To compare and explore the solution spaces of two selected semantics, we propose an approach that visually supports the user, via a combination of dimensionality reduction of argumentation extensions and a projection of extensions to sets of accepted or rejected arguments. We introduce the novel web-based visualization tool NEXAS that allows for an interactive exploration of the solution space together with a statistical analysis of the acceptance of individual arguments for the selected semantics, as well as provides an interactive correlation matrix for the acceptance of arguments. We validate the tool with a walk-through along three use cases.

Keywords. abstract argumentation, visualization, solution space exploration

1. Introduction

Abstract argumentation is a very active research field within argumentation theory. Besides the theoretical analysis and formal approaches [1] the development of efficient solvers has gained much attention [2], as also witnessed by the International Competition on Computational Models of Argumentation (ICCMA) [3]. This led to a number of solvers that can enumerate all extensions of most of the prominent argumentation semantics. However, the solution space for many semantics can become very large, in particular if many cycles of even length are contained in the given argumentation framework (AF).



General Settings

Current .apx input: public\data\output2
\massachusetts_srta_2014-11-13.gml.50.apx

Color Legend for Semantics:

●	stage	10000 extensions
●	cf2	10000 extensions
▨ ●	intersection	0 extensions

Color Legend for Correlation:

Tooltip Position: Pointer Location

Clear Selection

Export Selection

Arguments Bar Chart

Argument inspector* ●

Faceted navigation ●

Extensions Scatterplot

Dim. reduction includes semantics ○

Extension inspector* ○

Projects

New Project...

Figure 1. NEXAS interface showing a) the *Argument View*: bar chart with the relative frequencies of the arguments by semantics, b) the *Extension View*: scatterplot with the dimensionality reduced solution space, c) the *Correlation View*: correlation matrix of accepted arguments wrt. two semantics separated by the diagonal, and d) a sidebar with the color legend, settings and inspection options.

While theoretical results on AFs already provide a rather clear picture of the relations between the argumentation semantics, for example in terms of subset-relations or the existence of solutions for specific classes of AFs, there is no practical support to explore the solution space of AFs. Standard AF solvers are mainly targeted to i) compute extensions or decide for credulous and skeptical acceptance of arguments efficiently, or ii) to support the construction of arguments and AFs visually [4]. Up to now, the only combination of computational argumentation and visualization is by tools that use a solver in the backend for the computation of solutions or acceptance of arguments, and the visualization of the AF as a directed graph. The outcome of the computation is then visualized by highlighting arguments that are credulously or skeptically accepted, or one extension after another is highlighted in the graph for a selected semantics[1] [5]. With the current tool support the user does not get the full picture on the acceptance of individual arguments, or sets of arguments. When comparing two semantics, handcrafted tools are still needed to post-process all extensions and find out where they coincide or differ. Getting an overview on the whole solution space for specific semantics, to be able to zoom-in or -out into particular sub-spaces that contain similar extensions, is not at all supported by current tools.

To overcome these limitations, we embarked on an interdisciplinary endeavour proposing to utilize advanced visualization to interactively explore the solution space of AFs. As a result, we introduce the NEXAS tool for Navigating and Exploring Argumentation Solutions. In this work we combine the well known Answer Set Programming (ASP) based AF solver `aspartix` [6], and the recently developed approach for answer set navigation [7] to, on one side enumerate extensions, and on the other side navigate towards "interesting" sub-spaces of arguments.

Our main contributions are: (1) An approach that seamlessly integrates visualization techniques and systems for enumerating extensions for abstract argumentation frameworks. (2) Our novel web-based tool NEXAS, an interactive solution that visually realizes the aforementioned approach [7] through three major coordinated views fostering navigation, exploration and understanding of the AF (See Figure 1). (3) Walk-through

[1] Web interface of `ConArg` (`https://conarg.dmi.unipg.it/web_interface.php`)

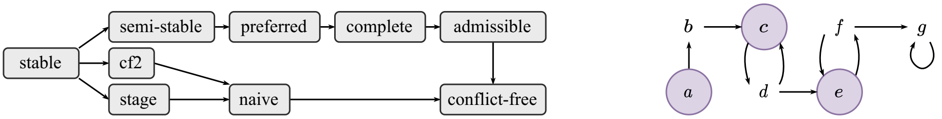


Figure 2. a) (left) An arrow from semantics σ to semantics τ encodes that each σ-extension is also a τ-extension. **b)** (right) Example AF with highlighted preferred extension $\{a, c, e\}$.

validation of the tool NEXAS on the basis of three use cases. A demo of the tool, video walk-through and test data can be found at https://imld.de/nexas.

2. Background and Related Work

2.1. Abstract Argumentation Frameworks

We recall the basic definitions for abstract argumentation frameworks. For more details we refer to the standard literature [8,9].

Definition 1 *An abstract argumentation framework (AF) is a pair denoted as $F = (A, R)$ where A is a finite set of arguments and $R \subseteq A \times A$ is the attack relation.*

Given a pair $(a, b) \in R$, we say a attacks b. An argument $a \in A$ is *defended* by a set $S \subseteq A$ if, for each $b \in A$ such that $(b, a) \in R$, there exists a $c \in S$ such that $(c, b) \in R$. Additionally, the *range* of S (w.r.t. R) is defined as $S_R^+ = S \cup \{x \mid \exists y \in S \text{ such that } (y, x) \in R\}$.

Definition 2 *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is* conflict-free (in F)*, if there are no $a, b \in S$, such that $(a, b) \in R$. $cf(F)$ denotes the collection of* conflict-free *sets of F. For a conflict-free set $S \in cf(F)$, it holds that*

- *S is an* admissible *set of F, i.e., $S \in adm(F)$, if each $a \in S$ is defended by S;*
- *S is a* stable *extension of F, i.e., $S \in stb(F)$, if $S_R^+ = A$;*
- *S is a* complete *extension of F, i.e., $S \in compl(F)$, if $S = \{s \in A \mid S \, defends \, s\}$;*
- *S is a* preferred *extension of F, i.e., $S \in prf(F)$, if $S \in compl(F)$ and there is no $T \in compl(F)$ with $T \supset S$;*
- *S is a* stage *extension of F, i.e., $S \in stg(F)$, if there is no $T \in cf(F)$, such that $S_R^+ \subset T_R^+$;*
- *S is a* maximal conflict-free *or* naive *set of F, i.e., $S \in naive(F)$, if $S \in cf(F)$ and for each $T \in cf(F)$, $S \not\subset T$;*
- *S is a* semi-stable *extension of F, i.e., $S \in semis(F)$, if and only if $S \in compl(F)$ and there is no $T \in compl(F)$, such that $S_R^+ \subset T_R^+$.*

Our tool also supports the SCC-recursive semantics *cf2*. For more details we refer to the following article [9]. Figure 2a) shows the relation between the introduced semantics in terms of subset-relations. Figure 2b) shows an example AF represented as a directed graph where the preferred extension $\{a, c, e\}$ is highlighted.

We denote the set of arguments of an AF F that are *credulously accepted* under semantics σ by $cred_\sigma(F) := \bigcup \sigma(F)$, and the set of arguments that are *skeptically accepted* under semantics σ by $skep_\sigma(F) := \bigcap \sigma(F)$.

2.2. Related Work

Visualization in Abstract Argumentation Frameworks. Although there are many argumentation systems [2], approaches that concentrate on the visualization of solution spaces for AFs have not yet been investigated in detail. The only work we are aware of is the `Neva` tool, a preliminary version of the NEXAS tool presented in [10].

The standard way to visualize extensions wrt. a semantics of an AF is by highlighting the accepted arguments in a directed graph, as shown in Figure 2b). In the following we discuss also related work on visualization for related domains, such as ASP. For example, `ARVis` [11] visualizes answer sets and their relations using a directed graph structure. Besides, the toolkit `Possible Worlds Explorer` (PWE) [12] is able to visualize individual answer sets and their structures. Additionally, PWE builds a distance matrix based on the similarity measure and applies hierarchically clustering on it to discover further substructures. On the other side, Betz et al. [13] implemented a tool based on argumentation theory that can structure and visualize multi-dimensional opinions by mapping them into two-dimensional space. Each opinion is a feature of the structured AF. They also cluster the opinion vectors (argumentation extensions) and calculate the coherence of opinion vectors. However, it is only suitable for very small data sets and focuses on the data relations.

Dimensionality Reduction. Dozens of dimensionality reduction techniques (also known as projections) are currently available and most are summarized in the surveys by Espadoto et al. [14] and Nonato et al. [15]. They are mainly used for exploration of multidimensional data, notably in AI applications and information visualization (InfoVis). For example, to explore gene expression patterns and correlate the results with classically defined neuroanatomy, Ji [16] projects hybridization gene expression data into a two-dimensional space using t-SNE and PCA algorithms and visualizes it using scatterplots. Among these techniques, the variants of MCA (*Multiple Correspondence Analysis*) [17,18] are among the few techniques capable to encoding categorical data to a visual space such that closeness reflects similarities [15].

Visualization of Highly Dimensional Data. From InfoVis, there are techniques tailored to highly dimensional data [19,20], like Parallel Coordinates, Prosection Views, Shape Coding, Recursive Pattern, etc. However, displaying a large amount of data objects with many features remains an open challenge where, usually, no single technique suffices. For this problem, tailoring multiple linked views has been proven to be effective [20], while also enabling the design of interactions and interplay between the views towards specific tasks. These design possibilities have recently been used to deal with specific use cases in various AI and formal methods domains (e.g., [21,22]). Furthermore, the visual complexity of the visualization plays an important role for explainability in AI, as expressed in the user needs of the What-If-Tool [23]. In fact, this tool successfully employs scatterplots, bar charts and line charts to provide insights about the model, data and features involved, when as stated before, one could opt for more intricate visualization techniques to deal with the complexity of the data.

Set Visualization. For the specific case of set-typed data, visualization techniques have also been studied well [24]. For example, based on Euler diagrams, EULERFORCE [25] uses a force-directed layout to optimize space usage, and SPEULER [26] includes semantic information in a two step layout method for highly overlapping sets. ONSET [27] uses a pixel-oriented technique to show which elements are included in the sets represented with juxtaposed grids, in contrast to the SET´O´GRAM approach [28], which uses superimposed bar charts to obtain insights. Other approaches use multiple coordinated views, like POWERSET [29], which uses bar charts and treemaps to show set intersections. Similar open challenges about the amount of data that can be visualized at once also apply to set visualization techniques, specially when many sets and their contents need to be compared. Thus, we chose familiar and effective representations that ease the cognitive load for users at different levels of expertise, as most of the existing work is best (or only) suited for small datasets.

3. NEXAS: Navigating and Exploring Argumentation Solution Spaces

So far, there is no tool that supports the solution space exploration for AFs. This means that in case there are hundreds or thousands of solutions, there is no way to see where for example two semantics coincide, or which arguments are accepted often or rarely.

Use Cases. In the following we describe typical Use Cases (UC) where the existing tool support is not enough. As a user for such scenarios we consider either i) a researcher that works in the field of abstract argumentation who uses argumentation systems to investigate certain aspects of argumentation semantics, or ii) a student in abstract argumentation who attends a course on argumentation theory and wants to learn more about the behavior of argumentation semantics. Thus, the tool is aimed to support the user visually in the analysis of the outcome on the computation of the argumentation semantics. The following use cases will help us to identify the goals on the visualization tool.

UC-1: The user wants to explore the AF solution space (the set of extensions) for specific argumentation semantics of a given AF. The main goal is to compare the semantics and to find out about similarities and differences. a) The individual acceptance rate of each argument should be given. The user wants to see the relative frequency of acceptance wrt. the semantics for each argument, and for the intersection of the extensions of both semantics. b) She wants to be able to select arguments and see in which extensions they are accepted. It is also desirable to be able to select several arguments and get the union of solutions highlighted where they are accepted. c) Each individual extension should be observable, and a representation that visualizes the complete solution space of two selected semantics is needed, which allows to identify which extensions share many arguments, and which have less in common.

UC-2: The solution space for one or both semantics might just be too big to be computed. On one side due to the time and space needed for the computation and on the other side because the information which needs to be processed is too big to be comprehensible. In most applications the user might just want to compute a particular subspace of the whole solution space, where some arguments are either contained in all or in none of the extensions.

UC-3: Besides highlighting the extensions where particular arguments are accepted (see UC-1), the user wants to "navigate" towards desired solution sub-spaces, by choosing which arguments should be accepted or rejected. To this end, the tool needs to show for which arguments such a selection is even possible. Those are arguments that are credulously but not skeptically accepted. Note that choosing skeptically accepted arguments would not change the solution space.

Design Goals. Within interdisciplinary brainstorming and feedback sessions involving experts in *Abstract Argumentation* as well as *Interactive Visualization*, we iteratively designed our NEXAS tool. Within these conversations, we considered needs of the argumentation community which resulted in developing a set of design goals (DG) that our tool must fulfill.

DG-1: Intuitive and Familiar Representations. We aim to foster intuitive understanding of the views by using traditional representations of the AF components while also encoding relevant information that the users can obtain insights from.

DG-2: Highlight Component Relations. A major challenge is to understand how components affect others. Thus, we aim to make these relations visible through linked interactions to foster understanding of the underlying framework.

DG-3: Maintainable and Customizable. The system design must be flexible and allow incorporation of further components in future iterations.

DG-4: Support Several Tasks and Workflows. We aim to support tasks with disjoint purposes and thus the available interactions must reflect such purposes.

DG-5: Ready-to-use. We aim to minimize setup complexity of the tool to account for various user environments.

4. Technical Design and Implementation

Figure 3 provides a simplified version of steps the data goes through in our system, also indicating the relationships between views.

Preprocessing Extensions. The backend uses the python library `pandas` [2] to perform operations on datasets during preprocessing. After computing the semantics using the ASP solver, we use a *one-hot-encoding* in order to construct two binary datasets D and D_s from the obtained extensions (answer sets). D simply encodes extensions as rows where columns correspond to arguments that belong to at least one extension (credulously accepted arguments) whose respective values $v \in \{0, 1\}$ indicate whether they are contained ($v = 1$) in the respective extensions, or not ($v = 0$). The other dataset D_s additionally contains flags for whether an extension is contained in one semantics, the other semantics or in both (the intersection) of them.

Dimensionality Reduction. Out of the presented reduction methods, t-SNE and MCA seem to be best suited for our application. Due to performance reasons we decided to use MCA in our tool. The two-dimensional reduction by MCA (*Multiple Correspondence Analysis*) using the python package `prince`[3] is performed on D and D_s, respectively. Ultimately the two-dimensional reduction provides the coordinates for the scatterplot that visualizes extensions. For D the dimensionality reduction is based on which arguments an extension contains, hence, the reduction reflects the similarity of extensions. For D_s the reduction in addition takes into account to which semantics an extension belongs. Performing a reduction on the latter data set may thus cause more distinctive clusters of data points in the scatterplot.

[2] https://pandas.pydata.org/ [3] https://github.com/MaxHalford/prince

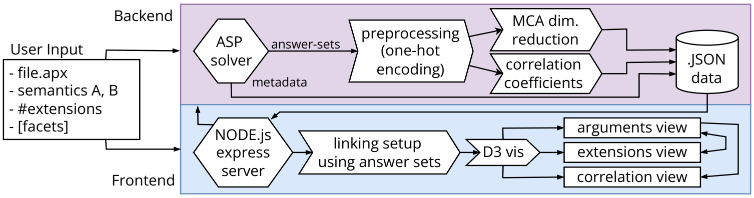


Figure 3. System diagram, describing our data management to build the visualizations.

Correlation. We compute correlation coefficients of arguments using the `pandas` library on dataset D, which simply reflects the containment of arguments among extensions. Datasets containing correlation coefficients are computed for each semantics, respectively, providing the frontend with data that can be used to visualize correlations in each of the semantics separately.

Navigating Extensions. We use the recently proposed ASP navigation framework [7] for navigating extensions of AFs. The main idea is to understand arguments that are credulously, but not skeptically accepted, as so called *facets* of the respective semantics of an AF, which can be included or excluded from the solution space in order to land in a sub-space of extensions. So called navigation steps are performed by inclusion or exclusion of facets. The inclusion of a facet prunes each extension that does not contain the corresponding argument, and accordingly, the exclusion of a facet prunes each extension that contains the corresponding argument. Navigation steps are achieved by adding corresponding integrity constraints to the respective answer set program that encodes the AF. By recursively enforcing that arguments are included in or excluded from extensions (answer sets), we can zoom into or out of the solution space, which ultimately provides us with functionality to navigate extensions. For more details on weighted faceted answer set navigation, interested readers are referred to [7].

Faceted navigation is realized visually in the frontend, using the one-hot-encoded data from the backend to perform navigation steps and display the resulting sub-spaces (addressing design goal DG-2). Furthermore, users can provide a list of facets to activate at startup (UC-2), to directly land in the resulting sub-space, which may be useful for very large solution space, as users can prune certain extensions right in the beginning.

Implementation Challenges. Several steps within the tool are computationally very expensive. This starts with the enumeration of extensions for the argumentation semantics, and continues with the processing of the data with dimensionality reduction and the visualization. Therefore, we preferred methods that are 1) efficient, 2) robust, 3) easy to adapt and 4) to combine (DG-3). To this end, we decided to use the `aspartix` [6] approach for the enumeration of extensions. Within the `aspartix` system suite, there are ASP encodings for most of the prominent semantics, and these encodings can be easily extended or modified independently from our tool NEXAS. As the `aspartix` approach relies on ASP technology, we can adapt the answer set navigation approach from [7] to allow a more interactive and informative exploration of the AF solution space. On the visualization front, to simplify access to the tool considering the work environments of our target users (DG-5), we provide NEXAS as a web application which—since it is limited by the resources of the browser tab—needs to avoid expensive operations and update the website components efficiently.

Data Generation. To generate the answer sets and miscellaneous data, we use a python script that calls the ASP solver `clingo` [30] together with the ASP encodings of `aspartix` [6] and the ASP solution space navigation approach [7]. For the computation of the semantics pairs we can often make use of the theoretical knowledge about the subset-relation between the semantics (see Figure 2). This means that for the pairs of semantics stable $\subseteq \sigma$ for $\sigma \in \{$semi-stable, stage, preferred, cf2$\}$, we only compute the extensions of the super-set, and identify within the answer sets obtained, which of the solutions are also an extension of the other semantics. The answer sets are finally stored in `.json` files for the frontend to use.

Web Application. NEXAS has a server implemented with `express.js`[4]. The HTML content is built using the templating engine `sprightly`[5] to facilitate component-based UI that can be easily maintained (DG-3). On the client side, the look and feel is achieved by adapting `Materialize`[6] components. The visualizations are built using `D3`[7], since it is flexible enough to support non-standard visual encodings.

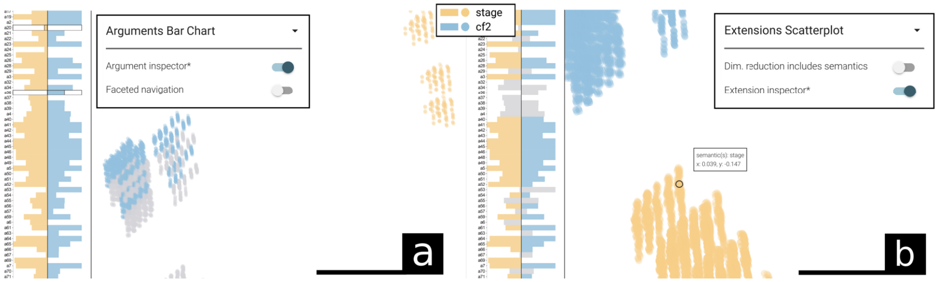
5. Visualization Design

The NEXAS tool consists of 3 major views to explore and navigate the solution space of an abstract argumentation framework. All views can be seen on Figure 1 and their influence on each other is encoded with arrows in Figure 3. The semantics are color encoded in all views (DG-2) using contrast to distinguish between them easily.

Argument View. This view encodes the set of arguments (Figure 1 a)). A double vertical bar chart is used to compare, for each argument, the relative frequency with which it is accepted in each of the two semantics ($\underline{A}$, $\underline{B}$). With a central axis used to distinguish between the semantics, we show A in the bars to the left and B in the bars to the right. The intersection of the semantics (i.e., the frequency with which an argument is accepted in both $\underline{A}$ and $\underline{B}$) is shown as a hatching pattern on both sides of the bar chart. The width of the intersection hatching can cover 100% of both sides of the chart, to indicate whether the intersection subsumes the solutions for either semantics. With this, we show at a glance how the semantics compare (DG-1). This bar chart can be used to filter and highlight content in the other views.

Extensions View. The solution space (i.e., set of extensions) is displayed in a scatterplot (Figure 1 b)). The extensions that belong either to $\underline{A}$ or $\underline{B}$ are encoded with the same colors as in the bar chart, and those that belong to both $\underline{A}$ and $\underline{B}$ are also encoded with a third color that visually blends the colors used for the two semantics. The MCA dimensionality reduction provides a spatial clustering based on the similarity of the contents of each solution, which also enables users to immediately get a feeling of how the extensions are distributed by semantics (DG-1). To show when several solutions are stacked on top of each other, we lowered the opacity of each point in the scatterplot. Furthermore, it is possible to include the semantics in the dimensionality reduction from the settings sidebar, emphasizing the grouping by semantics from the MCA. Users can also inspect the arguments contained in each solution by selecting them, which triggers highlights in the Argument View. This effect can be seen in Figure 4 b).

[4] `https://expressjs.com/`
[5] `https://www.npmjs.com/package/sprightly`
[6] `https://materializecss.com/`
[7] `https://d3js.org/`



Arguments Bar Chart
Argument inspector* ●
Faceted navigation ○

Extensions Scatterplot
Dim. reduction includes semantics ○
Extension inspector* ●

Figure 4. NEXAS interface showing a) the selection of a20 and 36 together with the extensions highlighted in the scatterplot in which they are accepted, b) the selection of one extension in the scatterplot together with the arguments highlighted in the argument bar char that are contained in it.

Correlation View. Figure 1 c) displays the correlation matrix split across the diagonal to compare the values in both semantics: A on the upper triangular matrix and B on the lower one. The semantics are shown in the color coding of the perimeter of both triangles (DG-2). Since the correlation matrix of a single semantics is symmetric, no information is lost. Additionally, hovering over any cell on the matrix highlights the same argument pair on the other side of diagonal, so that users can compare the correlations values (from -1 to 1) on demand. The vertical axis of the matrix of this view and bar chart of the Argument View correspond, such that hovering over an argument in the bar chart adds a visual guide over the row and column of the highlighted argument in the matrix.

Cross-view Interactions. We provide multiple ways of creating a selection of either arguments or extensions. We distinguish these as *Argument Inspector* and *Extension Inspector*, of which only one may be active at a time (DG-4). The user selection can be exported at any time depending on the active inspector. While the Argument Inspector is on, users may select arguments on the bar chart to highlight the extensions on the scatterplot that contain the selection (Figure 4 a)). For the Extension Inspector, the selection is done on the scatterplot and the result is reflected on the bar chart (Figure 4 b)).

On the other hand, the *Faceted Navigation* described in Section 3 can be enabled from the settings sidebar and alters the behavior of both the Argument and Extension Inspectors (to support UC-3). When active, the arguments which are not facets of the remaining extension set (i.e., the arguments are either contained in all solutions or none) are disabled. The Argument Inspector then allows users to include facets (click) or exclude them (ctrl + click), which updates the solution space by filtering the extensions accordingly. Multiple *simultaneous* selections are disallowed in Faceted Navigation, as after including or excluding a facet, the remaining arguments may not be facets of the resulting solution space. However, users can choose multiple facets *sequentially*, as the available facets update after each selection. Likewise, the Extension Inspector only allows one selected solution at a time while Faceted Navigation is active. For visual reference, see Figure 5.

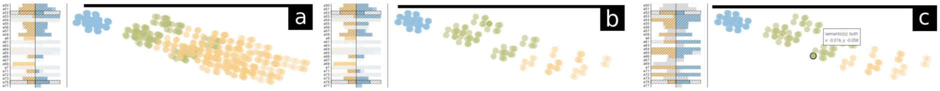


Figure 5. *Faceted Navigation* with a) the pre-selection of `a54` as Preset Facet, and included facets `a52`, `a21` and `a76`, b) after also including facet `a94`, and c) inspecting one extension using the *Extension Inspector*.

6. Validating NEXAS

Here we demonstrate the feasibility of our approach by means of a walk-through inspired by the use cases from Section 3. Initially the user provides the input AF [8]. All figures of the NEXAS interface use this input file. The upper bound[9] for the number of extensions to be computed for each of the semantics, respectively, is set to $10,000$. Further, semantics stage and cf2 are selected for the comparison.

After the data generation, the user sees the views as shown in Figure 1. As requested in UC-1 a) the argument view reveals that the semantics have no extension in common, as for none of the arguments there appears an overlay of stripes. When hovering over argument `a54`, a box appears and provides the information that it is accepted in 33.2% of the stage extensions and in 25.92% of the cf2 extensions. Further, one can observe that for instance argument `a32` is solely accepted in 50% of the stage extensions, and not for cf2 semantics. Following UC-1 b) the user clicks on `a20` and observes in the scatterplot that only the stage extensions, that contain `a20`, are displayed in color, all other points are grayed out. By further clicking on `a36`, additionally several cf2 extensions are again displayed in color, thus showing all extensions where `a20` or `a36` is accepted. After clearing the selection, the user switches to the Extension Inspector and clicks on one extension in the scatterplot. This made all arguments, that are not contained in that particular extension, grayed out in the argument bar chart, and thus all arguments that are still colored are accepted in that extension (UC-1 c)). Selecting another extension nearby the previous one reveals, as mentioned in UC-1 c), that extensions that are placed near to each other in the scatterplot share many arguments.

Considering UC-2, the user is interested in exploring the sub-space of solutions where argument `a54` is always accepted. She has two options, either to use faceted inspection, which then only considers the sub-space restricted to a maximum of $10,000$ for each of the semantics, or to restart the tool with a modified configuration where `a54` is activated as an inclusive facet so that all computed extensions contain `a54`. Using the second option, leads to sub-spaces that have 432 extensions in common. The argument `a32`, which previously was accepted in 50% of the stage extensions, is now accepted in 45.68%. The user selects `a52`, `a21` and `a76` in Faceted Navigation (UC-3), leading to the state visible in Figure 5 a). By further including facet `a94`, the number of extensions is reduced again as shown in Figure 5 b). Note that Faceted Navigation zooms in the solution space, thus all extensions that don't contain `a52`, `a21`, `a76` and `a94` are not displayed any longer, in contrast to the default behavior of the Argument Inspector that would gray out the other extensions. Next, the user observes one particular extension by switching to the Extension Inspector (while still having Faceted Navigation on), and clicking on an extension in the scatterplot. Then, in the argument bar chart on the left, all arguments are highlighted in color which are accepted in that extension (Figure 5 c)).

[8] `massachusetts_srta_2014-11-13.gml.50.apx` from the instances of ICCMA 2017 [31].

[9] Setting an upper bound for extensions could result in only a part of the solution space being considered.

7. Conclusion and Future Work

With our visual analysis tool NEXAS we propose a novel way of exploring the argumentation solution space, that offers a tight integration of state of the art solvers for AFs and interactive visualization design. Our validation shows that a user can easily observe information that, without the tool, would be very hard to comprehend, such as the relative frequency of acceptance of arguments, or which extensions belong to both semantics under consideration. The linked interactions within the three seamlessly integrated visualization views allow for multiple and advanced ways of solution space exploration.

For future work we will further refine and expand NEXAS, by integrating weights for facets, as described in [7]. Moreover, we will investigate how to represent the input AF in NEXAS, and how to integrate it in the views of the tool. User studies shall help us to integrate feedback from experts in the field of abstract argumentation. We believe that NEXAS opens up new ways to easily explore solution spaces, which will pave the way to also lift concepts from NEXAS to more general settings like ASP solution spaces.

8. Acknowledgements

This work was funded by the BMBF in projects 01IS20056_NAVAS and ScaDS.AI (Center for Scalable Data Analytics and Artificial Intelligence), by the Center for Advancing Electronics Dresden (cfaed), and by the DFG under Germany's Excellence Strategy: EXC-2068, 390729961 — Cluster of Excellence "Physics of Life" and EXC 2050/1, 390696704 — Cluster of Excellence "Centre for Tactile Internet" (CeTI) of TU Dresden, by DFG grant 389792660 as part of TRR 248 – CPEC.

References

[1] Baroni P, Gabbay DM, Giacomin M, van der Torre L. Handbook of Formal Argumentation. College Publications; 2018. Available from: https://books.google.de/books?id=_OnTswEACAAJ.

[2] Cerutti F, Gaggl SA, Thimm M, Wallner JP. Foundations of Implementations for Formal Argumentation. FLAP. 2017;4(8). Available from: http://www.collegepublications.co.uk/downloads/ifcolog00017.pdf.

[3] Lagniez J, Lonca E, Mailly J, Rossit J. Design and Results of ICCMA 2021. CoRR. 2021;abs/2109.08884., doi: 10.48550/arXiv.2109.08884.

[4] Janier M, Lawrence J, Reed C. OVA+: an Argument Analysis Interface. In: Parsons S, Oren N, Reed C, Cerutti F, editors. Proc. of COMMA 2014. vol. 266 of FAIA. IOS Press; 2014. p. 463-4., doi: 10.3233/978-1-61499-436-7-463.

[5] Bistarelli S, Rossi F, Santini F. ConArgLib: an argumentation library with support to search strategies and parallel search. J Exp Theor Artif Intell. 2021;33(6):891-918., doi: 10.1080/0952813X.2020.1789756.

[6] Dvorák W, Gaggl SA, Rapberger A, Wallner JP, Woltran S. The ASPARTIX System Suite. In: Prakken H, Bistarelli S, Santini F, Taticchi C, editors. Proc. of COMMA 2020. vol. 326 of FAIA. IOS Press; 2020. p. 461-2., doi: 10.3233/FAIA200534.

[7] Fichte JK, Gaggl SA, Rusovac D. Rushing and strolling among answer sets–navigation made easy. In: Proc. of the AAAI 2022. vol. 36; 2022. p. 5651-9., doi: 10.1609/aaai.v36i5.20506.

[8] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77(2):321-57., doi: 10.1016/0004-3702(94)00041-X.

[9] Baroni P, Caminada M, Giacomin M. An introduction to argumentation semantics. The knowledge engineering review. 2011;26(4):365-410., doi: 10.1017/S0269888911000166.

[10] Yang M, Gaggl SA, Rudolph S. Neva - Extension Visualization for Argumentation Frameworks. In: Prakken H, Bistarelli S, Santini F, Taticchi C, editors. Proc. of. COMMA 2020. vol. 326 of FAIA. IOS Press; 2020. p. 477-8., doi: 10.3233/FAIA200542.

[11] Ambroz T, Charwat G, Jusits A, Wallner JP, Woltran S. ARVis: Visualizing relations between answer sets. In: Cabalar P, Son TC, editors. Proc. of LPNMR 2013. Springer. Springer; 2013. p. 73-8., doi: 10.1007/978-3-642-40564-8_8.

[12] Gupta S, Cheng YY, Ludäscher B. Possible Worlds Explorer: Datalog & Answer Set Programming for the Rest of Us. In: Datalog; 2019. p. 44-55.

[13] Betz G, Hamann M, Mchedlidze T, von Schmettow S. Applying argumentation to structure and visualize multi-dimensional opinion spaces. Argument & Computation. 2019;10(1):23-40., doi: 10.3233/AAC-181004.

[14] Espadoto M, Martins RM, Kerren A, Hirata NST, Telea AC. Toward a Quantitative Survey of Dimension Reduction Techniques. IEEE Trans Vis Comput Graph. 2021;27(3):2153-73., doi: 10.1109/TVCG.2019.2944182.

[15] Nonato LG, Aupetit M. Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment. IEEE Trans Vis Comput Graph. 2019;25(8):2650-73., doi: 10.1109/TVCG.2018.2846735.

[16] Ji S. Computational genetic neuroanatomy of the developing mouse brain: dimensionality reduction, visualization, and clustering. BMC bioinformatics. 2013;14(1):1-14., doi: 10.1186/1471-2105-14-222.

[17] Greenacre M, Blasius J. Multiple correspondence analysis and related methods. Chapman and Hall/CRC; 2006., doi: 10.1201/9781420011319.

[18] Tenenhaus M, Young FW. An analysis and synthesis of multiple correspondence analysis, optimal scaling, dual scaling, homogeneity analysis and other methods for quantifying categorical multivariate data. Psychometrika. 1985 Mar;50(1):91-119., doi: 10.1007/BF02294151.

[19] Liu S, Maljovec D, Wang B, Bremer PT, Pascucci V. Visualizing High-Dimensional Data: Advances in the Past Decade. IEEE Trans Vis Comput Graph. 2017;23(3):1249-68., doi: 10.1109/TVCG.2016.2640960.

[20] Keim DA. Information visualization and visual data mining. IEEE Trans Vis Comput Graph. 2002;8(1):1-8., doi: 10.1109/2945.981847.

[21] Horak T, Coenen N, Metzger N, Hahn C, Flemisch T, Méndez J, et al. Visual Analysis of Hyperproperties for Understanding Model Checking Results. IEEE Trans Vis Comput Graph. 2022 10., doi: 10.1109/TVCG.2021.3114866.

[22] Chegini M, Bernard J, Berger P, Sourin A, Andrews K, Schreck T. Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning. Visual Informatics. 2019;3(1):9-17., doi: 10.1016/j.visinf.2019.03.002.

[23] Wexler J, Pushkarna M, Bolukbasi T, Wattenberg M, Viégas F, Wilson J. The What-If Tool: Interactive Probing of Machine Learning Models. IEEE Trans Vis Comput Graph. 2020;26(1):56-65., doi: 10.1109/TVCG.2019.2934619.

[24] Alsallakh B, Micallef L, Aigner W, Hauser H, Miksch S, Rodgers P. The State-of-the-Art of Set Visualization. Computer Graphics Forum. 2016;35(1):234-60., doi: 10.1111/cgf.12722.

[25] Micallef L, Rodgers P. eulerForce: Force-directed layout for Euler diagrams. Journal of Visual Languages & Computing Distributed Multimedia Systems DMS2014 Part I. 2014;25(6):924-34., doi: 10.1016/j.jvlc.2014.09.002.

[26] Kehlbeck R, Görtler J, Wang Y, Deussen O. SPEULER: Semantics-preserving Euler Diagrams. IEEE Trans Vis Comput Graph. 2022;28(1):433-42., doi: 10.1109/TVCG.2021.3114834.

[27] Sadana R, Major T, Dove A, Stasko J. OnSet: A Visualization Technique for Large-scale Binary Set Data. IEEE Trans Vis Comput Graph. 2014;20(12):1993-2002., doi: 10.1109/TVCG.2014.2346249.

[28] Freiler W, Matkovic K, Hauser H. Interactive Visual Analysis of Set-Typed Data. IEEE Trans Vis Comput Graph. 2008;14(6):1340-7., doi: 10.1109/TVCG.2008.144.

[29] Alsallakh B, Ren L. PowerSet: A Comprehensive Visualization of Set Intersections. IEEE Trans Vis Comput Graph. 2017;23(1):361-70., doi: 10.1109/TVCG.2016.2598496.

[30] Gebser M, Kaminski R, Kaufmann B, Lühne P, Obermeier P, Ostrowski M, et al. The Potsdam Answer Set Solving Collection 5.0. Künstliche Intell. 2018;32(2-3):181-2., doi: 10.1007/s13218-018-0528-x.

[31] Gaggl SA, Linsbichler T, Maratea M, Woltran S. Design and results of the Second International Competition on Computational Models of Argumentation. Artif Intell. 2020;279., doi: 10.1016/j.artint.2019.103193.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220147

Non-Admissibility in Abstract Argumentation

New Loop Semantics, Overview, Complexity Analysis

Wolfgang DVOŘÁK [a], Tjitze RIENSTRA [b], Leendert VAN DER TORRE [c,d], and Stefan WOLTRAN [a]

[a] *TU Wien*
[b] *Maastricht University*
[c] *University of Luxembourg*
[d] *Zhejiang University*

Abstract. In this paper, we give an overview of several recent proposals for non-admissible non-naive semantics for abstract argumentation frameworks. We highlight the similarities and differences between weak admissibility-based approaches and undecidedness-blocking approaches using examples and principles as well as a study of their computational complexity. We introduce a kind of strengthened undecidedness-blocking semantics combining some of the distinctive behaviours of weak admissibility-based semantics with the lower complexity of undecidedness-blocking approaches. We call it loop semantics, because in our new semantics, an argument can only be undecided if it is part of a loop of undecided arguments. Our paper shows how a principle-based approach and a complexity-based approach can be used in tandem to further develop the foundations of formal argumentation.

Keywords. abstract argumentation, semantics, complexity, weak admissibility

1. Introduction

Dung's admissibility-based (AB) semantics have been challenged in various ways, leading to a variety of new semantics [8,9,11,5]. These have been compared and classified on the basis of general principles as well as their computational complexity, such that the best semantics can be chosen for an application. Two desirable properties stand out. First, directionality together with SCC recursion lead to a kind of causal interpretation of attack [6] and allow for compositional computation [4]. Second, low computational complexity is not only advantageous for automated reasoning by artificial agents, but from the perspective of cognitive science, it also increases explainability by and for humans.

The best known non-admissible semantics are naive-based (NB) semantics. Under an NB semantics, each extension is a maximal conflict-free set of arguments. The most prominent example of an NB semantics is the CF2 semantics [3]. To illustrate the core idea, consider the framework (a) from Figure 1. In AB semantics, the only extension is the empty set, whereas under the CF2 semantics, *b* is accepted. To get the desirable properties of directionality and SCC-recursiveness, CF2 is defined in terms of a local

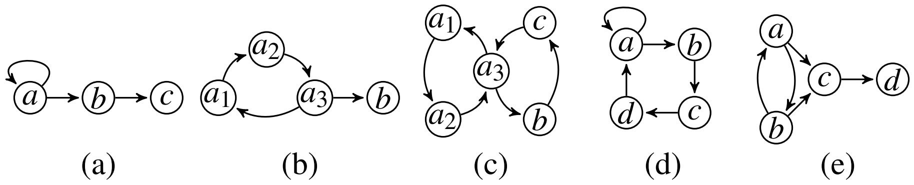


Figure 1. Five argumentation frameworks.

function that computes the maximal conflict-free subsets for each strongly connected component (or SCC) of a framework.

More recently, two new types of non-admissible semantics were introduced. They were motivated by the behaviour of AB semantics in examples such as framework (b) in Figure 1. Here, the set $\{b\}$ is not admissible since it does not defend itself from its attacker a_3. Nevertheless, one can argue that b is acceptable since a_3, being part of an odd-length cycle of arguments that are never accepted, does not pose an actual threat. Capturing this intuition thus requires a different notion of admissibility. The first type takes a reduction-based approach and is called weak admissibility (WA) [5]. The second type takes a labelling-based approach to define weaker acceptance criteria, called "undecidedness blocking" (UB) [11], which is analogous to "ambiguity blocking" and discussed in defeasible logics [15]. In contrast to the AB labelling approach, an undecided argument in UB may attack arguments that are labelled in. Further semantics that belong to the UB approach are the qualified and semi-qualified semantics [8]. These are defined by adapting the SCC recursive algorithm but keeping the base function admissible. The WA, UB and (semi-)qualified semantics all come in a grounded, complete and preferred flavour. These developments raise two questions: (1) How do these approaches compare in terms of examples, principles, and computational complexity? And (2) Which new kinds of semantics can be explored based on this overview?

Concerning the first question, the semantics of WA and UB approaches are remarkably similar for most benchmark examples. For example, they give the same results for the frameworks (b) and (c) from Figure 1. Moreover, both WA and UB preferred semantics are SCC-recursive and directional. The main distinction is in their computational complexity. As we show in this paper, the UB approach has a significantly lower computational complexity than the WA approach, for which we show PSPACE-completeness also for recently introduced variants (thus complementing the results in [14]).

The second question is how our analysis can be used to define new semantics. In particular, we are interested in approaches that combine the behaviour of WA semantics with the lower complexity of UB approaches. We define a new kind of semantics called loop semantics that extends UB with a new condition that ensures that arguments can only be undecided if they are part of a loop of undecided arguments. In this sense, the role of undecided arguments is to detect loops. We also define a notion of UB-admissibility, a concept that was missing thus far in the definition of UB semantics.

In this paper, we only consider non-admissible and non-naive based semantics. We do not consider the SCF2 semantics introduced by Cramer and van der Torre [7]. We focus our complexity analysis on complete and preferred variants of various semantics. Due to space limitations, we do not repeat all the variants of WA semantics described by Dauphin et al. [9], but we do include their semantics in the complexity analysis. For the same reason, for some of the proofs, we are only able to include proof sketches.

The layout of this paper is as follows. We first provide a brief overview of semantics based on weakly admissible and undecidedness blocking in Sections 2 and 3. We then present our new loop semantics in Section 4. All these semantics are illustrated using examples and principles. In Section 6 we provide a complexity overview of fifteen different kinds of non-admissible, non-naive semantics. We conclude in Section 7.

2. Weakly Admissible Semantics

An *argumentation framework* (abbreviated as AF) is a pair $F=(A,\rightarrow)$ where A is a set of arguments and $\rightarrow\subseteq A\times A$ the attack relation. We assume in this paper that A is finite. A set $E\subseteq A$ is *conflict-free* if there are no $x,y\in E$ such that $x\rightarrow y$. A set $E\subseteq A$ *defends* an argument $x\in A$ if for all $y\in A$ such that $y\rightarrow x$, there is a $z\in E$ such that $z\rightarrow y$ [12].

We focus in this paper on new variants of the *admissible*, *complete* and *preferred* semantics. The classical variants, denoted respectively by **ad**, **co** and **pr**, are defined as follows [12]. Let $F=(A,\rightarrow)$ be an AF. An **ad** extension of F is a set $E\subseteq A$ that is conflict-free and defends all its members. A **co** extension is an admissible extension that contains all arguments it defends. A **pr** extension is a maximal admissible extensions. In what follows we use, given a semantics σ, $\sigma(F)$ to denote the set of σ extensions of F.

Baumann, Brewka and Ulbricht [5] defined *weak admissibility* based on the principle that, given an AF $F=(A,\rightarrow)$ and set $E\subseteq A$, an argument needs to be defended by E only from arguments that are 'serious' in the sense that they appear in a weakly admissible set of the E*-reduct* of F. The E-reduct of F is denoted by F^E and defined by $F^E=(E^*,\rightarrow\cap(E^*\times E^*))$ where $E^*=A\setminus(E\cup E^+)$ and $E^+=\{y\in A|\exists x\in E,x\rightarrow y\}$. Let $F=(A,\rightarrow)$ be an AF. A set $E\subseteq A$ is weakly admissible (i.e., $E\in\mathbf{ad}^w(F)$) if and only if E is conflict free and for every attacker y of E we have $y\notin\cup\mathbf{ad}^w(F^E)$. To define weakly complete and preferred semantics we first define *weak defence*.

Definition 1 *[5] Let $F=(A,\rightarrow)$ be an AF. A set $E\subseteq A$* weakly defends *a set $X\subseteq A$ whenever, for every attacker y of X, either E attacks y, or it is the case that $y\notin\cup\boldsymbol{ad}^w(F^E)$, $y\notin E$, and $X\subseteq X'\in\boldsymbol{ad}^w(F)$.*

Definition 2 *[5] Let $F=(A,\rightarrow)$ be an AF and $E\subseteq A$. The weakly complete and weakly preferred semantics $\boldsymbol{co}^w$ and $\boldsymbol{pr}^w$ are defined as follows: $E\in\boldsymbol{co}^w(F)$ iff $E\in\boldsymbol{ad}^w(F)$ and for every X such that $E\subseteq X$ that is w-defended by E, we have $X\subseteq E$; and $E\in\boldsymbol{pr}^w(F)$ iff E is $\subseteq$-maximal in $\boldsymbol{ad}^w(F)$.*

Dauphin et al. [9] defined several variants of the weak admissibility based semantics. We omit the definitions due to space constraints but we include the complete variants denoted $\mathbf{co}^{w\forall}$, $\mathbf{co}^{\exists}$ and $\mathbf{co}^{\forall}$ in Table 1 as well as our complexity analysis in Section 6.

We conclude this section by pointing out a notable difference between the weakly complete and preferred semantics with respect to the *directionality* principle [2]. Given an AF $F=(A,\rightarrow)$ and a set $U\subseteq A$, we use $F{\downarrow_U}$ to denote the AF $(U,\rightarrow\cap U\times U)$ and, given a set $X\subseteq 2^A$, we use $X{\downarrow_U}$ to denote the set $\{E\cap U|E\in X\}$. A semantics σ is directional if, for every AF $F=(A,\rightarrow)$ and every unattacked set U of F (i.e., any set $U\subseteq A$ such that $x\in U$ and $y\rightarrow x$ implies that $y\in U$) we have that $E\in\sigma(F){\downarrow_U}=\sigma(F{\downarrow_U})$. The preferred and complete semantics both satisfy directionality [2]. However, while the weakly preferred semantics also satisfies directionality [5], the weakly complete semantics does not [8].

Table 1. Various semantics applied to the AFs from Figure 1.

Semantics	(a)	(b)	(c)	(d)	(e)
co [12]	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$, $\{b,d\}$	$\emptyset$, $\{a,d\}$, $\{b,d\}$
pr [12]	$\emptyset$	$\emptyset$	$\emptyset$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$
ad^w [5]	$\emptyset, \{b\}$	$\emptyset$, $\{b\}$	$\emptyset$, $\{a_1\}$, $\{b\}$	$\emptyset$, $\{d\}$, $\{b\}$, $\{b,d\}$	$\emptyset$, $\{a\}$, $\{b\}$, $\{a,d\}$, $\{b,d\}$, $\{d\}$
co^w [5]	$\{b\}$	$\{b\}$	$\{a_1\}$, $\{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{d\}$
pr^w [5]	$\{b\}$	$\{b\}$	$\{a_1\}$, $\{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$
co$^{w\forall}$ [9]	$\{b\}$	$\{b\}$	$\emptyset, \{a_1\}, \{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{d\}$
co$^{\exists}$ [9]	$\{b\}$	$\{b\}$	$\{a_1\}, \{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{d\}$
co$^{\forall}$ [9]	$\{b\}$	$\{b\}$	$\emptyset, \{a_1\}, \{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{d\}$
q-co	$\{b\}$	$\{b\}$	$\emptyset$	$\emptyset$, $\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
q-pr	$\{b\}$	$\{b\}$	$\emptyset$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$
sq-co	$\{b\}$	$\{b\}$	$\emptyset$	$\emptyset$, $\{b,d\}$	$\emptyset$, $\{a,d\}$, $\{b,d\}$
sq-pr	$\{b\}$	$\{b\}$	$\emptyset$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$
ub-co	$\emptyset, \{c\}, \{b\}$	$\emptyset$, $\{b\}$	$\emptyset$, $\{a_1\}$,$\{b\}$	$\emptyset$, $\{b,d\}$, $\{c\}$	$\emptyset$, $\{a,d\}$, $\{b,d\}$, $\{c\}$, $\{d\}$
ub-pr	$\{c\}$,$\{b\}$	$\{b\}$	$\{a_1\}$,$\{b\}$	$\{b,d\}$, $\{c\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub2-co	$\{b\}$	$\{b\}$	$\emptyset$, $\{a_1\}$,$\{b\}$	$\emptyset$, $\{b,d\}$, $\{c\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub2-pr	$\{b\}$	$\{b\}$	$\{a_1\}$,$\{b\}$	$\{b,d\}$, $\{c\}$	$\{a,d\}$, $\{b,d\}$
ub*-co	$\{b\}$	$\{b\}$	$\emptyset$, $\{a_1\}$,$\{b\}$	$\emptyset$, $\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub*-pr	$\{b\}$	$\{b\}$	$\{a_1\}$,$\{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub2*-co	$\{b\}$	$\{b\}$	$\emptyset$, $\{a_1\}$,$\{b\}$	$\emptyset$, $\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub2*-pr	$\{b\}$	$\{b\}$	$\{a_1\}$,$\{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$

3. Existing Semantics Based On Undecidedness Blocking

We now review a number of recently proposed semantics that are based, like weak admissibility, on weaker acceptance criteria. They are defined in terms of labellings. A labelling L of an AF F is a function that maps each argument of F to a label $\mathtt{I}$ (in, or accepted), $\mathtt{O}$ (out, or rejected) or $\mathtt{U}$ (undecided). We use $\mathscr{L}(F)$ to denote the set of all labellings of F. A labelling-based semantics σ maps each AF F to a set $\mathscr{L}_\sigma(F) \subseteq \mathscr{L}(F)$. A labelling L corresponds to the extension containing all arguments labelled $\mathtt{I}$ by L.

The semantics we review in this section are based on an “undecidedness blocking” mechanism. While in an admissible labelling, an argument is labelled $\mathtt{I}$ only if all its attackers are labelled $\mathtt{O}$, these semantics allow $\mathtt{I}$-labelled arguments to be attacked by $\mathtt{U}$-labelled arguments. The semantics we discuss differ in the conditions under which this is allowed. As we will see, these semantics are, for most benchmark examples, remarkably similar to the weak admissibility based semantics from the previous section. We start with the *qualified* and *semi-qualified* schemes due to Dauphin et al. [8]. Both schemes rely on the decomposition of an AF into its SCCs (strongly connected components). We denote the set of the SCCs of F by $SCCS(F)$. Given an AF $F = (A, \rightarrow)$, an *outparent* of an SCC S of F is an argument $x \in A \setminus S$ such that $x \rightarrow y$ for some $y \in S$. We denote by $OP_F(S)$ the set of the outparents of S. Given a labelling $L \in \mathscr{L}(F)$, we denote by $L{\downarrow_S}$ the restriction of L to S and, given a set $X \subseteq \mathscr{L}(F)$, denote by $X{\downarrow_S}$ the set $\{L{\downarrow_S} \mid L \in X\}$.

Qualified Semantics The qualified scheme is based on the *SCC decomposability* principle, which states that the set of the labellings of an AF F is decomposable into the product of the labellings of each SCC S as a function of the labels of the outparents of S.

Definition 3 *An* AF with input *is a tuple* $(F, A_{in}, \rightarrow_{in}, L_{in})$ *where:* $F = (A, \rightarrow)$ *is an AF,* A_{in} *is a set of* input arguments *such that* $A \cap A_{in} = \emptyset$*,* $\rightarrow_{in} \subseteq A_{in} \times A$ *is an* input attack relation*, and* $L_{in} \in \mathscr{L}(A_{in})$ *is an* input labelling*. A* local function *f assigns to every AF with the input* $(F, A_{in}, \rightarrow_{in}, L_{in})$ *a set* $f(F, A_{in}, \rightarrow_{in}, L_{in}) \subseteq \mathscr{L}(F)$*. We say that f* represents *the semantics* σ *if for every AF F,* $L \in \mathscr{L}_\sigma(F)$ *if and only if* $\forall S \in SCCS(F)$*,* $L{\downarrow_S} \in f(F{\downarrow_S}, OP_F(S), \rightarrow \cap OP_F(S) \times S, L{\downarrow_{OP_F(S)}})$*. A semantics* σ *is* SCC decomposable *if it is represented by some local function.*

Examples of SCC decomposable semantics are the complete and preferred semantics. We denote by $f_{\mathbf{co}}$ and $f_{\mathbf{pr}}$ the local functions representing these semantics. Their definitions can be found in [1]. The qualified variant of an SCC decomposable semantics σ (denoted as **q-**σ) is based on applying the local function representing σ with one change: when determining the labellings of an SCC S, the label U for an outparent x of S is treated like the label O. Thus, if x is attacked by an U-labelled argument y, and x and y belong to different SCCs, then the undecidedness of x does not propagate to y.

Definition 4 *[8] Let* σ *be an SCC-decomposable semantics represented by the local function* f_σ*. We define the* qualified σ *(**q-**σ) semantics as the semantics represented by the local function* $f_{\mathbf{q}\text{-}\sigma}$ *defined by* $f_{\mathbf{q}\text{-}\sigma}((A, \rightarrow), A_{in}, \rightarrow_{in}, L_{in}) = f_\sigma((A, \rightarrow), A_{in}, \rightarrow_{in}, L'_{in})$*, where* L'_{in} *is defined by* $L'_{in}(x) = \mathtt{I}$ *if* $L_{in}(x) = \mathtt{I}$*, and* $L'_{in}(x) = \mathtt{O}$ *if* $L_{in}(x) = \mathtt{O}$ *or* $L_{in}(x) = \mathtt{U}$.

Table 1 shows the **q-co** and **q-pr** extensions of the AFs from Figure 1. Here, we see that the **q-pr** extensions coincide with the weakly preferred extensions of AFs (a), (d) and (e), and that the **q-co** extensions coincide with the weakly complete extensions of AF (a). AF (e) demonstrates a crucial difference compared to weak admissibility. Here, $\{c\}$ is not weakly admissible because this set does not defend itself from a and b, both of which appear in weakly admissible sets of the $\{c\}$-reduct. However, under **q-co** semantics, the undecidedness of a and b does not propagate to c, as witnessed by the **q-co** labelling $(a = \mathtt{U}, b = \mathtt{U}, c = \mathtt{I}, d = \mathtt{O})$ with corresponding extension $\{c\}$.

Semi-qualified Semantics In the semi-qualified scheme, the label U of an outparent is treated like the label O, but *only if there is no other labelling in which that outparent is labelled* I. This is formalised using the notion of *weak SCC decomposability*.

Definition 5 *An* AF with total input *is a tuple* $(F, A_{in}, \rightarrow_{in}, L_{in}, S_{in})$ *where* $F, A_{in}, \rightarrow_{in}$ *and* L_{in} *are defined as in Definition 3,* $S_{in} \subseteq \mathscr{L}(A_{in})$*, and* $L_{in} \in S_{in}$*. We call* S_{in} *the* total input labellings *and* $L_{in} \in S_{in}$ *the* actual input labelling*. A* weak local function *g assigns to every AF with total input* $(F, A_{in}, \rightarrow_{in}, L_{in}, S_{in})$ *a set* $g(F, A_{in}, \rightarrow_{in}, L_{in}, S_{in}) \subseteq \mathscr{L}(F)$*. A weak local function g represents a semantics* σ *whenever, for every AF F,* $L \in \mathscr{L}_\sigma(F)$ *if and only if* $\forall S \in SCCS(F)$*,* $L{\downarrow_S} \in g(F{\downarrow_S}, OP_F(S), \rightarrow \cap OP_F(S) \times S, L{\downarrow_{OP_F(S)}}, \mathscr{L}_\sigma(F){\downarrow_{OP_F(S)}})$*. A semantics* σ *is* weakly SCC-decomposable *if some weak local function represents* σ*.*

Definition 6 *[8] Let* σ *be an SCC-decomposable semantics. Let* f_σ *denote the local function that represents* σ*. We define the* semi-qualified σ *(or **sq-**σ) semantics as the semantics represented by the weak local function* $g_{\mathbf{sq}\text{-}\sigma}$ *defined by* $g_{\mathbf{sq}\text{-}\sigma}((A, \rightarrow), A_{in}, \rightarrow_{in}, L_{in}, S_{in}) = g_\sigma((A, \rightarrow), A_{in}, \rightarrow_{in}, L'_{in})$*, where* L'_{in}*is defined by:* $L'_{in}(x) = \mathtt{I}$ *if* $L_{in}(x) = \mathtt{I}$*;* $L'_{in}(x) = \mathtt{O}$ *if* $L_{in}(x) = \mathtt{O}$*;* $L'_{in}(x) = \mathtt{O}$ *if* $L_{in}(x) = \mathtt{U}$ *and there is no* $L \in S_{in}$ *such that* $L(x) = \mathtt{I}$*; and* $L'_{in}(x) = \mathtt{U}$ *if* $L_{in}(x) = \mathtt{U}$ *and there is some* $L \in S_{in}$ *such that* $L(x) = \mathtt{I}$.

Table 1 shows the **sq-co** and **sq-pr** extensions of the AFs shown in Figure 1. Note that the AF (e) no longer has a **sq-co** labelling where c is accepted. The **sq-co** and **sq-pr** semantics are different from the weak admissibility-based semantics in that the AF (c) only has an empty **sq-co** and **sq-pr** extension. This is because the (semi-)qualified scheme applies its base semantics unchanged to single-SCC AFs. The UB semantics that we present next addresses this problem. Before moving on we note that all variants of the (semi-)qualified semantics considered here satisfy directionality [8].

UB and UB2 Semantics Dondio and Longo [11] proposed a semantics based on the following definition. Note that this definition equals that of a standard complete labelling if we add as a third condition that an argument is labelled `I` only if all its attackers are labelled `O`. This semantics can thus be understood as a variant of the complete semantics where `I`-labelled arguments may also be attacked by `U`-labelled arguments. We refer to this semantics as *UB* semantics and define a complete and preferred variant.

Definition 7 *Let $F = (A, \rightarrow)$ be an AF. A* **ub-co** *labelling of F is a labelling L such that (1) $L(x) = \mathtt{O}$ iff for some $y \in A$ s.t. $y \rightarrow x$, $L(y) = \mathtt{I}$, and (2) if $L(x) = \mathtt{U}$ then for some $y \in A$ s.t. $y \rightarrow x$, $L(y) = \mathtt{U}$. A **ub-pr** labelling is a **ub-co** labelling that is maximal with respect to `I`-labelled arguments.*

Looking at Table 1, we see that the weakly preferred and **ub-pr** extensions of the single-SCC AF (c) coincide. One way in which UB semantics diverges from weak admissibility is demonstrated by AF (a). Here, we have not only extension $\{b\}$ but also $\{c\}$ and $\emptyset$. This behaviour is due to the fact that, under UB semantics, the undecidedness of a may be blocked not only by b, but also by c or not at all. To avoid this behaviour, and to enforce the rule that undecidedness is blocked as early as possible, Dondio and Lungo propose to combine UB semantics with the SCC-recursive scheme [3]. These semantics, which we refer to as *UB2*, are directional by virtue of being SCC-recursive.

Definition 8 *Let $F = (A, \rightarrow)$ be an AF. The **ub2-co** semantics is defined by $L \in \mathscr{L}_{\textbf{ub2-co}}(F)$ iff the following conditions hold: If $|SCCS(F)| = 1$, then $L \in \mathscr{L}_{\textbf{ub-co}}(F)$; If $|SCCS(F)| > 1$, then for all $S \in SCCS(F)$: (1) $L{\downarrow}_{(S \setminus D_F(S,L))} \in \mathscr{L}_{\textbf{ub2-co}}(F{\downarrow}_{(S \setminus D_F(S,L))})$, and (2) $\forall x \in D_F(S,L), L(x) = \mathtt{O}$, where $D_F(S,L) = \{x \in S | \exists y \in A \setminus S, y \rightarrow x, L(y) = \mathtt{I}\}$ denotes the set of arguments in S that are attacked by an accepted argument not in S. The **ub2-pr** is defined similarly by replacing **co** with **pr** in this definition.*

Indeed, looking again at Table 1, the **ub2-pr** semantics coincides with the weakly preferred semantics in all of the examples we consider except for AF (d), where we see similar behaviour to that under **ub-pr** semantics: the undecidedness of a may be blocked not only by b, but also by c or not at all. The reason is that AF (d) consists of a single SCC, which leads to the same extensions under UB and UB2 semantics.

4. A New Semantics Based on Undecidedness Blocking

We now propose a new variant of UB semantics called *UB**. Our aim is to ensure that only arguments that are part of a loop can be undecided. While UB semantics requires that undecided arguments are *attacked by* an undecided argument, UB* semantics require that they also *attack* an undecided argument.

Definition 9 *Let $F = (A, \rightarrow)$ be an AF. A* ***ub*-co*** *labelling of F is a* ***ub-co*** *labelling of F such that, for all $x \in A$, if $L(x) = \mathtt{U}$, then $L(y) = \mathtt{U}$ for some $y \in A$ s.t. $x \rightarrow y$. A* ***ub*-pr*** *labelling is a* ***ub*-co*** *labelling that is maximal with respect to* $\mathtt{I}$*-labelled arguments.*

Looking again at Table 1, we see that **ub*-co** and **ub*-pr** semantics of the AF (d) no longer include $\{c\}$ as an extension. Unfortunately, this scheme is not yet sufficient to ensure that only arguments that are part of a loop can be undecided. It also allows arguments to be labelled $\mathtt{U}$ if they lie on a directed path between two cycles. For instance, the AF $(\{a,b,c\},\{(a,a),(a,b),(b,c),(c,c)\})$ has a **ub-co** labelling where b is labelled $\mathtt{U}$. Another problem is that the **ub*-pr** semantics is not directional. To see why, let F be the AF (e) from Figure 1. This AF has a **ub*-pr** labelling where a and b are $\mathtt{U}$, while $F{\downarrow}\{a,b\}$ does not have a **ub*-pr** labelling where a and b are $\mathtt{U}$. To ensure that an argument is undecided only if it is part of a cycle, we combine the UB* semantics with the SCC-recursive scheme. We refer to them as *loop semantics*. The complete and preferred loop semantics, denoted **ub2*-co** and **ub2*-pr**, are defined as in Definition 8 but replacing $\mathscr{L}_{\textbf{ub-co}}(F)$ and $\mathscr{L}_{\textbf{ub-pr}}(F)$ in condition 1 with $\mathscr{L}_{\textbf{ub*-co}}(F)$ and $\mathscr{L}_{\textbf{ub*-co}}(F)$. By virtue of being SCC-recursive, both of these semantics are directional. Furthermore, arguments are labelled $\mathtt{U}$ only if they are part of a loop. For example, the AF $(\{a,b,c\},\{(a,a),(a,b),(b,c),(c,c)\})$ does not posses a **ub*-co** labelling in which b is labelled $\mathtt{U}$.

5. UB-Admissibility

While Dondio et al. [11] define complete and preferred variants of the UB semantics, as well as a grounded variant (defined as an $\mathtt{U}$-maximal **ub-co** labelling) they do not define the concept UB admissibility. Note that the notion of "weak admissibility" defined in [10] is in fact a kind of UB completeness, and was renamed as such in [11]. Here we propose a notion of admissibility corresponding to the UB semantics. Having such a notion is useful because, as in Dung's semantics, admissibility leads to local explanations as to why an argument is acceptable in AB semantics. That is, to check whether an argument belongs to a complete extension, we do not have to compute complete extension in their entirety: all we need to do is to find an admissible set containing the argument.

Standard admissibility is defined in terms of conflict-freeness and defence. In WA semantics, however, it is the other way around, in the sense that WA defence is defined in terms of weak admissibility. In both AB and WA approaches, complete extensions are defined in terms of (regular or weak) admissibility. In the definition of UB admissibility we introduce here, it is the other way around. We define UB admissibility in terms of complete UB extensions. The following definition defines UB-admissibility in terms of the UB-preferred semantics. Note that the same definition can also be applied in combination with the other semantics we have defined.

Definition 10 *Let $F = (A, \rightarrow)$ be an AF. An* UB-admissible *set of F is a set $E \subseteq A$ such that for some UB-preferred set E' of F we have that (1) $E \subseteq E'$; and (2) for all $x \in A$ such that $x \rightarrow E$ and $E' \rightarrow x$, we have $E \rightarrow x$.*

To illustrate, consider the AF (e) from Figure 1, which has the UB-admissible extensions $\emptyset$, $\{a\}$, $\{b\}$, $\{a,d\}$, $\{b,d\}$ and $\{c\}$, where only the latter three are UB-preferred. We leave a more detailed study of this notion of admissibility for future work.

Table 2. Complexity of weak-admissible based semantics compared to classical semantics (“c” is used as a shorthand for “complete”)

σ	$Cred_\sigma$	$Scept_\sigma$	Ver_σ
co	NP-c	P-c	in P
pr	NP-c	Π_2^P-c	coNP-c
ad^w	PSPACE-c	trivial	PSPACE-c
co^w	PSPACE-c	PSPACE-c	PSPACE-c
pr^w	PSPACE-c	PSPACE-c	PSPACE-c
co$^{w\forall}$	PSPACE-c	PSPACE-c	PSPACE-c
co$^{\exists}$	PSPACE-c	PSPACE-c	PSPACE-c
co$^{\forall}$	PSPACE-c	PSPACE-c	PSPACE-c
q-co	NP-c	P-c	in P
q-pr	NP-c	Π_2^P-c	coNP-c

σ	$Cred_\sigma$	$Scept_\sigma$	Ver_σ
sq-co	Δ_2^P-c	in Δ_2^P	in Δ_2^P
sq-pr	Δ_2^P-c	Π_2^P-c	in Δ_2^P
ub-co	in P	in P	in P
ub-pr	in P	coNP-c	in P
ub2-co	NP-c	in P	in P
ub2-pr	NP-c	coNP-c	in P
ub*-co	NP-c	in P	in P
ub*-pr	NP-c	in Π_2^P	in coNP
ub2*-co	NP-c	in P	in P
ub2*-pr	NP-c	in Π_2^P	in coNP

6. Complexity Results

We start by briefly recalling some complexity classes. We assume that the reader is familiar with the basic concepts of computational complexity theory (see e.g. [13]) as well as the standard classes P, NP and coNP. In addition, we will consider the classes: $\Delta_2^P = P^{NP}$ of problems that can be solved in deterministic polynomial time when the algorithm has access to an NP oracle; $\Pi_2^P = coNP^{NP}$ of problems that can be solved in non-deterministic polynomial time when the algorithm has access to an NP oracle; and PSPACE of problems that can be solved using only the polynomial space of memory and exponential time. We have $P \subseteq NP/coNP \subseteq \Delta_2^P \subseteq \Pi_2^P \subseteq PSPACE$. The standard decision problems studied for an AF F and a semantics σ are: (1) Credulous/sceptical acceptance $Cred_\sigma/Scept_\sigma$ (does a given argument appear in at least one extension?); and (2) Verification Ver_σ: (does a given extension appear in $\sigma(F)$?) In what follows we study the complexity of these problems with regards to the semantics we consider (see Table 2).

Completeness notions based on BBU weak admissibility. To the best of our knowledge, the only existing complexity results for the semantics under consideration are those of [14] which show that the semantics of Baumann, Brewka and Ulbricht [5] are PSPACE-complete. By carefully inspecting the reductions in [14], we obtain that the different notions of weakly complete semantics of Dauphin et al. [9] are also PSPACE-hard, which is not surprising since these semantics are defined on top of weak admissibility. Moreover, it is easy to verify that the different notions of defence in the semantics of Dauphin et al. can be tested within PSPACE and thus PSPACE-completeness follows.

Qualified Semantics. We now turn to the complexity of qualified semantics, which are similar to the original semantics and thus it is not surprising that the complexity is unchanged. That is, we can verify a **q-co** labelling in polynomial time by processing the SCCs in topological order and, for each argument, check whether its label is valid w.r.t. its neighbours (differentiating between outparents and non-outparents). To verify a **q-pr** labelling, we also verify maximality, which can be done in coNP by the standard algorithm. The upper bounds for the reasoning problems are then arrived at by standard guessing and checking algorithms, taking into account that the unique minimal **q-co** can be computed via fixed-point iteration. The hardness results are obtained due to (a) the hardness results for *co* and *pr* semantics hold for strongly connected graphs and (b) on strongly connected graphs, qualified semantics coincide with the base semantics.[1]

Proposition 1 *The complexity results for* ***q-co*** *and* ***q-pr*** *semantics in Table 2 hold.*

Semi-qualified semantics. For semi-qualified semantics, verifying a labelling gets harder. In order to test whether a labelling is **sq-co**, it is not sufficient to validate the labels of the arguments with respect to the labels of the other arguments in the same labelling scheme, but we have to consider also the other labellings. However, once we know which of the arguments in the earlier SCCs are credulously accepted (and thus which are also labelled $\mathtt{O}$ at least once), we can verify the label of an argument in polynomial time.

That is, for all reasoning tasks, we first process the SCCs in a topological ordering, and for each SCC, we decide which of the arguments are credulously accepted. That is, for each argument a, we nondeterministically guess a labelling that labels a as $\mathtt{I}$ and labels all the arguments that are not in S or preceding SCCs as $\mathtt{U}$. Such a labelling can then be verified in polynomial time, given that we already know which of the arguments in the preceding SCCs are credulously accepted. As we have to do this for each of the arguments, this part takes a linear number of NP oracle calls, i.e. we have a Δ_2^{P} algorithm for credulous acceptance. Given that we have computed all credulously accepted arguments, for **sq-co**, we can then run a polynomial time algorithm to verify a given labelling, while for **sq-pr**, we have to perform an additional maximality check which just requires an additional NP-oracle call. In total, this gives a Δ_2^{P} algorithm for the verification problem. The standard guessing and checking algorithm for sceptical acceptance now provides an Π_2^{P} algorithm for **sq-pr** while for **sq-co**, we can run a polynomial time fixed-point iteration to compute the unique minimal labelling, which results in a Δ_2^{P} algorithm.

The hardness of $Scept_{\textbf{sq-pr}}$ holds because the hardness results for *pr* semantics hold even for single-SCC AFs. Next, consider the Δ_2^{P}-hardness of $Cred_{\textbf{sq-pr}} = Cred_{\textbf{sq-co}}$. This is by a reduction from the Δ_2^{P}-complete problem of deciding whether for a propositional formula in CNF φ given by a set of clauses C over atoms $x_1,\dots,x_n$, the lexicographical maximum satisfying assignment of ϕ sets x_n to true. The reduction builds n SCCs, each corresponding to an adaptation of the standard translation from SAT to AFs, but which are in a linear order. The AF $G_\varphi = (A,R)$ is constructed as follows: $A = \{x_i^j \mid 1 \leq i \leq j \leq n\} \cup \{\bar{x}_i^j \mid 1 \leq i,j \leq n, i \neq j\} \cup \{c^j \mid 1 \leq j \leq n, c \in C\} \cup \{t^j, b^j \mid 1 \leq j \leq n\}$; and

$$
\begin{aligned}
R =& \{(x_i^j,\bar{x}_i^j),(\bar{x}_i^j,x_i^j) \mid x_i^j \in A\} \cup \{(x_i^j,c^j) \mid x_i^j \in A, x_i \in c \in C\} \cup \\
& \{(\bar{x}_i^j,c^j) \mid \bar{x}_i^j \in A, \neg x_i \in c \in C\} \cup \{(c^j,t^j),(t^j,b^j),(b^j,b^j) \mid 1 \leq j \leq n\} \cup \\
& \{(b^j,x_i^j) \mid x_i^j \in A\} \cup \{(b^j,\bar{x}_i^j) \mid \bar{x}_i^j \in A\} \cup \\
& \{(x_i^i,c^j) \mid x_i \in c \in C, 1 \leq i \leq j \leq n)\} \cup \{(x_i^i,\bar{x}_i^j) \mid 1 \leq i < j \leq n\}.
\end{aligned}
$$

Notice that the upper index of the arguments denote the SCC they belong to, and only the last line in the definition of the attack relation introduces attacks between SCCs.

[1] Note that these hardness proofs construct AFs with an empty grounded extension. Then, one can add a self-attacking argument g that symmetrically attacks all other arguments, without changing the complete extensions.

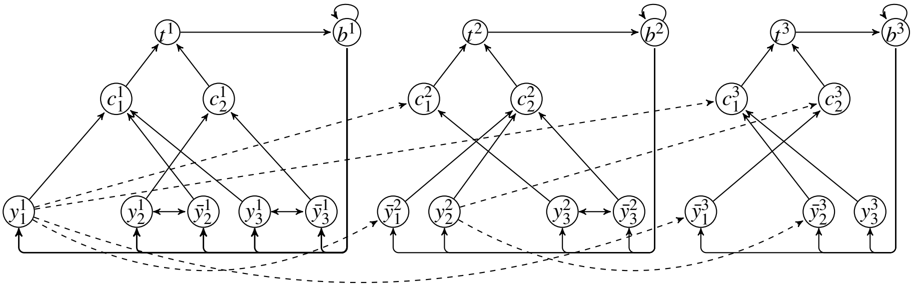


Figure 2. Illustration of the reduction G_φ for the formula φ with clauses $\{\{y_1,\neg y_2,y_3\},\{\neg y_1,y_2,\neg y_3)\}$. Attacks between different SCCs are highlighted as dashed lines

The intuition is as follows. In the first SCC, we test whether some assignment sets x_1 to true. If so, we fix this assignment by adding x_1^1 to the extension, which then attacks all arguments $\bar{x}_1^j$. If not, all arguments in the first SCC are labelled U, and we have to pick $\bar{x}_1^j$ (we may assume that C contains clauses $(x_i,\neg x_i)$ in the latter SCCs), we proceed like that, and in the j-th SCC, we check whether some assignment sets x_j to true given the already fixed assignment on earlier variables. One can show that x_n is true in the lexicographic-maximum-satisfying assignment of φ iff x_n^n is credulously accepted in G_φ.

Proposition 2 *The complexity results for* ***sq-co*** *and* ***sq-pr*** *semantics in Table 2 hold.*

UB Semantics. While **ub-**semantics requires that defended arguments are labelled I, they do not require that all I-labelled arguments are defended. It turns out that this lack of admissibility reduces the complexity significantly. First, notice that by definition in UB-complete labellings: (a) if an argument a is labelled I, then all arguments attacked by a must be labelled O; and (b) if all attackers of an argument a are labelled O, then a must be labelled I. We can compute UB-complete labellings by starting with a set S of I-labelled arguments, and then use the two rules from above to propagate labels until either we obtain that an argument must be both labelled I and O or a fixed point is reached. In the former cases, we have that there is no UB-complete labelling that labels all the arguments in S as I. In the latter case, we label the remaining arguments U to obtain UB-complete labelling. By that, we have that the grounded labelling is the unique minimal **ub-co** labelling, and thus sceptical acceptance is P-complete. To decide on credulous acceptance w.r.t. **ub-co** (and **ub-***pr*), one can fix the label of the query argument as I and apply the characteristic function until a fixed point is reached. If the result is conflict free, the argument is credulously accepted, otherwise it is not accepted. The conditions for a **ub-co** labelling can be easily checked in polynomial time, When verifying **ub-pr** labellings, we have to also check the maximality condition. Let S be the set of I-labelled arguments. We can now test for each U-labelled argument a whether $S\cup\{a\}$ is contained in some UB-complete labelling. This can be done simply by the above-mentioned fixed-point iteration and is thus in polynomial time. Given that verification is in P, we can solve $Scept_{\textbf{ub-pr}}$ with the standard guessing and checking approach in coNP. We next show that $Scept_{\textbf{ub-pr}}$ is also coNP-hard. To this end, consider the following adaptation of standard reduction (cf. Figure 2). Given a propositional formula φ in CNF given by a set of clauses C over atoms Y, we define φ as $F_\varphi = (A,R)$ (cf. Figure 2), where

$A = \{\varphi, \bar{\varphi}_1, \bar{\varphi}_2\} \cup C \cup Y \cup \bar{Y}$ and $R = \{(c, \bar{\varphi}_1) \mid c \in C\} \cup \{(l,c),(c,c) \mid l \in c, c \in C\} \cup \{(x,\bar{x}),(\bar{x},x) \mid x \in Y\} \cup \{(\bar{\varphi}_1, \varphi), (\varphi, \bar{\varphi}_2), (\bar{\varphi}_2, \bar{\varphi}_1)\}$. If all clause arguments c_i are labelled `O`, none of the arguments in the cycle can be accepted. Otherwise, if at least one c_i remains `U`, we can accept φ set $\bar{\varphi}_2$ as `O` and $\bar{\varphi}_1$ as `U`. We thus have that the argument φ is sceptically accepted iff formula φ is unsatisfiable.

Proposition 3 *The complexity results for* ***ub-co*** *and* ***ub-pr*** *semantics in Table 2 hold.*

UB2 semantics. We now turn to the SCC-recursive variants of the **ub-**semantics, **ub2-**semantics. In order to verify a labelling, we can follow the SCC-recursive schema and apply the verification of the base semantics in the base case. Since verification of **ub-**semantics is in polynomial time, we obtain that **ub2-**semantics can also be verified in polynomial time. The NP/coNP-membership of credulous/sceptical reasoning is then verified by the standard guessing and checking algorithms, and the matching hardness results are verified by standard reductions for credulous and sceptical acceptance.

Proposition 4 *The complexity results for* ***ub2-co*** *and* ***ub2-pr*** *semantics in Table 2 hold.*

UB and UB*2 semantics.* Similarly, for the UB-complete semantics, we can verify UB*-complete and UB*2-complete extensions in polynomial time, and thus the remaining upper bounds are obtained by standard procedures.

Now, consider the verification of an UB*-preferred extension. The additional condition that an `U`-labelled argument has to attack an `U`-labelled argument allows for a similar behaviour to that of standard Dung complete and preferred semantics. Consider an argument a that is labelled `I` and an argument b that attacks only argument a. We have that b cannot be labelled `U` or `I` and thus has to be labelled `O`. That is, we have to find an argument c that can be labelled `I` and defends a against b. With this observation, one can easily adapt the standard translations such that the NP/coNP hardness results for the reasoning tasks of complete and preferred semantics also transfer to UB* complete and UB* preferred semantics (even for strongly connected graphs).

Proposition 5 *The complexity results for* ***ub*-co****,* ***ub*-pr ub2*-co*** *and* ***ub2*-pr*** *semantics in Table 2 hold.*

7. Conclusion

We reviewed several recent proposals for non-admissible non-naive semantics for abstract argumentation and studied their complexity. We focused in particular on semantics that behave similar to the weakly complete and preferred semantics, but are based on undecidedness blocking mechanisms. Our complexity results (Table 2) show that this approach has significantly lower complexity than the weak admissibility based approach. We also defined a variant called *loop semantics*, that (1) assigns the label `U` only to arguments that are part of a loop, and (2) allows arguments labelled `I` to be attacked by `U`-labelled arguments. Unlike the weakly complete semantics, the complete variant of this semantics satisfies directionality. We plan to investigate the properties of this new semantics, as well as the derived notion of UB-admissibility, in future work.

Acknowledgements

This work was funded by the Austrian Science Fund (FWF) under grant Y698 and by the Vienna Science and Technology Fund (WWTF) under grant ICT19-065. Leendert van der Torre acknowledges financial support from the Fonds National de la Recherche Luxembourg (INTER/Mobility/19/13995684/DLAl/van der Torre).

References

[1] Pietro Baroni, Guido Boella, Federico Cerutti, Massimiliano Giacomin, Leendert van der Torre, and Serena Villata. On the input/output behavior of argumentation frameworks. *Artificial Intelligence*, 217:144–197, 2014.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Engineering Review*, 26(4):365–410, 2011.

[3] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1-2):162–210, 2005.

[4] Pietro Baroni, Massimiliano Giacomin, and Beishui Liao. Locality and modularity in abstract argumentation. In *Handbook of formal argumentation*, pages 937–979. London, UK: College Publications, 2018.

[5] Ringo Baumann, Gerhard Brewka, and Markus Ulbricht. Shedding new light on the foundations of abstract argumentation: Modularization and weak admissibility. *Artificial Intelligence*, 310:103742, 2022.

[6] Alexander Bochman. *Explanatory Nonmonotonic Reasoning*, volume 4 of *Advances in Logic*. World Scientific, 2005.

[7] Marcos Cramer and Leendert van der Torre. SCF2 - an argumentation semantics for rational human judgments on argument acceptability. In *Proceedings of the 8th Workshop on Dynamics of Knowledge and Belief (DKB-2019) and the 7th Workshop KI & Kognition (KIK-2019)*, pages 24–35, 2019.

[8] Jeremie Dauphin, Tjitze Rienstra, and Leendert van der Torre. A principle-based analysis of weakly admissible semantics. In *Proceedings of COMMA2020*, pages 167–178. IOS Press, 2020.

[9] Jérémie Dauphin, Tjitze Rienstra, and Leendert van der Torre. New weak admissibility semantics for abstract argumentation. In *Logic and Argumentation - Proceedings of CLAR2021*, volume 13040 of *Lecture Notes in Computer Science*, pages 112–126. Springer, 2021.

[10] Pierpaolo Dondio. Weakly-admissible semantics and the propagation of ambiguity in abstract argumentation semantics. Technical report, Technological University Dublin, 2019.

[11] Pierpaolo Dondio and Luca Longo. Weakly complete semantics based on undecidedness blocking. *arXiv preprint arXiv:2103.10701*, 2021.

[12] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[13] Wolfgang Dvořák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In *Handbook of Formal Argumentation*, pages 631–687. London, UK: College Publications, 2018.

[14] Wolfgang Dvořák, Markus Ulbricht, and Stefan Woltran. Recursion in abstract argumentation is hard - on the complexity of semantics based on weak admissibility. In *Proceedings of AAAI2021*, pages 6288–6295, 2021.

[15] Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, 14(5):675–702, 2004.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220148

Treewidth for Argumentation Frameworks with Collective Attacks

Wolfgang DVOŘÁK [a], Matthias KÖNIG [a] and Stefan WOLTRAN [a]
[a] *TU Wien, Institute of Logic and Computation*

Abstract. Abstract Argumentation is a key formalism to resolve conflicts in incomplete or inconsistent knowledge bases. Argumentation Frameworks (AFs) and extended versions thereof turned out to be a fruitful approach to reason in a flexible and intuitive setting. The addition of collective attacks, we refer to this class of frameworks as SETAFs, enriches the expressiveness and allows for compacter instantiations from knowledge bases, while maintaining the computational complexity of standard argumentation frameworks. This means, however, that standard reasoning tasks are intractable and worst-case runtimes for known standard algorithms can be exponential. In order to still obtain manageable runtimes, we exploit graph properties of these frameworks. In this paper, we initiate a parameterized complexity analysis of SETAFs in terms of the popular graph parameter treewidth. While treewidth is well studied in the context of AFs with their graph structure, it cannot be directly applied to the (directed) hypergraphs representing SETAFs. We thus introduce two generalizations of treewidth based on different graphs that can be associated with SETAFs, i.e., the primal graph and the incidence graph. We show that while some of these notions allow for parameterized tractability results, reasoning remains intractable for other notions, even if we fix the parameter to a small constant.

Keywords. Abstract Argumentation, Collective Attacks, SETAF, Treewidth

1. Introduction

Argumentation is a key area in Artificial Intelligence. Abstract Argumentation as introduced by Dung [1] serves as a unifying framework to capture argumentation processes in a formal yet intuitive setting. In standard argumentation frameworks (AFs), discussions are formalized as a directed graph where the nodes represent abstract arguments (independent of their internal structure), and the edges represent the attack relation. It turned out that the binary attack relation of AFs occasionally limits the expressiveness of frameworks, in particular if one is not willing to introduce artificial arguments to model technicalities. To avoid this issue, Nielsen and Parsons proposed a relaxation of this restriction: collective attacks [2]. If a set S collectively attacks an argument, said argument is only effectively defeated by S if all arguments in S are accepted by an agent. The resulting class of frameworks is referred to as SETAFs. It was shown that SETAFs are indeed more expressive than AFs [3].

Due to SETAFs being highly expressive yet intuitive, there is now an increased interest in this formalism within the research community. While in the general case the same (mostly intractable) complexity upper bounds hold [4], tractable graph classes have only recently been investigated [5,6]. We add to this by starting the analysis of computational properties of SETAFs in the context of *parameterized complexity*. A problem is *fixed-parameter tractable* (FPT), if we can find a numerical parameter p describing the instances such that for constant values of p the runtime is polynomial in the instance size (and the degree of the polynomial is independent of p). Implementations of these parameterized algorithms often work well in practice, if instances are not randomly generated but admit an exploitable structure. One prominent such parameter is *treewidth*. A low treewidth indicates a certain "tree-likeness" of a graph, and as problems often become easy on trees, adapted versions of these easy algorithms can often be applied to instancees with low treewidth. In AFs, it has been shown that reasoning is indeed fixed-parameter tractable w.r.t. treewidth [7]. We investigate how this notion of treewidth is applicable to the *directed hypergraph*-structure of SETAFs and show that certain generalizations admit FPT algorithms, while other reasonable attempts do not. In particular, our contributions can be summarized as follows. After recalling the necessary formal background in Section 2, we discuss the challenges of defining treewidth for SETAFs in Section 3 and present two different notions of treewidth: primal-treewidth and incidence-treewidth. In Section 4 we present negative results regarding primal-treewidth, namely that reasoning remains intractable even for small parameter values. Section 5 establishes FPT results for incidence-treewidth via a generic argument utilizing Monadic Second Order logic; this theoretical result gets refined and improved in Section 6 where we discuss a dynamic programming algorithm tailored to SETAFs. Finally, in Section 7 we conclude and give pointers to possibly interesting directions for future research.

2. Background

We start with the definition of an Argumentation Framework with Collective Attacks (SETAF) [2] as a generalization of (standard) Argumentation Frameworks (AFs) [1].

Definition 1. *A SETAF is a pair $SF = (A,R)$ where A is a finite set of* arguments*, and $R \subseteq (2^A \setminus \{\emptyset\}) \times A$ is the* attack relation. *For an attack $(T,h) \in R$ we call T the* tail *and h the* head *of the attack. SETAFs (A,R), where for all $(T,h) \in R$ it holds that $|T| = 1$, amount to (standard Dung) AFs. In that case, we usually write (t,h) to denote the set-attack $(\{t\},h)$. We write $S \mapsto_R a$ if there is a set $T \subseteq S$ with $(T,a) \in R$. Moreover, we write $S' \mapsto_R S$ if $S' \mapsto_R a$ for some $a \in S$. We drop subscript R in $\mapsto_R$ if there is no ambiguity. For $S \subseteq A$, we use S^+_R to denote the set $\{a \mid S \mapsto_R a\}$ and define the* range *of S (w.r.t. R), denoted $S^\oplus_R$, as the set $S \cup S^+_R$.*

The well-known notions of conflict and defense from classical Dung-style-AFs naturally generalize to SETAFs.

Definition 2. *Let $SF = (A,R)$ be a SETAF. A set $S \subseteq A$ is* conflicting *in SF if $S \mapsto_R a$ for some $a \in S$. $S \subseteq A$ is* conflict-free *in SF, if S is not conflicting in SF, i.e. if $T \cup \{h\} \not\subseteq S$ for each $(T,h) \in R$. $cf(SF)$ denotes the set of all conflict-free sets in SF.*

Definition 3. *Let $SF = (A,R)$ be a SETAF. An argument $a \in A$ is* defended *(in SF) by a set $S \subseteq A$ if for each $B \subseteq A$, such that $B \mapsto_R a$, also $S \mapsto_R B$. A set $T \subseteq A$ is defended (in SF) by S if each $a \in T$ is defended by S (in SF).*

Table 1. Complexity for AFs and SETAFs (C-c denotes completeness for C).

	grd	*adm*	*com*	*pref*	*stb*
$Cred_\sigma$	P-c	NP-c	NP-c	NP-c	NP-c
$Skept_\sigma$	P-c	trivial	P-c	Π_2^{P}-c	coNP-c

The semantics we study in this work are the grounded, admissible, complete, preferred, and stable, semantics, which we will abbreviate by *grd*, *adm*, *com*, *pref*, and *stb*, respectively [2,4,8]. Acceptable sets of arguments w.r.t. a semantics are called *extensions*.

Definition 4. *Given a SETAF $SF = (A,R)$ and a conflict-free set $S \in cf(SF)$. Then,*

- *$S \in adm(SF)$, if S defends itself in SF,*
- *$S \in com(SF)$, if $S \in adm(SF)$ and $a \in S$ for all $a \in A$ defended by S,*
- *$S \in grd(SF)$, if $S = \bigcap_{T \in com(SF)} T$,*
- *$S \in pref(SF)$, if $S \in adm(SF)$ and there is no $T \in adm(SF)$ s.t. $T \supset S$,*
- *$S \in stb(SF)$, if $S \mapsto a$ for all $a \in A \setminus S$,*

The relationship between the semantics has been clarified in [2,4,8] and matches with the relations between the semantics for Dung AFs, i.e. for any SETAF *SF*:

$$stb(SF) \subseteq pref(SF) \subseteq com(SF) \subseteq adm(SF) \subseteq cf(SF)$$

Complexity. We assume the reader to have basic knowledge in computational complexity theory[1], in particular we make use of the complexity classes P (polynomial time), NP (non-deterministic polynomial time), coNP, and Π_2^{P}. For a SETAF $SF = (A,R)$ and an argument $a \in A$, we consider the standard reasoning problems (under semantics σ):

- Credulous acceptance $Cred_\sigma$: Is a contained in at least one σ extension of SF?
- Skeptical acceptance $Skept_\sigma$: Is a contained in all σ extensions of SF?

The complexity landscape of SETAFs coincides with that of Dung AFs and is depicted in Table 1. As SETAFs generalize Dung AFs the hardness results for Dung AFs [9] carry over to SETAFs, also the same upper bounds hold for SETAFs [4].

For a more fine-grained complexity analysis we also make use of the complexity class FPT (fixed-parameter tractability): a problem is fixed-parameter tractable w.r.t. a parameter if there is an algorithm with runtime $O(f(p) \cdot n^k)$, where n is the size of the input, k is an integer constant, p is an integer describing the instance called the *parameter value*, and $f(\cdot)$ is an arbitrary computable function independent of n (typically at least exponential in p). For fixed (i.e., constant) parameter values p, FPT-runtime is polynomial (and the degree of the polynomial does not depend on p).

3. Graph Notions and Tree Decompositions of SETAFs

In this section we discuss approaches to apply the notion of *treewidth* [10] to SETAFs.

Definition 5 (Treewidth)**.** *Let $G = (V,E)$ be an undirected graph. A* tree decomposition (TD) *of G is a pair $(\mathcal{T},\mathcal{X})$, where $\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T})$ is a tree and $\mathcal{X} = (X_n)_{n \in V_\mathcal{T}}$ is a set of bags (a bag is a subset of V) such that*

[1] For a gentle introduction to complexity theory in the context of formal argumentation, see [9].

1. $\bigcup_{n \in V_{\mathcal{T}}} X_n = V$;
2. *for each* $v \in V$*, the subgraph induced by* v *in* $\mathcal{T}$ *is connected; and*
3. *for each* $\{v, w\} \in E$*,* $\{v, w\} \subseteq X_n$ *for some* $n \in V_{\mathcal{T}}$*.*

The width *of a TD is* $\max\{|X_n| \mid n \in V_{\mathcal{T}}\} - 1$*, the* treewidth *of G is the minimum width of all TDs for G.*

For fixed k it can be decided in linear time whether a graph has treewidth at most k; moreover an according tree decomposition can be computed in linear time [11]. For practical applications there are heuristic approaches available that will return decompositions of reasonable width very efficiently [12]. However, as the underlying structure of SETAFs is a *directed hypergraph*, this notion is not directly applicable in our context. We can use "standard" directed graphs to describe SETAFs, and apply treewidth by simply ignoring the direction of the involved arcs. For SETAFs there is the *primal graph* [5] and the *incidence graph* [6] as such notions, each of which leads to its own treewidth notion for SETAFs. First, we utilize the primal graph to define *primal-treewidth*.

Definition 6 (Primal Graph)**.** *Let* $SF = (A, R)$ *be a SETAF. The* primal graph *of SF is defined as* $\texttt{Primal}(SF) = (A, R')$*, where* $R' = \{(t, h) \mid (T, h) \in R, t \in T\}$*. The* primal-treewidth $\texttt{ptw}(SF)$ *is defined as the treewidth of* $\texttt{Primal}(SF)$*.*

It is easy to see that several SETAFs can map to the same primal graph. However, restrictions on the primal graph are often useful to obtain computational speedups for otherwise hard problems [5,6]. In contrast, the *incidence graph* uniquely describes a SETAF, as attacks are explicitly modeled in this notion. Again, we utilize the incidence graph to define *incidence-treewidth*.

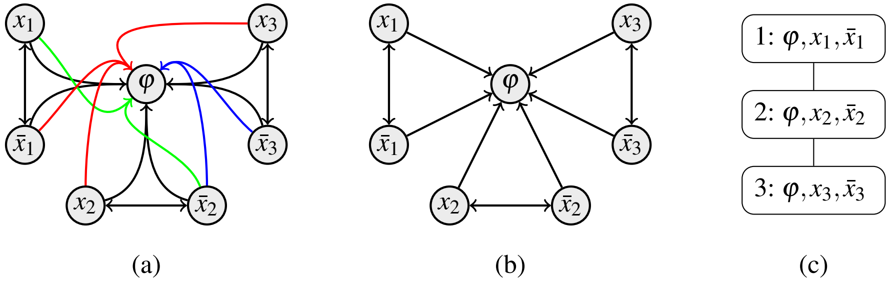
Definition 7 (Incidence Graph)**.** *Let* $SF = (A, R)$ *be a SETAF and let* $tails(SF) = \{T \mid (T, h) \in R\}$*. We define the bipartite* incidence graph *of SF as* $\texttt{Inc}(SF) = (V, E)$ *with* $V = A \cup tails(SF)$ *and* $E = \{(t, T), (T, h) \mid (T, h) \in R, t \in T\}$*. The* incidence-treewidth $\texttt{itw}(SF)$ *is defined as the treewidth of* $\texttt{Inc}(SF)$*.*

We want to highlight that (a) both of these notions properly generalize the classical notion of treewidth commonly applied to Dung-style AFs, and (b) these measures coincide on AFs. Formally:

Proposition 8. *The "standard" treewidth of AFs F coincides with* $\texttt{ptw}(F)$ *and* $\texttt{itw}(F)$*.*

Proof. The case of primal-treewidth is immediate. For incidence-treewidth note that we can construct $\texttt{Inc}(F)$ from F by substituting each edge $r = (a, b) \in R$ by a fresh vertex r and two edges (a, r), (r, b). It is well known that this operation preserves treewidth. □

We will first show that reasoning on SETAFs with fixed primal-treewidth remains hard (Section 4). Incidence-treewidth on the other hand admits FPT algorithms—we will initially establish this by characterizing the SETAF semantics via Monadic Second Order logic (MSO) [13,14] (Section 5). We utilize this characterization to obtain the desired upper bounds, as in this context we can apply a meta-theorem due to Courcelle [15,16]. In a nutshell, it states that every graph property that can be characterized in MSO can be checked in polynomial time. However, this generic method typically produces infeasible constants in practice, which is why in Section 6 we will refine these results and provide a prototypical algorithm with feasible constants for stable semantics (cf. [7]).


x_1
x_3
φ
$\bar{x}_1$
$\bar{x}_3$
x_2
$\bar{x}_2$
1: $\varphi, x_1, \bar{x}_1$
2: $\varphi, x_2, \bar{x}_2$
3: $\varphi, x_3, \bar{x}_3$

(a) (b) (c)

Figure 1. (a) The framework SF_φ from the proof of Theorem 9 for $\varphi = (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_2 \vee x_3)$, (b) $\texttt{Primal}(SF_\varphi)$, and (c) a tree-decomposition of $\texttt{Primal}(SF_\varphi)$ of width 2.

4. Decomposing the Primal Graph

We start with an investigation of the treewidth of the primal graph. It has been shown that various restrictions on the primal graph can lead to computational ease [5,6]. However, we will see that reasoning remains hard for SETAFs with constant primal-treewidth (in contrast to the AF-case, where we observe FPT results). We establish this via reductions from (QBF-)SAT, illustrated in Figures 1 and 2. Intuitively, the attacks between the dual literals x and $\bar{x}$ represent the choice between assigning x to *true* or *false*. The collective attack $(\{x,\bar{x}\},\varphi)$ ensures that we take at least one of x and $\bar{x}$ into any extension in order to defend φ, making sure we only construct proper truth assignments. Finally, the remaining attacks towards φ correspond to the clauses and make sure that we cannot defend φ if for any clause we set all duals of its literals *true*—as this means the clause is not satisfied.

Theorem 9. *The problems $Cred_\sigma$ for $\sigma \in \{adm, com, stb, pref\}$ are* NP*-complete, and $Skept_{stb}$ is* coNP*-complete for SETAFs SF with* $\texttt{ptw}(SF) \geq 2$.

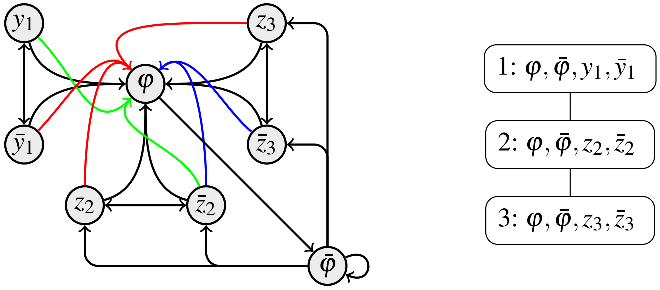
Proof. The membership coincides with the general case. For the respective hardness results, consider the following reduction from SAT (see Figure 1). Let X be the set of atoms and C be the set of clauses of a boolean formula φ (given in CNF). We denote a clause $c \in C$ as the set of literals in the clause, e.g. the clause $x_1 \vee \bar{x}_2 \vee \bar{x}_3$ correspond to the set of literals (arguments, resp.) $\{x_1, \bar{x}_2, \bar{x}_3\}$. By x^d we denote the dual of a literal (e.g. $x^d = \bar{x}$ and $\bar{x}^d = x$). Let $SF_\varphi = (A,R)$, where $A = \{x, \bar{x} \mid x \in X\} \cup \{\varphi\}$, and

$$R = \{(\{x^d \mid x \in c\}, \varphi) \mid c \in C\} \cup \{(\{x,\bar{x}\},\varphi),(x,\bar{x}),(\bar{x},x) \mid x \in X\}$$

Now it holds that φ is in at least one σ extension for $\sigma \in \{adm, com, stb, pref\}$ if and only if φ is satisfiable.

($\Rightarrow$) An admissible set E containing φ contains exactly one of each pair $x,\bar{x}$, as otherwise φ would not be defended against the attack $(\{x,\bar{x}\},\varphi)$. Moreover, as $\bar{c} \not\subseteq E$ for each attack $(\bar{c},\varphi)$—$\bar{c}$ consists of the *duals* of the literals in c—this means at least one argument corresponding to a literal of each clause $c \in C$ is in E. Hence, E corresponds to a satisfying assignment of φ.

($\Leftarrow$) Every satisfying assignment of φ corresponds to a stable extension of SF_φ: let I be the interpretation satisfying φ, the corresponding set $E = \{\varphi\} \cup \{x \mid I(x) = true\} \cup \{\bar{x} \mid I(x) = false\}$ is then stable (admissible, complete, preferred): As I satisfies φ, for each attack corresponding to a clause not all tail-arguments are in E, and hence φ is defended.


y_1
z_3
φ
$\bar{y}_1$
$\bar{z}_3$
z_2
$\bar{z}_2$
$\bar{\varphi}$
1: $\varphi, \bar{\varphi}, y_1, \bar{y}_1$
2: $\varphi, \bar{\varphi}, z_2, \bar{z}_2$
3: $\varphi, \bar{\varphi}, z_3, \bar{z}_3$

(a)

(b)

Figure 2. (a) SF_Φ from the proof of Theorem 10 for $\Phi = \forall\{y_1\}\exists\{z_2, z_3\}(y_1 \vee \bar{z}_2 \vee \bar{z}_3) \wedge (\bar{y}_1 \vee z_2) \wedge (z_2 \vee z_3)$, and (b) a tree-decomposition of $\texttt{Primal}(SF_\Phi)$ of width 3.

For the coNP completeness result, we add an argument $\bar{\varphi}$ and an attack $(\varphi, \bar{\varphi})$. If φ is unsatisfiable, φ will be attacked and $\bar{\varphi}$ will be contained in every stable extension. Finally note that stable extensions are admissible, complete, and preferred. The constant primal-treewidth is immediate, as illustrated in the example of Figure 1(c). □

Also for preferred semantics reasoning remains intractable for SETAFs with fixed primal-treewidth. For this result, we extend the construction from Theorem 9 by an additional argument $\bar{\varphi}$ that attacks the existentially quantified variables (see Figure 2).

Theorem 10. *$Skept_{pref}$ is Π_2^P-complete for SETAFs SF with* $\texttt{ptw}(SF) \geq 3$.

Proof. We show this by a reduction from the Π_2^{P}-complete $QBF_\forall^2$ problem. Let $\Phi = \forall Y \exists Z \varphi$ be a $QBF_\forall^2$-formula with φ in CNF. We construct the SETAF SF_Φ by extending F_φ from Theorem 9 in the following way (for an example see Figure 2): First, we set $X = Y \cup Z$ and add all arguments and attacks according to the construction of F_φ. Moreover we add an argument $\bar{\varphi}$ and attacks $(\bar{\varphi}, \bar{\varphi}), (\varphi, \bar{\varphi})$. The last step is to add attacks $(\bar{\varphi}, z), (\bar{\varphi}, \bar{z})$ for each $z \in Z$. Now φ is in every preferred extension of SF_Φ if and only if Φ is valid. We start with some general observations: $\bar{\varphi}$ cannot be in any admissible set, and φ can only be in an admissible set S if for each $x \in Y \cup Z$ exactly one of x and $\bar{x}$ is in S. Moreover, in order to have $S \cap (Z \cup \bar{Z}) \neq \emptyset$, the argument $\bar{\varphi}$ has to be attacked by S, and consequently $\varphi \in S$. This means for every admissible set S that $S \cap (Y \cup \bar{Y} \cup Z \cup \bar{Z})$ corresponds to a satisfying assignment of the formula φ. In summary, every assignment on the variables Y corresponds to an admissible set, and every other admissible set in SF_Φ contains φ and represents a satisfying assignment for φ.

($\Rightarrow$) Assume φ is in every preferred extension. Since every set $S \subseteq 2^Y$ is admissible and in order to accept φ for each $x \in Y \cup Z$ either x or $\bar{x}$ have to be accepted, we know that for every assignment of Y variables there is an assignment satisfying φ.

($\Leftarrow$) Now assume Φ is valid, i.e. for each assignment I_Y on the variables Y, there is an assignment I_Z on Z such that $I_Y \cup I_Z$ satisfies φ. From this and our observations above it follows that φ is in every preferred extensions.

It is easy to see that the primal-treewidth of F_Φ is bounded by 3 (see Figure 2(b)). □

Hence, under standard complexity-theoretical assumptions, these problems do not become tractable when parameterized by the primal-treewidth.

5. Parameterized Tractability via Incidence-Treewidth

In this section, we establish tractability for reasoning in SETAFs with constant incidence treewidth by utilizing a meta-theorem due to Courcelle [15,16]. In particular, we use the tools of Monadic Second Order logic (MSO) to characterize the semantics of SETAFs (similarly, this has been done for AFs [13,14]). MSO generalizes first-order logic in the sense that it is also allowed to quantify over sets. Domain elements in our settings are vertices of an (incidence)-graph, i.e., arguments or attacks. Hence, MSO in our context consists of *variables* corresponding to domain elements (indicated by lower case letters), and *set-variables* corresponding to sets of domain elements (indicated by uppercase letters). Moreover, we use the standard logical connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, as well as quantifiers $\exists, \forall$ for both types of variables. We use the unary predicates $A(\cdot)$ and $R(\cdot)$ to indicate an element being an argument or an attack of our SETAF, respectively. Moreover, we use the binary predicate $E(x,y)$ to indicate an edge in the incidence graph between incidence-vertices x and y. Alternatively, we write $a \in A$, $r \in R$, $(x,y) \in E$ for $A(a), R(r), E(x,y)$, respectively. Based on these basic definitions, we define notational shortcuts to conveniently characterize SETAF-properties. Let $SF{=}(A,R)$ be a SETAF and $\mathtt{Inc}(SF){=}(V,E)$ its incidence graph. We define the following notion for $T \subseteq V$ and $h \in V$: let $(T,h) \in R$ be short-hand notation for $\exists r\,(r \in R \wedge (r,h) \in E \wedge \forall t(t \in T \leftrightarrow (t,r) \in E))$. This notion consists of four parts: (1) vertex r corresponds to an attack, (2) h is the head of the attack r, (3) the arguments in T constitute the tail of r. We utilize this to avoid dealing with the attack-vertices of the incidence graph in our semantics characterizations. We borrow the following "building blocks" from [14] (slightly adapted for our setting).

$$\begin{array}{ll}
X \subseteq Y = \forall x\,(x \in X \rightarrow x \in Y) & X \not\subset Y = \neg(X \subset Y) \\
X \subset Y = X \subseteq Y \wedge \neg(Y \subseteq X) & x \notin X = \neg(x \in X) \\
X \not\subseteq Y = \neg(X \subseteq Y) & x \in X_R^{\oplus} = x \in X \vee \exists Y(Y \subseteq X \wedge (Y,x) \in R)
\end{array}$$

We can express (subset-)maximality: $\max_{A,P(.),\subseteq}(X) = P(X) \wedge \neg\exists Y(Y \subseteq A \wedge P(Y) \wedge X \subset Y)$, and analogously (subset-)minimality: $\min_{A,P(.),\subseteq}(X) = \max_{A,P(.),\supseteq}(X)$ [14] for any expressible property $P(\cdot)$. Having these tools at hand, we can characterize the SETAF semantics in an intuitive way. It is easy to verify that these exactly correspond to the respective notions from Definition 4. Utilizing these building blocks, we can encode the semantics *cf*, *adm*, *com*, *grd*, *stb*, *pref* exactly as in AFs [13,14].

Definition 11. *Let $SF = (A,R)$ be a SETAF and let $\mathtt{Inc}(SF) = (V,E)$ be its incidence graph. For a set $X \subseteq V$ where $\forall x \in X(x \in A)$:*

$$\begin{aligned}
cf(X) &= \forall T,h\,((T,h) \in R \rightarrow (T \not\subseteq X \vee h \notin X)) \\
adm(X) &= cf(X) \wedge \forall T,h(((T,h) \in R \wedge h \in X) \rightarrow \exists S,t(S \subseteq X \wedge t \in T \wedge (S,t) \in R)) \\
com(X) &= adm(X) \wedge \forall x((x \in A \wedge x \notin X) \rightarrow \\
&\qquad\qquad \exists S((S,x) \in R \wedge \neg\exists T(T \subseteq X \wedge (X,s) \in R \wedge s \in S))) \\
grd(X) &= \min_{A,com(\cdot),\subseteq}(X) \\
stb(X) &= cf(X) \wedge \forall x(x \in A \rightarrow x \in X_R^{\oplus}) \\
pref(X) &= \max_{A,adm(\cdot),\subseteq}(X)
\end{aligned}$$

We can immediately apply Courcelle's theorem [15,16] to obtain the desired result.

Theorem 12. *Let SF be a SETAF. For the semantics under our consideration, reasoning is fixed-parameter tractable w.r.t.* $\mathtt{itw}(SF)$.

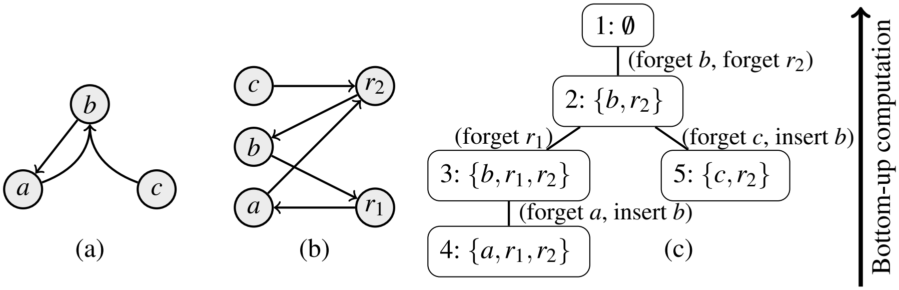


Figure 3. Running example for Section 6: (a) SETAF SF; (b) $\texttt{Inc}(SF)$; (c) tree decomposition of $\texttt{Inc}(SF)$. The edge labels indicate how (c) can be transformed into a nice tree decomposition.

6. Dynamic Programming on SETAFs

In the following, we specify a dynamic programming algorithm utilizing incidence-treewidth to reason in stable semantics. Ultimately, we will show that this algorithm allows us to reason efficiently in SETAFs with fixed incidence-treewidth.

Node Types. To illustrate the idea of this algorithm, we restrict the tree-decompositions of the incidence graph to *nice tree-decompositions*: a tree-decomposition $(\mathscr{T},\mathscr{X})$ is called *nice* if $\mathscr{T}=(V_\mathscr{T},E_\mathscr{T})$ is a rooted tree with an empty bag in the root node, and if each node $t\in\mathscr{T}$ (shorthand notion for $n\in V_\mathscr{T}$) is of one of the following types:

1. *Leaf*: n has no children in $\mathscr{T}$,
2. *Forget*: n has one child n', and $X_n = X_{n'}\setminus\{v\}$ for some $v\in X_{n'}$,
3. *Insert*: n has one child n', and $X_n = X_{n'}\cup\{v\}$ for some $v\notin X_{n'}$,
4. *Join*: n has two children n', n'', with $X_n = X_{n'} = X_{n''}$.

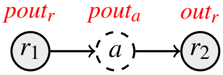
Any tree-decomposition can be transformed into a nice tree decomposition with the same width in linear time [17]. Let $SF=(A,R)$ be a SETAF and $\texttt{Inc}(SF)=(V,E)$. For sets $U\subseteq V$, by U^A, U^R we identify the sets $(U\cap A)$, $(U\cap R)$, respectively. By $X_{\geq n}$ we denote the union of all bags X_m where $m\in V_\mathscr{T}$ appears in the subtree of $\mathscr{T}$ rooted in n.

Colorings. We use *colors* to keep track of the arguments and attacks that appear in the bag X_n of node $n\in V_\mathscr{T}$. These colorings characterize extension candidates that are consistent with the framework rooted in the node in question. For an argument a in relation to an extension E, we use the color in_a to indicate $a\in E$. The color out_a indicates there is an attack $r=(T,a)$ with $a\notin T, T\subseteq E$, and $r\in X^R_{\geq n}$, i.e., a is attacked by E and a "responsible" attack appears in the subtree of $\mathscr{T}$ rooted in n. Finally, the color $pout_a$ (*provisionally* out) indicates there is an attack $r=(T,a)$ with $a\notin T, T\subseteq E$, and $r\notin X^R_{\geq n}$, i.e., a is attacked by E but the "responsible" attack appears "above" the node n in $\mathscr{T}$. Similarly, for attacks (T,h) we use the color in_r to indicate $T\subseteq E$. The color out_r means that there is an argument $a\in T$ (i.e., in the tail) that is attacked by E, and the "responsible" attack appears in the subtree of $\mathscr{T}$ rooted in n. Finally, $pout_r$ means that an argument $a\in T$ is attacked by E, but the "responsible" attack appears "above" the node n in $\mathscr{T}$. Formally, a *coloring* for a node $n\in V_\mathscr{T}$ is a function $C: X_n\to\{in_a, out_a, pout_a, in_r, out_r, pout_r\}$. By $[C]$ we denote the set $\{a\mid C(a)=in_a\}$. Colorings in a node $t\in\mathscr{T}$ characterize extension candidates—partial evaluations of the framework rooted in t. Colorings are generated in the leaves, and unsuitable extension candidates are successively eliminated when traversing the tree in a bottom-up manner. For an example see Figure 3.

(a)

r_1	a	r_2
in_r	out_a	out_r
$pout_r$	in_a	in_r
$pout_r$	in_a	$pout_r$
$pout_r$	$pout_a$	out_r

(b)



(c)

r_1	r_2
in_r	out_r
$pout_r$	in_r
$pout_r$	$pout_r$

Figure 4. Example for valid colorings for (a) the leaf node 4 from Figure 3 and (c) the preceding forget node for argument a. Subfigure (b) illustrates the subgraph of the incidence graph that corresponds to the leaf node together with the coloring that is discarded by the forget node.

Leaf Nodes. Intuitively, in leaf nodes we guess one of two possibilities: *in* or *out*/*pout* for each argument and each attack, and keep every "consistent" coloring. Whether an argument/attack is colored *out* or *pout* depends only on whether the attack in this coloring is already present in the current leaf node. Formally, a *valid coloring* for a leaf n is each coloring that satisfies the conditions in the box below. The valid colorings for leaf node 4 of our running example (Figure 3) are depicted in Figure 4(a).

For each argument $a \in X_n^A$:

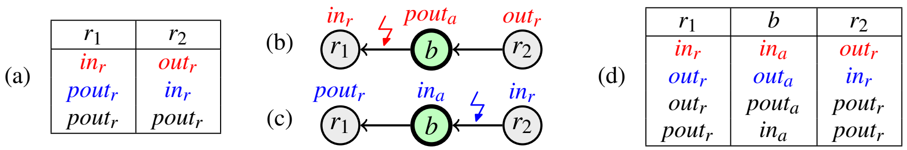
$$C(a) = in_a \Rightarrow \forall r = (T,a) \in X_n^R\colon C(r) \in \{pout_r, out_r\}$$
$$C(a) = out_a \Leftrightarrow \exists r = (T,a) \in X_n^R\colon C(r) = in_r$$

For every attack $r = (T,h) \in X_n^R$:

$$C(r) = in_r \Rightarrow C(h) \neq in_a \wedge \forall t \in T \cap X_n^A\colon C(t) = in_a$$
$$C(r) = out_r \Leftrightarrow \exists t \in T \cap X_n^A\colon C(t) \in \{pout_a, out_a\}$$

Forget Nodes. We examine forget-argument nodes and forget-attack nodes separately. Let n be a *forget-argument node* with child n' such that $X_n^A = X_{n'}^A \setminus \{a\}$. We have to discard all colorings C where $C(a) = pout_a$, as in these colorings a is supposed to be attacked by $[C]$. As we forget a in the current node and by the definition of a tree-decomposition, this cannot happen: consider again the running example from Figure 3. a is forgotten between bag 4 and 3; i.e., in the "upper" part of the tree decomposition, no attacks towards a can be added. Hence, there cannot be an attack colored in_r towards a that confirms a being attacked, and the provisional color $pout_a$ cannot be updated to out_a. Formally, if C is a valid coloring for n' and $C(a) \neq pout_a$, then $C - a$ is a valid coloring for n, where $(C-a)(b) = C(b)$ for each $b \in X_n$. We handle *forget-attack nodes* in the same way: if n is a forget node with child n' such that $X_n^R = X_{n'}^R \setminus \{r\}$, and if $C(r) \neq pout_r$, then $(C-r)$ is a valid coloring for n, where $(C-r)(b) = C(b)$ for each $b \in X_n$.

Insert Nodes. We distinguish the two cases where we insert an argument and insert an attack. Whenever we insert an argument a, we have to consider up to two different scenarios: $(C+a)$: the added argument is attacked by the extension. In this case the added argument is colored $pout_a$ or out_a, depending on whether the "responsible" attack is already in the current bag. In case a is in the tail of an attack, we can color this attack out_r. $(C \dot{+} a)$: the added argument is in the extension. In both cases we have to check whether the result will be consistent with the existing colors, i.e., for $(C+a)$ the added argument must not be in the tail of an attack that is colored in_r, and for $(C \dot{+} a)$ there must not be an attack colored in_r towards the added argument. Assume we would color the inserted argument b as $out_a/pout_a$ while b is in the tail of attack r, which we already colored in_r in a previous step. Of course, this is not consistent with our intended meaning of the attack color in_r (see (Figure 5(b)). On the other hand, assume we color b as in_a while it is attacked by r which we already colored in_r in a previous step. This would



(a)

r_1	r_2
in_r	out_r
$pout_r$	in_r
$pout_r$	$pout_r$

(d)

r_1	b	r_2
in_r	in_a	out_r
out_r	out_a	in_r
out_r	$pout_a$	$pout_r$
$pout_r$	in_a	$pout_r$

Figure 5. "Insert b" node between node 4 and 3 (in the running example from Figure 3 after "Forget a" from Figure 4). Subfigures (b) and (c) show inconsistent colorings, (d) shows the resulting valid colorings.

introduce a conflict in the constructed extension (see (Figure 5(c)). The operations $C+a$ and $C \dotplus a$ are defined the box below. Formally: Let n be an *insert-argument node* with child n' s.t. $X_n^A = X_{n'}^A \cup \{a\}$. If C is a valid coloring for n',

- if $\nexists r = (T,h) \in X_n^R$: $(C(r) = in_r \wedge a \in T)$, then $C+a$ is a valid coloring for n;
- if $\nexists r = (T,a) \in X_n^R$: $(C(r) = in_r)$, then $C \dotplus a$ is a valid coloring for n.

$$(C+a)(b) = \begin{cases} out_a & \text{if } b = a \wedge \exists r = (T,a) \in X_n^R\colon (C(r) = in_r \wedge a \notin T) \\ pout_a & \text{if } b = a \wedge \nexists r = (T,a) \in X_n^R\colon (C(r) = in_r \wedge a \notin T) \\ out_r & \text{if } b = (T,h) \wedge a \in T \wedge C(b) = pout_r \\ C(b) & \text{otherwise} \end{cases}$$

$$(C \dotplus a)(b) = \begin{cases} in_a & \text{if } b = a \\ C(b) & \text{otherwise} \end{cases}$$

For *insert-attack nodes*, we also have to consider two cases for an attack $r = (T,h)$: $(C+r)$: the extension attacks T, either in the current bag (in which case we color the attack out_r), or possibly in the "upper" parts of $\mathscr{T}$, then we color the attack $pout_r$. $(C \dotplus r)$: this case indicates $T \subseteq E$ for the extension E. In this case the head of the attack can be set to out_a. Again, we can only apply this coloring if it is consistent with the previous colors. We will use the operations $C+r$ and $C \dotplus r$ as defined below. Let n be an *insert-attack node* with child n' such that $X_n^R = X_{n'}^R \cup \{r = (T,h)\}$. If C is a valid coloring for n',

- then $C+r$ is a valid coloring for n;
- if $(h \notin X_n^A \vee C(h) \neq in_a) \wedge \forall t \in T \cap X_t^A : C(t) = in_a$, then $C \dotplus r$ is a valid coloring for n.

$$(C+r)(b) = \begin{cases} out_r & \text{if } b = r \wedge \exists t \in T \cap X_n^A\colon C(t) \in \{pout_a, out_a\} \\ pout_r & \text{if } b = r \wedge \nexists t \in T \cap X_n^A\colon C(t) \in \{pout_a, out_a\} \\ C(b) & \text{otherwise} \end{cases}$$

$$(C \dotplus r)(b) = \begin{cases} in_r & \text{if } b = r \\ out_a & \text{if } r = (T,h) \wedge b = h \\ C(b) & \text{otherwise} \end{cases}$$

Join Nodes. In these nodes we combine the colorings of immediate child nodes. Let n be a join node with children n', n''. If C is a valid coloring for n' and D is a valid coloring for n'' with $[C] = [D]$ and $\{r \mid C(r) = in_r\} = \{r \mid D(r) = in_r\}$, then $C \bowtie D$ is a valid coloring for n (see the box below).

$$(C \bowtie D)(b) = \begin{cases} in_a & \text{if } C(b) = D(b) = in_a \\ out_a & \text{if } C(b) = out_a \vee D(b) = out_a \\ pout_a & \text{otherwise (if } b \in X^A) \\ in_r & \text{if } C(b) = D(b) = in_r \\ out_r & \text{if } C(b) = out_r \vee D(b) = out_r \\ pout_r & \text{otherwise (if } b \in X^R) \end{cases}$$

Theorem 13. *With the presented algorithm, $Cred_{stb}$, $Skept_{stb}$ as well as counting the number of stable extensions can be done in time $O(5^k \cdot k \cdot (|A| + |R|))$. Moreover, we can enumerate all stable extensions with linear delay.*

Proof. We can assume the number of nodes to be bounded by $O(|A| + |R|)$. For each node, the number of (valid) colorings (i.e., rows in our tables of colorings) is bounded by 3^k and that we can find and access rows in linear time w.r.t. k. In leaf nodes, we can check the colorings in time $O(k^2)$ for each of the $O(3^k)$ possible colorings, resulting in $O(3^k \cdot k^2)$. In forget nodes, we can check the conditions and compute eventually resulting colorings in time $O(k)$ for each of the $O(3^k)$ colorings of the child node, resulting in $O(3^k \cdot k)$. In insert nodes, we can check the conditions and compute eventually resulting colorings in time $O(k^2)$ for each of the $O(3^k)$ colorings of the child node, resulting in $O(3^k \cdot k^2)$. Finally, for join nodes we have to consider $3^k \cdot 3^k = 9^k$ pairs. However, we only need to consider 5^k pairs if we assume the data structure to be properly sorted, e.g. lexicographically by treating the colors in_a/in_r as 0 and $pout_a/out_a/pout_r/out_r$ as 1. As each table has $O(3^k)$ rows, sorting is in $O(3^k \cdot k)$. Let C be a coloring such that $m \leq k$ arguments/attacks are colored as in_a/in_r. There exist at most 2^{k-m} distinct colorings C' with $\forall x\colon (C(x) \in \{in_a, in_r\} \Leftrightarrow C'(x) \in \{in_a, in_r\})$. There are $\binom{k}{m}$ possibilities resulting from the choice of m, resulting in $\sum_{m=0}^{k} \binom{k}{m} \cdot 2^{k-m} \cdot 2^{k-m} = 5^k$ join pairs. We can then compute $C \bowtie D$ in $O(k)$, resulting in $O(5^k \cdot k)$ for join nodes, dominating the runtime of the other node types. The resulting runtime for the algorithm is $O(5^k \cdot k \cdot (|A| + |R|))$.

We can decide $Cred_{stb}/Skept_{stb}$ for $a \in A$ by flagging colorings that contain/do not contain a. In each node we update the flag accordingly; the flag in the root node indicates credulous/skeptical acceptance. We can keep the count of extensions w.r.t. each coloring, and to enumerate the extensions once the dynamic programming algorithm is done we can traverse the tree top-down and output the extensions with linear delay (cf. [7]). □

Other Semantics. The core concepts to characterize stable extensions carry over to other admissibility-based semantics, where also undecidedness can occur. This can be handled in a similar manner as the $pout_r/out_r$ colors, where one indicates "confirmed undecidedness" and another color indicates "provisional undecidedness". The latter color can be "updated" to the former if a suitable witness is present (either in an insert- or join node). Again, colorings containing provisional colors have to be removed in forget nodes.

7. Discussion

In this paper, we investigated the treewidth parameter for reasoning tasks in SETAFs. We showed that reasoning with constant primal-treewidth remains hard (contrasting the results for the special case of AFs), while constant incidence-treewidth allows us to reason and count in polynomial time. Finally, we improved these generically obtained results by providing a dynamic programming algorithm tailored for SETAFs, highlighting interesting differences to the AF-case that arise from the generalization step. The underlying structure of SETAFs is a directed hypergraph. While there are measures available for general hypergraphs, the directed case is not as well explored—this work contributes to this, as we provide an alternative treewidth measure in this context. Moreover, while there are several systems available to compute the treewidth of undirected simple graphs efficiently—be it exactly or heuristically—the situation for implementations of hypertreewidth is less advanced. Finally, reasoning in frameworks with fixed *directed* graph parameters (e.g., cycle rank, directed path-width, etc.) already turned out to be intractable for AFs [7]; which carries over to SETAFs. Hence, we decided to focus on the treewidth-based measures, so that we can implement the presented algorithms in the future.

The results of this paper may serve as a starting point for further parameterized analysis of computational properties of SETAFs. Considering SETAFs in recent additions to the treewidth literature in the context of argumentation constitutes interesting topics for future research, see e.g. [18]. For example, recently treewidth has been investigated in conjunction with *backdoors* in [19], effectively decreasing the relevant parameter value.

Acknowledgments. This research has been supported by the Vienna Science and Technology Fund (WWTF) through project ICT19-065, and by the Austrian Science Fund (FWF) through projects P32830 and Y698.

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artif Intell. 1995;77(2):321-58.

[2] Nielsen SH, Parsons S. A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In: Proceedings of ArgMAS 2006. Springer; 2006. p. 54-73.

[3] Dvořák W, Fandinno J, Woltran S. On the expressive power of collective attacks. Argument Comput. 2019;10(2):191-230.

[4] Dvořák W, Greßler A, Woltran S. Evaluating SETAFs via Answer-Set Programming. In: Proceedings of SAFA 2018. vol. 2171 of CEUR Workshop Proceedings. CEUR-WS.org; 2018. p. 10-21.

[5] Dvořák W, König M, Woltran S. Graph-Classes of Argumentation Frameworks with Collective Attacks. In: Proceedings of JELIA 2021. vol. 12678 of LNCS. Springer; 2021. p. 3-17.

[6] Dvořák W, König M, Woltran S. On the Complexity of Preferred Semantics in Argumentation Frameworks with Bounded Cycle Length. In: Proceedings of KR 2021; 2021. p. 671-5.

[7] Dvořák W, Pichler R, Woltran S. Towards fixed-parameter tractable algorithms for abstract argumentation. Artif Intell. 2012;186:1-37.

[8] Flouris G, Bikakis A. A comprehensive study of argumentation frameworks with sets of attacking arguments. Int J Approx Reason. 2019;109:55-86.

[9] Dvořák W, Dunne PE. Computational Problems in Formal Argumentation and their Complexity. In: Handbook of Formal Argumentation. College Publications; 2018. p. 631-87.

[10] Robertson N, Seymour PD. Graph minors. II. Algorithmic aspects of tree-width. J Algorithms. 1986;7(3):309-22.

[11] Bodlaender HL. A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth. SIAM J Comput. 1996;25(6):1305-17.

[12] Abseher M, Musliu N, Woltran S. htd - A Free, Open-Source Framework for (Customized) Tree Decompositions and Beyond. In: Proceedings of CPAIOR 2017. Springer; 2017. p. 376-86.

[13] Dunne PE. Computational properties of argument systems satisfying graph-theoretic constraints. Artif Intell. 2007;171(10-15):701-29.

[14] Dvořák W, Szeider S, Woltran S. Abstract Argumentation via Monadic Second Order Logic. In: Proceedings of SUM 2012. vol. 7520 of LNCS. Springer; 2012. p. 85-98.

[15] Courcelle B. Recognizability and second-order definability for sets of finite graphs. Université de Bordeaux; 1987. I-8634.

[16] Courcelle B. Graph rewriting: an algebraic and logic approach. In: Handbook of theoretical computer science, Vol. B. Amsterdam: Elsevier; 1990. p. 193-242.

[17] Kloks T. Treewidth, Computations and Approximations. vol. 842 of LNCS. Springer; 1994.

[18] Fichte JK, Hecher M, Mahmood Y, Meier A. Decomposition-Guided Reductions for Argumentation and Treewidth. In: Zhou Z, editor. Proceedings of IJCAI 2021; 2021. p. 1880-6.

[19] Dvořák W, Hecher M, König M, Schidler A, Szeider S, Woltran S. Tractable Abstract Argumentation via Backdoor-Treewidth. In: Proceedings of AAAI 2022; 2022. p. 5608-15.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220149

Rule-PSAT: Relaxing Rule Constraints in Probabilistic Assumption-Based Argumentation

Xiuyi Fan
School of Computer Science and Engineering
Nanyang Technological University
Singapore

Abstract. Probabilistic rules are at the core of probabilistic structured argumentation. With a language $\mathcal{L}$, probabilistic rules describe conditional probabilities $\Pr(\sigma_0|\sigma_1,\ldots,\sigma_k)$ of deducing some sentences $\sigma_0 \in \mathcal{L}$ from others $\sigma_1,\ldots,\sigma_k \in \mathcal{L}$ by means of prescribing rules $\sigma_0 \leftarrow \sigma_1,\ldots,\sigma_k$ with head σ_0 and body $\sigma_1,\ldots,\sigma_k$. In Probabilistic Assumption-based Argumentation (PABA), a few constraints are imposed on the form of probabilistic rules. Namely, (1) probabilistic rules in a PABA framework must be acyclic, and (2) if two rules have the same head, then the body of one rule must be the subset of the other. In this work, we show that both constraints can be relaxed by introducing the concept of *Rule Probabilistic Satisfiability (Rule-PSAT)* and solving the underlying joint probability distribution on all sentences in $\mathcal{L}$. A linear programming approach is presented for solving Rule-PSAT and computing sentence probabilities from joint probability distributions.

Keywords. Probabilistic Argumentation, Probabilistic Satisfiability

1. Introduction

Probabilistic Assumption-based Argumentation (PABA) [1] provides a probabilistic extension to the Assumption-based Argumentation (ABA) framework [2] by allowing probabilistic rules in argument construction. As a form of probabilistic structured argumentation (along with p-ASPIC [3] and probabilistic argumentation with logic [4,5]) PABA was shown to admit as instances several other probabilistic argumentation approaches and with an implementation engine developed [6], and complexity results studied in [7].

A few design choices have been made in PABA to ensure its semantics and inference approaches sound. Namely:

1. if there are two rules with the same head having different probabilities, then the body of one rule must be the subset of the other (Definition 2.1, [1]);
2. there is no infinite path starting from a probabilistic parameter in its dependency graph in a PABA framework (Lemma 2.1, [1]).

Constraint 1 specifies that a probability sentence can only be deduced from at most one set of antecedents; whereas Constraint 2 specifies that paths leading to probability sentences must be acyclic. In this work, we show that both constraints can be relaxed by considering *Probabilistic Satisfiability* [8]. We see that the two constraints given by [1] are design choices to ensure probabilistic satisfiability. However, as we illustrate in this work, without these two constraints, there are cases where probabilistic satisfiability can still hold with well-understood inference processes available. In other words, we show that there is no intrinsic reason to disallow multiple rules with the same heads and cyclic graphs when constructing probabilistic extensions to ABA. Thus, this work provides a generalisation to the probabilistic rules given in PABA with a sound inference method for sentence probability calculation.

The rest of this paper is organised as the follows. Section 2 reviews two concepts introduced in the literature that are needed in this work. Section 3 introduces of *Rule-PSAT* that describe probabilistic consistency. Section 4 presents an inference approach for reasoning Rule-PSAT. Section 5 compares our work with Nilsson's probabilistic logic / satisfiability in detail. We conclude in Section 6.

2. Background

In this work, we need two notions, *deduction for a sentence*, and *complete conjunction set* of a language introduced in the literature as follows.

Given a language $\mathcal{L}$, and a set of rules $\mathcal{R}$ built with sentences in $\mathcal{L}$, a *deduction* [2] for $\sigma \in \mathcal{L}$ supported by $S \subseteq \mathcal{L}$ and $R \subseteq \mathcal{R}$ denoted $S \vdash^R \sigma$ is a finite tree with

- nodes labelled by sentences in $\mathcal{L}$ or by a special symbol τ that is not in $\mathcal{L}$,
- the root labelled by σ,
- leaves either labelled by τ or sentences in S,
- non-leaves labelled by σ' with, as children, the elements of the body of some rule in $\mathcal{R}$ with head σ', and R the set of all such rules.

Deduction is a fundamental concept in rule-based systems. We will refer to it in Section 3.

Given a language $\mathcal{L}$ with n sentences, the *Complete Conjunction Set (CC Set)* [9] of $\mathcal{L}$ is the set of 2^n conjunction of sentences such that each conjunction contains n distinct sentences. For instance, for $\mathcal{L} = \{\sigma_0, \sigma_1\}$, the CC set of $\mathcal{L} = \{\neg\sigma_0 \wedge \neg\sigma_1, \neg\sigma_0 \wedge \sigma_1, \sigma_0 \wedge \neg\sigma_1, \sigma_0 \wedge \sigma_1\}$. As we will discuss in the next section, a CC set defines the universe of all possible worlds given by the language.

3. Rule-PSAT

We start by introducing the core representation of this work, namely the notion of a *probabilistic rule (p-rule)*, as follows.

Definition 1. Given a language $\mathcal{L}$, a *probabilistic rule (p-rule)* is

$$\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k : [\theta]$$

for $k \geq 0, \sigma_i \in \mathcal{L}, 0 \leq \theta \leq 1$. σ_0 is referred to as the *head* of the p-rule, $\sigma_1, \ldots, \sigma_k$ the *body*, and θ the *probability.*

Given a language $\mathcal{L}$ and a set of p-rules $\mathcal{R}$, we say that $\mathcal{R}$ is *defined over* $\mathcal{L}$ iff all sentences in p-rules in $\mathcal{R}$ are in $\mathcal{L}$.

The rule in Definition 1 states that the probability of σ_0, when $\sigma_1 \ldots \sigma_k$ all hold, is θ. In other words, this rule states that $\Pr(\sigma_0 | \sigma_1, \ldots, \sigma_k) = \theta$. Note that this is the same interpretation of probabilistic rules introduced in [1].

To introduce Rule-PSAT, we need to consider the set of sentences that are "deducible". This is constructed with the notion of deduction as follows.[1]

Definition 2. Given a language $\mathcal{L}$ and a set of p-rules $\mathcal{R}$ defined over $\mathcal{L}$, the *deducible set* $\mathcal{L}_0 = \{\sigma \in \mathcal{L} | \emptyset \vdash^R \sigma$, where $R \subseteq \mathcal{R}\}$.

The *deducible rules* $\mathcal{R}_0 = \{\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k : [\cdot] \in \mathcal{R} | \sigma_i \in \mathcal{L}_0, i = 0 \ldots k\}$.

We illustrate deducible set and rules in Example 1.

Example 1. Let $\mathcal{L} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$, $\mathcal{R} = \{\sigma_0 \leftarrow \sigma_1 : [\alpha]; \sigma_1 \leftarrow: [\beta]; \sigma_2 \leftarrow \sigma_3 : [\gamma]\}$. We have $\mathcal{L}_0 = \{\sigma_0, \sigma_1\}$ and $R_0 = \{\sigma_0 \leftarrow \sigma_1 : [\alpha]; \sigma_1 \leftarrow: [\beta]\}$.

Definition 3. Given a language $\mathcal{L}$ and a set of p-rules $\mathcal{R}$, let Ω be the CC set of $\mathcal{L}_0$. A function $\pi : \Omega \to [0, 1]$ is a *consistent probability distribution* with respect to $\mathcal{R}$ on $\mathcal{L}$ for Ω iff:[2]

1. For all $\omega_i \in \Omega$,

$$0 \leq \pi(\omega_i) \leq 1. \tag{1}$$

2. It holds that:

$$\sum_{\omega_i \in \Omega} \pi(\omega_i) = 1. \tag{2}$$

3. For each p-rule $\sigma_0 \leftarrow: [\theta] \in \mathcal{R}_0$, it holds that:

$$\sum_{\omega_i \in \Omega, \omega_i \models \sigma_0} \pi(\omega_i) = \theta. \tag{3}$$

4. For each p-rule $\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k : [\theta] \in \mathcal{R}_0$, $(k > 0)$, it holds that:

$$\frac{\sum_{\omega_i \in \Omega, \omega_i \models \sigma_0 \wedge \ldots \wedge \sigma_k} \pi(\omega_i)}{\sum_{\omega_i \in \Omega, \omega_i \models \sigma_1 \wedge \ldots \wedge \sigma_k} \pi(\omega_i)} = \theta. \tag{4}$$

[1] We use the notion of "deduction" with the symbol "$\vdash^R$" introduced in Section 2 without modification by treating p-rules as non-probabilistic rules in this context.

[2] In this work, symbols $\neg$, $\wedge$, and $\models$ take their standard meaning as in classical logic.

Our notion of consistency as given in Definition 3 consists of two parts. Equations 1 and 2 assert π being a probability distribution over the CC set of $\mathcal{L}_0$; whereas equations 3 and 4 assert that each p-rule should be viewed as defining conditional probabilities for which the probability of the head of the p-rule conditioned on the body is the probability. In particular, Equation 3 can be viewed as a special case of 4 as when the body is empty, the head is conditioned on the universe. In other words, for p-rule $\sigma_0 \leftarrow: [\theta]$, we assert $\Pr(\sigma_0) = \theta$ with Equation 3; for $\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k : [\theta]$, we assert $\Pr(\sigma_0|\sigma_1, \ldots, \sigma_k) = \theta$ with Equation 4.

Example 2. (Example 1 continued.) $\Omega = \{\neg\sigma_0 \wedge \neg\sigma_1, \sigma_0 \wedge \neg\sigma_1, \neg\sigma_0 \wedge \sigma_1, \sigma_0 \wedge \sigma_1\}$.
From $\sigma_0 \leftarrow \sigma_1 : [\alpha]$, we have

$$\frac{\pi(\sigma_0 \wedge \sigma_1)}{\pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \sigma_1)} = \alpha. \tag{5}$$

From $\sigma_1 \leftarrow: [\beta]$, we have

$$\pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \sigma_1) = \beta. \tag{6}$$

π is a consistent probability distribution iff Equations 5 and 6 hold, as well as

$$\pi(\neg\sigma_0 \wedge \neg\sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \sigma_1) = 1, \tag{7}$$

and

$$0 \leq \pi(\neg\sigma_0 \wedge \neg\sigma_1), \pi(\sigma_0 \wedge \neg\sigma_1), \pi(\neg\sigma_0 \wedge \sigma_1), \pi(\sigma_0 \wedge \sigma_1) \leq 1. \tag{8}$$

With consistency defined, we are ready to define Rule-PSAT as follows.

Definition 4. The *Rule Probabilistic Satisfiability (Rule-PSAT)* problem is to determine for a set of p-rules $\mathcal{R}$ on a language $\mathcal{L}$, whether there exists a consistent probability distribution for the CC set of $\mathcal{L}_0$ with respect to $\mathcal{R}$.

Example 3. (Example 2 continued.) To test whether $\mathcal{R}$ is Rule-PSAT on $\mathcal{L}$, we need to solve Equations 5-8 for π as $\mathcal{R}$ is Rule-PSAT iff a solution exists. It is easy to see that this is the case as:

$$\begin{aligned}
\pi(\sigma_0 \wedge \sigma_1) &= \alpha\beta \\
\pi(\neg\sigma_0 \wedge \sigma_1) &= \beta - \alpha\beta \\
\pi(\sigma_0 \wedge \neg\sigma_1) + \pi(\neg\sigma_0 \wedge \neg\sigma_1) &= 1 - \beta
\end{aligned}$$

Since $0 \leq \alpha, \beta \leq 1$, we have $0 \leq \pi(\sigma_0 \wedge \sigma_1), \pi(\neg\sigma_0 \wedge \sigma_1) \leq 1$. We can let $\pi(\sigma_0 \wedge \neg\sigma_1) = 0$, $\pi(\neg\sigma_0 \wedge \neg\sigma_1) = 1 - \beta$ and obtain one solution for π. As the system is under-specified, we have infinitely many solutions to $\pi(\sigma_0 \wedge \neg\sigma_1)$ and $\pi(\neg\sigma_0 \wedge \neg\sigma_1)$ in the range of $[0,\, 1 - \beta]$.

The next example gives a p-rule set that is not Rule-PSAT.

Example 4. Let $\mathcal{R}$ contain three p-rules: $\sigma_0 \leftarrow \sigma_1 : [0.9], \sigma_0 \leftarrow: [0.8], \sigma_1 \leftarrow: [0.9]$. From $\sigma_0 \leftarrow \sigma_1 : [0.9]$ and Equation 4, we have

$$\frac{\pi(\sigma_0 \wedge \sigma_1)}{\pi(\sigma_0 \wedge \sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1)} = 0.9. \tag{9}$$

From $\sigma_1 \leftarrow: [0.9]$, we have

$$\pi(\sigma_0 \wedge \sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1) = 0.9. \tag{10}$$

Substitute (10) in (9), we have $\pi(\sigma_0 \wedge \sigma_1) = 0.81$.

From $\sigma_0 \leftarrow: [0.8]$, we have

$$\pi(\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1) = 0.8.$$

Thus, $\pi(\sigma_0 \wedge \neg\sigma_1) = -0.01$, which does not satisfy $0 \leq \pi(\omega_i) \leq 1$.

From a Rule-PSAT solution, which characterises a probability distribution over the CC set, one can compute sentence probabilities by marginalising over sentences. In other words, we can compute the probability of sentences by summing up $\pi(\omega_i)$.

Given a language $\mathcal{L}$ and a set of p-rules $\mathcal{R}$, if there is a consistent probability distribution π for Ω, the CC set of $\mathcal{L}_0$, with respect to $\mathcal{R}$, then for any $\sigma \in \mathcal{L}_0$, its probability $\Pr(\sigma)$ is:

$$\Pr(\sigma) = \sum_{\omega_i \in \Omega, \omega_i \models \sigma} \pi(\omega_i). \tag{11}$$

Clearly, from Definition 3, we see that $0 \leq \Pr(\sigma) \leq 1$ and $\Pr(\sigma) + \Pr(\neg\sigma) = 1$. (Note that although Equation 11 is similiar to 3, 11 refers to all sentences $\sigma \in \mathcal{L}_0$, whereas 3 refers to sentences that are heads of rules with an empty body.)

Example 5. (Example 3 continued.) Taking π shown in Example 3, (with $\pi(\sigma_0 \wedge \neg\sigma_1) = 0$) the probabilities of σ_0 and σ_1 can be computed as follows.

$$\begin{aligned} \Pr(\sigma_0) &= \pi(\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1) = \alpha\beta \\ \Pr(\sigma_1) &= \pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \sigma_1) = \beta \end{aligned}$$

If a set of p-rules $\mathcal{R}$ is satisfiable, then the range of the probability of any sentence in $\mathcal{L}_0$ can be found with mathematical optimisation. The upper and the lower bounds of the probability of a sentence $\sigma \in \mathcal{L}_0$ can be found by maximising and minimising the RHS of Equation 11 subject to Equations 1-4, respectively.

Example 6. (Example 5 continued.) To compute the upper and lower bounds of $\Pr(\sigma_0)$, we maximise and minimise $\Pr(\sigma_0) = \pi(\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1)$, respectively. We see that $\Pr(\sigma_0)$ is at its max when $\pi(\sigma_0 \wedge \neg\sigma_1)$ is. Since $0 \leq \pi(\sigma_0 \wedge \neg\sigma_1) \leq 1 - \beta$, we have the upper bound of $\Pr(\sigma_0)$ taking its value $\alpha\beta + 1 - \beta$. Similarly, $\Pr(\sigma_0)$ takes its min value when $\pi(\sigma_0 \wedge \neg\sigma_1) = 0$. Thus, the lower bound of $\Pr(\sigma_0)$ is $\alpha\beta$.

Note that there is no restriction imposed on the form of p-rules other than the ones given in Definition 1, as illustrated in the next two examples (Examples 7 and 8) - a set of p-rules can be consistent even if there are rules in this set forming cycles or having two rules with the same head.

Example 7. Consider a set of p-rules $\mathcal{R} = \{\sigma_0 \leftarrow \sigma_1 : [0.7], \sigma_1 \leftarrow \sigma_0 : [0.6], \sigma_1 \leftarrow: [0.5]\}$. We can see that there are infinitely many different (finite) deductions for both σ_0 and σ_1 due to the cycle formed by *deduce* σ_0 *from* σ_1 and *deduce* σ_1 *from* σ_0. However, we can still compute a (unique) solution for π over the CC set of $\{\sigma_0, \sigma_1\}$. Using Equations 2 to 4, we have:

$$\begin{aligned}
0.7 &= \pi(\sigma_0 \wedge \sigma_1)/(\pi(\sigma_0 \wedge \sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1)),\\
0.6 &= \pi(\sigma_0 \wedge \sigma_1)/(\pi(\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1)),\\
0.5 &= \pi(\sigma_0 \wedge \sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1),\\
1 &= \pi(\neg\sigma_0 \wedge \neg\sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1) + \pi(\sigma_0 \wedge \sigma_1).
\end{aligned}$$

Solutions found are: $\pi(\neg\sigma_0 \wedge \neg\sigma_1) = 0.27$, $\pi(\sigma_0 \wedge \neg\sigma_1) = 0.23$, $\pi(\neg\sigma_0 \wedge \sigma_1) = 0.15$, $\pi(\sigma_0 \wedge \sigma_1) = 0.35$.

Example 8. Consider a set of p-rules $\mathcal{R} = \{\sigma_0 \leftarrow \sigma_1 : [0.6], \sigma_0 \leftarrow \sigma_2 : [0.5], \sigma_1 \leftarrow: [0.7], \sigma_2 \leftarrow: [0.6]\}$, There are two p-rules with head σ_0. They have different bodies and probabilities. We set up equations as follows.[3]

$$\begin{aligned}
0.6 &= (\pi(111) + \pi(110))/(\pi(010) + \pi(011) + \pi(110) + \pi(111)),\\
0.5 &= (\pi(101) + \pi(111))/(\pi(001) + \pi(011) + \pi(101) + \pi(111)),\\
0.7 &= \pi(010) + \pi(011) + \pi(110) + \pi(111),\\
0.6 &= \pi(001) + \pi(011) + \pi(101) + \pi(111),\\
1 &= \pi(000) + \pi(001) + \pi(010) + \pi(011) + \pi(100) + \pi(101) + \pi(110) + \pi(111).
\end{aligned}$$

Solve these, a solution maximising $\Pr(\sigma_0)$ is follows: $(\Pr(\sigma_0) = 0.7)$

$$\begin{array}{llll}
\pi(000) = 0, & \pi(001) = 0.02, & \pi(010) = 0, & \pi(011) = 0.28,\\
\pi(100) = 0.15, & \pi(101) = 0.13, & \pi(110) = 0.25, & \pi(111) = 0.17.
\end{array}$$

A solution minimising $\Pr(\sigma_0)$ is: $(\Pr(\sigma_0) = 0.42)$

$$\begin{array}{llll}
\pi(000) = 0.14, & \pi(001) = 0.16, & \pi(010) = 0.14, & \pi(011) = 0.14,\\
\pi(100) = 0, & \pi(101) = 0, & \pi(110) = 0.12, & \pi(111) = 0.3.
\end{array}$$

[3] To simplify the presentation, Boolean values are used as shorthand for the sentences. E.g., 111, 011, and 001 denote $\sigma_0 \wedge \sigma_1 \wedge \sigma_2$, $\neg\sigma_0 \wedge \sigma_1 \wedge \sigma_2$, and $\neg\sigma_0 \wedge \neg\sigma_1 \wedge \sigma_2$, respectively.

4. Solve Rule-PSAT

Given a set of p-rules $\mathcal{R} = \{\rho_1, \dots, \rho_m\}$ constructed on some language $\mathcal{L}$ such that $\mathcal{L}_0$ contains n sentences, to test whether $\mathcal{R}$ is Rule-PSAT, we set up a linear system

$$A\Pi = B, \tag{12}$$

where A is an $(m+1) \times 2^n$ matrix, $\Pi = [\pi(\omega_1), \dots, \pi(\omega_{2^n})]^T$, B an $(m+1) \times 1$ matrix.[4] We construct A and B in a way such that R is Rule-PSAT iff Π has a solution in $[0,1]^{2^n}$, as follows.

For each rule $\rho_i \in \mathcal{R}$, if $\rho_i = \sigma_0 \leftarrow: [\theta]$ has an empty body, then

$$A[i,j] = \begin{cases} 1, & \text{if } \omega_j \models \sigma_0; \\ 0, & \text{otherwise;} \end{cases} \tag{13}$$

and

$$B[i] = \theta. \tag{14}$$

Otherwise, $\rho_i = \sigma_0 \leftarrow \sigma_1, \dots, \sigma_k : [\theta]$, then

$$A[i,j] = \begin{cases} \theta - 1, & \text{if } \omega_j \models \sigma_0 \wedge \sigma_1 \wedge \dots \wedge \sigma_k; \\ \theta, & \text{if } \omega_j \models \neg\sigma_0 \wedge \sigma_1 \wedge \dots \wedge \sigma_k; \\ 0, & \text{otherwise;} \end{cases} \tag{15}$$

and

$$B[i] = 0. \tag{16}$$

Row $m+1$ in A and B are $1 \dots 1$ and 1, respectively.

Example 9. (Example 6 continued.) Let $\rho_0 = \sigma_0 \leftarrow \sigma_1 : [\alpha]$, $\rho_1 = \sigma_1 \leftarrow: [\beta]$. Here, $m = 2$, $n = 2$. From Equations 12 to 16, we have

$$A = \begin{bmatrix} 0 & \alpha & 0 & \alpha - 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$\Pi = [\pi(\neg\sigma_0 \wedge \neg\sigma_1), \pi(\neg\sigma_0 \wedge \sigma_1), \pi(\sigma_0 \wedge \neg\sigma_1), \pi(\sigma_0 \wedge \sigma_1)]^T$, and $B = [0, \beta, 1]^T$. It is easy to see that Π has solutions as shown in Example 3.

Theorem 1. Given a set of p-rules $\mathcal{R}$ on some language $\mathcal{L}$, $\mathcal{R}$ is Rule-PSAT iff Equation 12 has a solution for Π in $[0,1]^{2^n}$.

[4] We let $\{\omega_1, \dots, \omega_{2^n}\}$ be the CC set of $\mathcal{L}_0$. We consider elements in this set being ordered with their Boolean values. E.g., for $\mathcal{L}_0 = \{\sigma_0, \sigma_1\}$, the four elements in the CC set are ordered such that $\{\omega_1 = \neg\sigma_0 \wedge \neg\sigma_1, \omega_2 = \neg\sigma_0 \wedge \sigma_1, \omega_3 = \sigma_0 \wedge \neg\sigma_1, \omega_4 = \sigma_0 \wedge \sigma_1\}$.

Table 1.: Performance Demonstration for Solving Rule-PSAT with the Python *scipy linprog* Library with an Interior-point Method.

Number of Sentences	6	7	8	9	10	11	12
Run Time (s)	0.014	0.022	0.041	0.11	0.55	3.32	22.38

Proof. (Sketch.) Equations 1 to 4 are satisfied by a Π solution in $[0,1]^{2^n}$ as follows.

1. If $\Pi \in [0,1]^{2^n}$, then $0 \leq \pi(\omega_i) \leq 1$ for all ω_i.
2. Since Row $m+1$ in A and B are 1s, we have the sum of all $\pi(\omega_i)$ being 1.
3. For each p-rule $\sigma_0 \leftarrow:[\theta]$, Equations 13 and 14 ensure that Equation 3 is satisfied.
4. For each p-rule $\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k:[\theta]$, Equation 15 and 16 ensure that Equation 4 is satisfied with simple algebra.

Thus, we see that Equation 12, $A\Pi = B$, is nothing but a linear system representation of Equations 1-4, which characterise probability distributions over the CC set of $\mathcal{L}_0$ with conditionals. □

Table 1 demonstrates the performance of a Python implementation of the linear system approach for solving Rule-PSAT introduced in this section. The implementation is built with the open source *scipy linprog* library[5], using an interior-point method. We observe that the average running time grows exponentially as the number of sentences in a set of p-rules. This is expected as the size of the CC set is 2^n (n the number of sentences in $\mathcal{L}_0$); and interior-point method has a super-linear complexity [10]. P-rules in this experiment are randomly generated with maximum length of rule body 4, the average time of 10 runs for each configuration is reported.

5. Discussion

Many works have been published on probabilistic argumentation in recent years, e.g. [11,12,13,14,15,16,17,4]. With very few exceptions, notably [4,5], existing works are predominantly defined with abstract argumentation, having probability distributions defined over argumentation graphs. In [4,5], arguments are constructed with probabilistic logic. As probabilistic rules are also used to construct arguments, we compare our work with probabilistic logic.

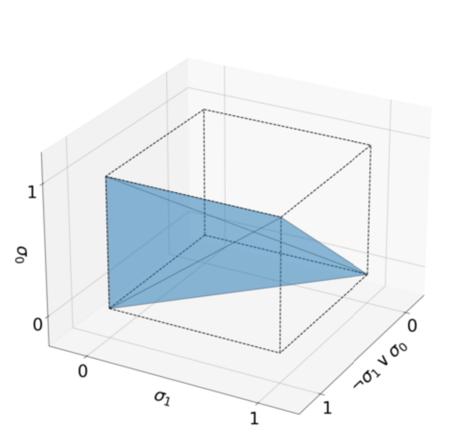
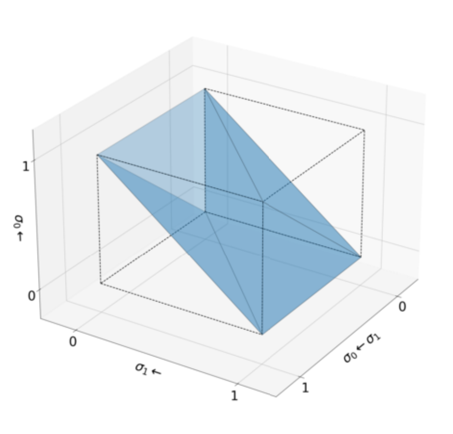
Nilsson [8] introduces Probabilistic Satisfiability with probabilistic logic, considering knowledge bases in Conjunctive Normal Form. A modus ponens example,[6]

$$\text{If } \sigma_1, \text{ then } \sigma_0.\ \sigma_1.\ \text{Therefore, } \sigma_0.$$

[5] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html

[6] This example is used in [8]. The figure on the left hand side of Table 2 is a reproduction of Figure 2 in [8].

Table 2.: Comparison of Consistent Probability Regions between Nilsson's Probabilistic Logic and Probabilistic Rules on an modus ponens instance.

Probabilistic Logic	Probabilistic Rule
$\neg\sigma_1 \vee \sigma_0 : [\alpha], \quad \sigma_1 : [\beta], \quad \sigma_0 : [\gamma].$	$\sigma_0 \leftarrow \sigma_1 : [\alpha],\ \sigma_1 \leftarrow: [\beta],\ \sigma_0 \leftarrow: [\gamma].$
	

is shown in Table 2. The probabilities of the conditional claim is α, the antecedent β and the consequent γ. With Nilsson's probabilistic logic, this is interpreted as:

$$\neg\sigma_1 \vee \sigma_0 : [\alpha], \quad \sigma_1 : [\beta], \quad \sigma_0 : [\gamma],$$

which gives rise to equations

$$\pi(\neg\sigma_1 \wedge \sigma_0) + \pi(\sigma_1 \wedge \sigma_0) + \pi(\neg\sigma_1 \wedge \neg\sigma_0) = \alpha, \tag{17}$$

$$\pi(\sigma_1 \wedge \sigma_0) + \pi(\sigma_1 \wedge \neg\sigma_0) = \beta, \tag{18}$$

$$\pi(\sigma_1 \wedge \sigma_0) + \pi(\neg\sigma_1 \wedge \sigma_0) = \gamma. \tag{19}$$

With probabilistic rules discussed in this work, the interpretation to modus ponens is the three p-rules as follows.

$$\sigma_0 \leftarrow \sigma_1 : [\alpha], \quad \sigma_1 \leftarrow: [\beta], \quad \sigma_0 \leftarrow: [\gamma],$$

which gives rise to equations 5, 18 and 19. The two shaded polyhedrons shown in Table 2 illustrate probabilistic consistent regions for α, β and γ, with probabilistic logic and probabilistic rule, respectively, as defined by their corresponding equations together with equations 1 and 2. The consistent region in the probabilistic logic case is a tetrahedron, with vertices (0,0,1), (1,0,0), (1,1,0) and (1,1,1). The consistent region in the probabilistic rule case is an octahedron, with vertices (0,0,0), (0,0,1), (0,1,0), (1,0,0), (1,1,0) and (1,1,1). It is argued in [18] that the conditional probability interpretation to modus ponens is more reasonable than the probabilistic logic interpretation in practical settings.

The principal benefit of this analysis comes from observing that both methods are nothing but imposing constraints on the feasible regions of the spaces defined by clauses (in the case of probabilistic logic) or p-rules (in the case of probabilistic rules). In this sense, reasoning on such probability and logic combined forms is about identifying feasible regions determined by solutions to Π in $A\Pi = B$.[7]

It is recognised that solving Π with large matrix A is difficult. The size of A is exponential to the number of sentences in the language; and linear programming methods are super-linear to the size of the CC set (as we illustrate in Section 4). Nilsson suggests that partition could be considered on B so Π can be solved with divide-and-conquer techniques. However, as [19] show that PSAT is NP-complete in its general form, it becomes more plausible to consider important and/or practically useful instances of the generic PSAT problem where reasoning does not rely on exact solution to the probability distribution on the CC set.

A few such instances are considered in the literature. For example, Williamson [20] discusses the case when sentences are *disjoint*. In such cases, for any two sentences σ_0, σ_1 in a language, we have

$$\Pr(\sigma_0 \wedge \sigma_1) = 0.$$

Henderson et.al [9,21,22] discuss the case when sentences are *independent*. In such cases, for any two sentences σ_0, σ_1 in a language,

$$\Pr(\sigma_0 \wedge \sigma_1) = \Pr(\sigma_0)\Pr(\sigma_1).$$

Both settings can greatly reduce the complexity of reasoning as one does not need to explicitly consider joint probabilities amongst sentences and thus one can work in the space defined by n sentences in the language, instead of considering solutions in the 2^n space formed by the CC set.

However, we believe neither of the two is suitable in probabilistic rule settings such as the one discussed in this work, as they both trivialise conditional probabilities. In other words, as $\Pr(\sigma_0|\sigma_1) = \Pr(\sigma_0 \wedge \sigma_1)/\Pr(\sigma_1)$ by the definition of conditional probability, assuming $\Pr(\sigma_0 \wedge \sigma_1) = 0$ makes $\Pr(\sigma_0|\sigma_1) = 0$; whereas assuming $\Pr(\sigma_0 \wedge \sigma_1) = \Pr(\sigma_0)\Pr(\sigma_1)$ makes $\Pr(\sigma_0|\sigma_1) = \Pr(\sigma_0)$. Effectively, these two assumptions make us commit to

$$\sigma \leftarrow \_ : [0] \text{ and } \sigma \leftarrow \_ : [\Pr(\sigma)]$$ [8]

for all p-rules over all sentences σ, respectively. Since neither of the two seems realistic in practical settings, we stay with solving the joint distribution on the CC set for computing sentence probabilities.[9]

[7] Constructions of A differ between Nilsson's probabilistic logic and this work. However, both are designed for solving the full joint probability distribution over the CC set.

[8] Here, _ stands for an anonymous variable as in Prolog.

[9] On this note, PABA also forces independence between their probabilistic parameters by having $\Pr(\omega) = \prod_{(Q,[a:x]) \in GE_\omega} x$, where $(Q, [a : x])$ is a deduction for a and GE_ω is the set of grounded extensions containing all possible worlds, in which a possible world ω is an element in the CC set of all probabilistic parameters (Definition 2.1 and Lemma 2.1 in [1]). However, as PABA also supports non-probabilistic rules and assumptions, the independence assumption is not imposed on all sentences in a PABA framework as some of them are not probabilistic.

6. Conclusion

In this work, we introduce a generalisation to probabilistic rules (p-rule) used in Probabilistic Assumption-based Argumentation. We show that by introducing Rule Probabilistic Satisfiability, we can accommodate probabilistic rules forming cycles and allow multiple rules with the same head but different bodies in the same p-rule set. A reasoning method using linear programming is introduced with a software implementation developed. This work can be viewed as a building block for probabilistic structured argumentation frameworks that use rules to construct arguments. Future work will focus on (more) efficient reasoning and / or approximation approaches.

Acknowledgements

I thank Dr Kristijonas Čyras for insightful discussions on PABA. I also thank the anonymous referees for their helpful suggestions.

References

[1] Dung PM, Thang PM. Towards (Probabilistic) Argumentation for Jury-based Dispute Resolution. In: Baroni P, Cerutti F, Giacomin M, Simari GR, editors. Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010. vol. 216 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2010. p. 171-82. Available from: https://doi.org/10.3233/978-1-60750-619-5-171.

[2] Čyras K, Fan X, Schulz C, Toni F. Assumption-Based Argumentation: Disputes, Explanations, Preferences. IfCoLog JLTA. 2017;4(8).

[3] Rienstra T. Towards a Probabilistic Dung-style Argumentation System. In: Ossowski S, Toni F, Vouros GA, editors. Proceedings of the First International Conference on Agreement Technologies, AT 2012, Dubrovnik, Croatia, October 15-16, 2012. vol. 918 of CEUR Workshop Proceedings. CEUR-WS.org; 2012. p. 138-52. Available from: http://ceur-ws.org/Vol-918/111110138.pdf.

[4] Hunter A. Reasoning with Inconsistent Knowledge using the Epistemic Approach to Probabilistic Argumentation. In: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020; 2020. p. 496-505. Available from: https://doi.org/10.24963/kr.2020/50.

[5] Hunter A. Argument Strength in Probabilistic Argumentation Using Confirmation Theory. In: Vejnarová J, Wilson N, editors. Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceedings. vol. 12897 of Lecture Notes in Computer Science. Springer; 2021. p. 74-88.

[6] Hung ND. Inference procedures and engine for probabilistic argumentation. International Journal of Approximate Reasoning. 2017;90:163-91. Available from: https://doi.org/10.1016/j.ijar.2017.07.008.

[7] Čyras K, Heinrich Q, Toni F. Computational complexity of flat and generic Assumption-Based Argumentation, with and without probabilities. Artificial Intelligence. 2021;293:103449. Available from: https://doi.org/10.1016/j.artint.2020.103449.

[8] Nilsson NJ. Probabilistic Logic. Artificial Intelligence. 1986;28(1):71-87. Available from: https://doi.org/10.1016/0004-3702(86)90031-7.

[9] Henderson TC, Simmons R, Serbinowski B, Cline M, Sacharny D, Fan X, et al. Probabilistic sentence satisfiability: An approach to PSAT. Artificial Intelligence. 2020;278. Available from: https://doi.org/10.1016/j.artint.2019.103199.

[10] Potra FA, Wright SJ. Interior-point methods. Journal of Computational and Applied Mathematics. 2000;124(1):281-302. Available from: https://www.sciencedirect.com/science/article/pii/S0377042700004337.

[11] Kohlas J. Probabilistic argumentation systems: A new way to combine logic with probability. Journal of Applied Logic. 2003;1(3-4):225-53. Available from: https://doi.org/10.1016/S1570-8683(03)00014-4.

[12] Hunter A. Some Foundations for Probabilistic Abstract Argumentation. In: Proc. of COMMA. vol. 245. IOS Press; 2012. p. 117-28. Available from: https://doi.org/10.3233/978-1-61499-111-3-117.

[13] Thimm M. A Probabilistic Semantics for abstract Argumentation. In: Raedt LD, Bessiere C, Dubois D, Doherty P, Frasconi P, Heintz F, et al., editors. ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012. vol. 242 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2012. p. 750-5. Available from: https://doi.org/10.3233/978-1-61499-098-7-750.

[14] Gabbay DM, Rodrigues O. Probabilistic Argumentation: An Equational Approach. Logica Universalis. 2015;9(3):345-82. Available from: https://doi.org/10.1007/s11787-015-0120-1.

[15] Hunter A, Thimm M. Probabilistic Reasoning with Abstract Argumentation Frameworks. Journal of Artificial Intelligence Research. 2017;59:565-611. Available from: https://doi.org/10.1613/jair.5393.

[16] Fazzinga B, Flesca S, Furfaro F. Complexity of fundamental problems in probabilistic abstract argumentation: Beyond independence. Artif Intell. 2019;268:1-29. Available from: https://doi.org/10.1016/j.artint.2018.11.003.

[17] Mantadelis T, Bistarelli S. Probabilistic abstract argumentation frameworks, a possible world view. Int J Approx Reason. 2020;119:204-19. Available from: https://doi.org/10.1016/j.ijar.2019.12.006.

[18] Nilsson N. Probabilistic Logic Revisited. Artificial Intelligence. 1993;59(1-2):39-42. Available from: https://doi.org/10.1016/0004-3702(93)90167-A.

[19] Georgakopoulos GF, Kavvadias DJ, Papadimitriou CH. Probabilistic satisfiability. Journal of Complexity. 1988;4(1):1-11. Available from: https://doi.org/10.1016/0885-064X(88)90006-4.

[20] Williamson J. Probability Logic. In: Gabbay D, Johnson R, Ohlbach HJ, Woods J, editors. Handbook of the Logic of Argument and Inference: the Turn Toward the Practical. Elsevier; 2002. p. 397-424.

[21] Henderson T, Simmons R, Serbinowski B, Fan X, Mitiche A, Cline M. Probabilistic Logic for Intelligent Systems. In: Strand M, Dillmann R, Menegatti E, Ghidoni S, editors. Intelligent Autonomous Systems 15. vol. 867 of Advances in Intelligent Systems and Computing. Cham: Springer International Publishing; 2018. p. 129-41. Available from: https://doi.org/10.1007/978-3-030-01370-7_11.

[22] Henderson TC, Simmons R, Sacharny D, Mitiche A, Fan X. A probabilistic logic for multi-source heterogeneous information fusion. In: 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI 2017, Daegu, Korea (South), November 16-18, 2017. IEEE; 2017. p. 530-5. Available from: https://doi.org/10.1109/MFI.2017.8170375.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220150

Composite Argumentation Systems with ML Components

Nguyen Duy HUNG [a,1], Nam-Van HUYNH [b] and Thanaruk THEERAMUNKONG [a]
Tho-Quy NHU [c]

[a] *Sirindhorn International Institute of Technology, Thailand*
[b] *Japan Advanced Institute of Science and Technology, Japan*
[c] *Hanoi University of Industry, Vietnam*

Abstract. Today AI systems are rarely made without Machine Learning (ML) and this inspires us to explore what aptly called composite argumentation systems with ML components. Concretely, against two theoretical backdrops of PABA (Probabilistic Assumption-based Argumentation) and DST (Dempster-Shafer Theory), we present a framework for such systems called c-PABA. It is argued that c-PABA lends itself to a development tool as well and to demonstrate we show that DST-based ML classifier combination and multi-source data fusion can be implemented as simple c-PABA frameworks.

1. Introduction

Today AI systems are rarely made without Machine Learning (ML) though one may criticize the overuse of ML especially in tasks demanding explainability. On the other hand, Argumentation is widely viewed as an inherently explainable AI formalism, however practical argumentation systems are still hard to develop. This inspires us to explore a synthesis of ML and Argumentation that fosters AI systems with the elements of both formalisms. Concretely, against two theoretical backdrops of PABA (Probabilistic Assumption-based Argumentation [5]) and DST (Dempster-Shafer Theory), we present a framework for composite argumentation systems with ML components called c-PABA. It is argued that c-PABA lends itself to a development tool for these systems as well and to demonstrate we show that DST-based ML classifier combination and multi-source data fusion can be implemented as simple c-PABA frameworks. The rationale behind our selection of two theoretical backdrops (DST and PABA) is as follows. DST, which alone is often described as a generalisation of the probability theory, has long been proven very suitable for representing knowledge under uncertainty and ignorance. Predictions of ML models belong to knowledge of this kind, and hence ML models can be viewed as sources generating DST data. Since a composite argumentation system in our view can contain many components some of which may be ML models, DST lends itself to an appropriate model for the information exchanged between these components. To model the workings of these components as well as the whole composite argumentation system,

[1] Corresponding author; E-mail: hung.nd.siit@gmail.com

we choose PABA rather than an abstract PA model such as [10] because of two reasons: a) PABA deals the material out of which arguments are constructed, and hence allows us to go down to the level of DST data exchanged between components; b) PABA reasoning engines for precise results [7] as well as approximate any-time results [9] are recently available. Such engines can run c-PABA frameworks (since as will be seen, c-PABA can be translated back to PABA), and hence one can use c-PABA as a design as well as development tool for the above described kind of composite argumentation systems. The remaining of this paper is structured as follows. We recall DST and PABA in Section 2. Then we develop three complementary techniques respectively for: translating DST data to PABA (Section 3), generating DST data from PABA, and fusing DST data with existing PABA frameworks (Section 4). We then accumulate these techniques to present the c-PABA framework (Section 5). Due to the lack of space, the proofs of theorems and lemmas are moved to an on-line appendix[2].

2. Background

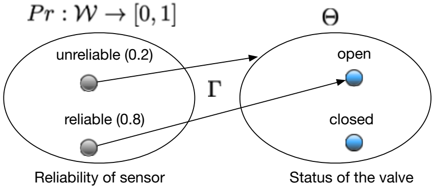
2.1. Dempster Shafer Theory (DST [3,13])

Definition 1. ***A Demspter's structure*** *is a tuple $\mathcal{D} = (\mathcal{W}, Pr, \Gamma, \Theta)$ where Θ is an exhaustive set of mutually exclusive answers for some question (Frame of Discernment or FoD for short); $\mathcal{W}$ is a finite set of possible worlds; $Pr : \mathcal{W} \to [0,1]$ is a probability distribution; $\Gamma : \mathcal{W} \to 2^{\Theta}$ is a multi-valued mapping from $\mathcal{W}$ into Θ. For $X \subseteq \Theta$,* ***the degree of belief*** *and* ***the degree of plausibility in*** *X are defined as follows.*

$$Bel_{\mathcal{D}}(X) \triangleq \sum_{\omega \in \mathcal{W}: \emptyset \neq \Gamma(\omega) \subseteq X} Pr(\omega) \quad \text{and} \quad Pl_{\mathcal{D}}(X) \triangleq \sum_{\omega \in \mathcal{W}: \emptyset \neq \Gamma(\omega) \cap X} Pr(\omega)$$

Intuitively Γ says that if ω is the actual world, then the answer is in $\Gamma(\omega)$. The interval $[Bel_{\mathcal{D}}(X), Pl_{\mathcal{D}}(X)]$ delineates the probability that the answer is in X. For example, suppose that a sensor which is unreliable in 20% of the times, is installed to check a valve's status. If the sensor indicates "valve open", the best conclusion one can make is $0.8 \leq Prob(open) \leq 1$ because if the sensor is unreliable, one has no information about the valve's status. Hence one does not want to represent the observation by a standard probability distribution over space $\Theta = \{open, closed\}$ but by a Dempster's structure depicted in Fig. 1 with: $\Theta = \{open, closed\}$, $\Gamma = \{reliable \mapsto \{open\}, unreliable \mapsto \Theta\}$, and $\mathcal{W} = \{reliable, unreliable\}$ with two possible worlds having probabilities 0.8 and 0.2. Clearly $[Bel_{\mathcal{D}}(\{open\}), Pl_{\mathcal{D}}(\{open\})] = [0.8, 1]$.

Figure 1. A Dempster's structure $\mathcal{D} = (\mathcal{W}, Pr, \Gamma, \Theta)$



[2] http://ict.siit.tu.ac.th/~hung/comma22-proofs.pdf

Definition 2. *A **mass function over FoD** Θ is a function $m : 2^{\Theta} \to [0,1]$ such that $\sum\limits_{X \subseteq \Theta} m(X) = 1$. For $X \subseteq \Theta$, $Bel_m(X) \triangleq \sum\limits_{Y \subseteq \Theta : \emptyset \neq Y \subseteq X} m(Y)$ and $Pl_m(X) \triangleq \sum\limits_{Y \subseteq \Theta : \emptyset \neq X \cap Y} m(Y)$.*

And here are some more definitions. $X \subseteq \Theta$ is said to be a **focal** element of $m : 2^{\Theta} \to [0,1]$ iff $m(X) \neq 0$. The set of all focal elements of m is denoted by $focal(m)$. The set $\{(X_i, \mu_i)\}_{i=1}^{|focal(m)|}$ where $X_i \in focal(m)$ and $\mu_i = m(X_i)$ is called the **focal specification** of m. For a Demspter's structure $\mathcal{D} = (\mathcal{W}, Pr, \Gamma, \Theta)$, $m_{\mathcal{D}}$ denotes the mass function: $m_{\mathcal{D}}(X) = \sum\limits_{\omega \in \Theta : \Gamma(\omega) = X} Pr(\omega)$. Clearly $Bel_{\mathcal{D}}(X) = Bel_{m_{\mathcal{D}}}(X)$ and $Pl_{\mathcal{D}}(X) = Pl_{m_{\mathcal{D}}}(X)$.

Example 1. *Consider $\mathcal{D} = (\mathcal{W}, Pr, \Gamma, \Theta)$ where $\Theta = \{\theta_1, \theta_2, \theta_3\}$; $(\mathcal{W}, Pr)$ is generated from two independent events α_1, α_2 with $Pr(\alpha_1) = 0.4$ and $Pr(\alpha_2) = 0.7$; and Γ is shown in the table below. The focal specification of $m_{\mathcal{D}}$ is $\{(\emptyset, 0.28), (\{\theta_2\}, 0.12), (\{\theta_2, \theta_3\}, 0.6)\}$.*

$\mathcal{W}$	$Pr(\omega_i)$	$\Gamma(\omega_i)$
$\omega_1 = \{\alpha_1, \alpha_2\}$	$0.4 \times 0.7 = 0.28$	$\{\}$
$\omega_2 = \{\alpha_1, \neg\alpha_2\}$	$0.4 \times 0.3 = 0.12$	$\{\theta_2\}$
$\omega_3 = \{\neg\alpha_1, \alpha_2\}$	$0.6 \times 0.7 = 0.42$	$\{\theta_2, \theta_3\}$
$\omega_4 = \{\neg\alpha_1, \neg\alpha_2\}$	$0.6 \times 0.3 = 0.18$	$\{\theta_2, \theta_3\}$

In general mass functions can represent real-world data directly. Moreover those over the same FoD can be combined using various combination rules. Due to the lack of space, we focus on only the Smet's rule.

Definition 3. *Given two mass functions m_1, m_2 over the same FoD Θ, **Smet's rule** returns a combined mass function $m_1 \otimes m_2(X) \triangleq \sum\limits_{B \cap C = X} m_1(B) m_2(C)$, $\forall X \subseteq \Theta$.*

For a set $\mathcal{M}$ of mass functions over the same FoD (aka *a DST evidence base*), any order of applying $\otimes$ yields the same result, denoted $\bigotimes \mathcal{M}$. Hence the Smet's rule derives two functions $Bel_{\mathcal{M}}(X) \triangleq Bel_{\bigotimes \mathcal{M}}(X)$ and $Pl_{\mathcal{M}}(X) \triangleq Pl_{\bigotimes \mathcal{M}}(X)$, which together define the Smet's semantics for $\mathcal{M}$.

2.2. Argumentation frameworks

An **Abstract Argumentation** (AA [4]) framework a pair $(\mathcal{A}rg, \mathcal{A}tt)$ of a set $\mathcal{A}rg$ of arguments and an attack relation $\mathcal{A}tt \subseteq \mathcal{A}rg \times \mathcal{A}rg$. An argument $A \in \mathcal{A}rg$ is *acceptable wrt to $S \subseteq \mathcal{A}rg$* iff S attacks every argument attacking A. S is *admissible* iff S does not attack itself (aka conflict-free) and each argument in S is acceptable wrt S; a *preferred* extension iff S is a maximal (wrt $\subseteq$) admissible set. $A \in \mathcal{A}rg$ is *credulously* (resp. *skeptically*) acceptable if it is acceptable wrt a preferred extension (resp. any preferred extension).

Assuming a logical language $\mathcal{L}$, an **Assumption-based Argumentation** (ABA [1]) framework is a tuple $\mathcal{F} = (\mathcal{R}, \mathcal{A}, \overline{})$ where: $\mathcal{R}$ is a set of inference rules of the form $l_0 \leftarrow l_1, \ldots, l_n$ ($n \geq 0$, $l_i \in \mathcal{L}$); $\mathcal{A} \subseteq \mathcal{L}$ is a set of assumptions; $\overline{} : \mathcal{A} \to \mathcal{L}$ maps each assumption to its contrary. In this paper we restrict ourselves to *flat* ABA frameworks where assumptions do not appear in the heads of inference rules. An argument (Q, π) for $\pi \in \mathcal{L}$ supported by a set of assumptions $Q \subseteq \mathcal{A}$ is a backward deduction from π to Q. An argument (Q, π) attacks an argument (Q', π') if $\pi = \overline{a}$ for some $a \in Q'$. A proposition π is said to be credulously/skeptically acceptable, denoted $\mathcal{F} \vdash_{cr} \pi$ (resp. $\mathcal{F} \vdash_{sk} \pi$) if in the

AA framework consisting of above defined arguments and attacks, there is a credulously (skeptically) acceptable argument (Q, π). For short we may specify an ABA framework $(\mathcal{R}, \mathcal{A}, \overline{})$ by just a pair $(\mathcal{R}, \mathcal{A})$ and for each assumption $a \in \mathcal{A}$, we write $\overline{a}$ in inference rules of $\mathcal{R}$ as if $\overline{a}$ were a "legal" sentence (which of course refers to the one returned by the omitted contrary function $\overline{}$). Inference rules of $\mathcal{R}$ with the same head are grouped together by connecting their bodies with symbol | as demonstrated by the example below.

Example 2. *A flat ABA framework describing $\Theta = \{\theta_1, \theta_2, \theta_3\}$ as a set of exhaustive and mutually exclusive propositions is $\mathcal{F} = (\mathcal{A}, \mathcal{R})$ with $\mathcal{A} = \{\theta_1, \theta_2, \theta_3, \{\theta_1, \theta_2\}, \{\theta_1, \theta_3\}, \{\theta_2, \theta_3\}, \{\theta_1, \theta_2, \theta_3\}\}$ saying that one can assume any proposition $\theta_i \in \Theta$ to be true. Consequently one can also assume any disjunction of these propositions. Note that these disjunctions are written in the set-based clausal form (e.g. $\{\theta_1, \theta_2\}$ means $\theta_1 \vee \theta_2$) so that in the general case (as in Def. 6) we can simply write $\mathcal{A} = 2^{\Theta} \setminus \{\emptyset\}$. $\mathcal{R}$ consists of two groups of rules:*

- $\overline{\theta_1} \leftarrow \neg\theta_1 \mid \theta_2 \mid \theta_3.$ $\overline{\theta_2} \leftarrow \theta_1 \mid \neg\theta_2 \mid \theta_3.$ $\overline{\theta_3} \leftarrow \theta_1 \mid \theta_2 \mid \neg\theta_3.$ *saying that each assumption $\theta_i \in \Theta$ can be disproved by either proving its classical negation $\neg\theta_i$ or proving any mutually exclusive assumption θ_j, $j \neq i$.*
- $\overline{\{\theta_1, \theta_2\}} \leftarrow \overline{\theta_1}, \overline{\theta_2}.$ $\overline{\{\theta_1, \theta_3\}} \leftarrow \overline{\theta_1}, \overline{\theta_3}.$ $\overline{\{\theta_2, \theta_3\}} \leftarrow \overline{\theta_2}, \overline{\theta_3}.$ $\overline{\{\theta_1, \theta_2, \theta_3\}} \leftarrow \overline{\theta_1}, \overline{\theta_2}, \overline{\theta_3}.$ *saying that a disjunction of several assumptions is disproved by proving each and every contraries of the assumptions.*

Some arguments of $\mathcal{F}$ are $(\{\theta_i\}, \{\theta_1, \theta_2, \theta_3\})$ and $(\{\theta_i\}, \theta_i)$ with $i \in \{1,2,3\}$. Clearly $\mathcal{F} \vdash_{sk} \{\theta_1, \theta_2, \theta_3\}$ (saying that some element of Θ holds certainly) and $\mathcal{F} \vdash_{cr} \theta_i$ but $\mathcal{F} \not\vdash_{sk} \theta_i$ (saying that it is probable but not certain that θ_i holds).

2.3. PABA

A PABA [5] framework can be seen as a probability distribution of ABA frameworks. In this paper, we focus on a class of PABA frameworks, called Bayesian.

Definition 4. *A (Bayesian) PABA framework is a triple $\mathcal{P} = (\mathcal{V}, Pr, \mathcal{F})$ where $\mathcal{F} = (\mathcal{R}, \mathcal{A})$ is an ABA framework, and*

1. *$\mathcal{V}$ is a finite set of so-called **probabilistic assumptions** such that no elements of $\mathcal{V} \cup \neg\mathcal{V}$*[3] *occurs in $\mathcal{A}$ or in the head of a rule in $\mathcal{R}$.*
2. *Pr is a probability distribution over the set of all possible worlds, where a* possible world *is a maximal (wrt set inclusion)* consistent[4] *subset of $\mathcal{V} \cup \neg\mathcal{V}$.*

Definition 5. *Given $\mathcal{P} = (\mathcal{V}, Pr, \mathcal{F})$, the **acceptability probability** of a proposition π under semantics s is $Prob(\mathcal{P} \vdash_s \pi) \triangleq \sum_{\omega \in \mathcal{W} : \mathcal{F}_\omega \vdash_s \pi} Pr(\omega)$, where $\mathcal{W}$ is the set of all possible worlds and $\mathcal{F}_\omega$ is the ABA framework obtained from $\mathcal{F}$ by adding facts $\{\alpha \leftarrow \mid \alpha \in \omega\}$.*

Note that the above definitions leave it open the representation of *Pr* (hence they do not demand any probabilistic relationships between probabilistic assumptions). However for convenience we shall specify *Pr* by a Problog [6] program, using especially *probabilistic facts* and *annotated disjunctions* as done in [9]. Note that a probabilistic

[3] $\neg\mathcal{V} = \{\neg\alpha \mid \alpha \in \mathcal{V}\}$

[4] No α and $\neg\alpha$ co-exist in the set.

fact of Prolog is a sentence of the form "$p :: x.$" where $p \in [0,1]$ and x is a proposition saying that x holds with probability p. An annotated disjunction is of the form "$p_1 :: x_1; \ldots; p_i :: x_i; \ldots; p_n :: x_n.$" saying that propositions $x_1, \ldots, x_i, \ldots x_n$ are mutually exclusive and hold with respective probabilities $p_1, \ldots, p_i, \ldots p_n$ whose sum must equal 1. Let's have an example for illustration.

Example 3. *Consider PABA framework $\mathcal{P} = (\mathcal{V}, Pr, \mathcal{F})$ where $\mathcal{V} = \{\alpha_1, \alpha_2\}$; Pr is Problog program $\{0.4 :: \alpha_1.\ \ 0.7 :: \alpha_2.\}$ with two probabilistic facts (saying that α_1, α_2 hold with probabilities 0.4 and 0.7 respectively); $\mathcal{F}$ is obtained from the ABA framework in Example 2 by adding the following rules:*

$\neg\theta_1 \leftarrow \alpha_1, \alpha_2.\quad \neg\theta_1 \leftarrow \alpha_1, \neg\alpha_2.\quad \neg\theta_1 \leftarrow \neg\alpha_1, \alpha_2.$
$\neg\theta_2 \leftarrow \alpha_1, \alpha_2.\quad \neg\theta_3 \leftarrow \alpha_1, \neg\alpha_2.\quad \neg\theta_1 \leftarrow \neg\alpha_1, \neg\alpha_2$
$\neg\theta_3 \leftarrow \alpha_1, \alpha_2.$

From the acceptabilities of θ_2 in different possible worlds shown in the table below, we have $Prob(\mathcal{P} \vdash_{sk} \theta_2) = 0.12$ but $Prob(\mathcal{P} \vdash_{cr} \theta_2) = 0.12 + 0.42 + 0.18$.

ω	$\mathcal{F}_\omega \vdash_{cr} \theta_2$?	$\mathcal{F}_\omega \vdash_{sk} \theta_2$?	$Pr(\omega)$
$\omega_1 = \{\alpha_1, \alpha_2\}$	*No*	*No*	0.28
$\omega_2 = \{\alpha_1, \neg\alpha_2\}$	*Yes*	*Yes*	0.12
$\omega_3 = \{\neg\alpha_1, \alpha_2\}$	*Yes*	*No*	0.42
$\omega_4 = \{\neg\alpha_1, \neg\alpha_2\}$	*Yes*	*No*	0.18

3. Translating DST data into PABA

In this section we show that any DST data, be it a Dempster's structure, a DST mass function or a DST evidence base, can always be translated into a PABA framework. Let's start by translating FoDs - the basic component of all forms of DST data, to ABA frameworks. As suggested by Example 2, each possible answer θ of the given FoD Θ is represented by an assumption the contrary of which can be proven by either proving either the classical negation $\neg\theta$, or by assuming an alternative answer θ' from Θ.

Definition 6. *For a FoD $\Theta = \{\theta_1, \ldots, \theta_i, \ldots, \theta_k\}$,* ***the canonical ABA translation of*** *Θ, denoted $\mathcal{FD}_\Theta$, is the ABA framework $(\mathcal{A}_\Theta, \mathcal{R}_\Theta)$ where*

1. *$\mathcal{A}_\Theta = 2^\Theta - \{\emptyset\}$ saying that any non-empty subset of Θ may contain the actual answer (For simplicity, a singleton set $\{\theta\} \in \mathcal{A}_\Theta$ is written as θ[5]).*
2. *$\mathcal{R}_\Theta$ is the minimal set such that*
   (a) *For each $\theta_i \in \Theta$, $\mathcal{R}_\Theta$ contains $\overline{\theta_i} \leftarrow \theta_1 \mid \cdots \mid \theta_{i-1} \mid \neg\theta_i \mid \theta_{i+1} \mid \cdots \mid \theta_k$. saying that θ_i can be disproved by proving its classical negation $\neg\theta_i$ or taking an alternative assumption θ_j, $j \neq i$.*
   (b) *For each subset $X \in \mathcal{A}_\Theta$ where $|X| \geq 2$, $\mathcal{R}_\Theta$ contains a rule with head $\overline{X}$ and body $\{\overline{\theta} \mid \theta \in X\}$ saying that the answer is not in X if every $\theta \in X$ is disproved.*

Example 4. *For FoD $\Theta = \{\theta_1, \theta_2, \theta_3\}$, $\mathcal{FD}_\Theta$ coincides with the ABA framework given in Example 2.*

[5] So Θ is a subset as well as an element of $\mathcal{A}_\Theta$.

The following lemma asserts that $\mathcal{FD}_\Theta$ and FoD Θ are "semantically equivalent"

Lemma 1. *Let $\Theta = \{\theta_1, \dots, \theta_k\}$ be a FoD. For any $X \in 2^\Theta$,*

1. *If $X \neq \Theta, \emptyset$, then $\mathcal{FD}_\Theta \vdash_{cr} X$ but $\mathcal{FD}_\Theta \not\vdash_{sk} X$ (representing that X possibly but not surely contains the answer).*
2. *If $X = \Theta$, then $\mathcal{FD}_\Theta \vdash_{sk} X$ (representing that Θ surely contains the answer); If $X = \emptyset$, then $\mathcal{FD}_\Theta \not\vdash_s X$ for any semantics s.*

Note that there are different ABA frameworks with the same semantics as $\mathcal{FD}_\Theta$, for example the one obtained from $\mathcal{FD}_\Theta$ by adding a rule $\neg\theta_i \leftarrow false$. However $\mathcal{FD}_\Theta$ is clearly the most obvious. Likewise, a Dempster's structure $\mathcal{D}$ can be translated to PABA in many ways but the PABA framework $PABA_\mathcal{D}$ defined below is the most obvious.

Definition 7. *Let $\mathcal{D} = (\mathcal{W}, Pr, \Gamma, \Theta)$ be a Demspter's structure.* ***The canonical PABA translation of*** *$\mathcal{D}$ is the PABA framework $PABA_\mathcal{D} = (\mathcal{V}, Pr, \mathcal{F})$ where*

1. *$PABA_\mathcal{D}$ and $\mathcal{D}$ have the same set of possible worlds $\mathcal{W}$ and probability distribution Pr, and*
2. *$\mathcal{F} = (\mathcal{A}_\Theta, \mathcal{R}_\Theta \cup \mathcal{R}_\Gamma)$ is the ABA framework obtained from ABA framework $\mathcal{FD}_\Theta = (\mathcal{A}_\Theta, \mathcal{R}_\Theta)$ by adding a set of rules $\mathcal{R}_\Gamma = \bigcup_{\omega \in \mathcal{W}} \{\neg\theta \leftarrow \omega \mid \theta \in \Theta - \Gamma(\omega)\}$.*

Here $\mathcal{R}_\Gamma$ represents the multi-valued function $\Gamma : \mathcal{W} \to 2^\Theta$. Recall that for a possible world ω, Γ says that the answer must be in $\Gamma(\omega)$ or equivalently must not be some $\theta \in \Theta - \Gamma(\omega)$. Hence $\neg\theta \leftarrow \omega$ occurs as an inference rule in $\mathcal{R}_\Gamma$.

Example 5. *For the Demspter's structure $\mathcal{D}$ in Example 1, $PABA_\mathcal{D}$ coincides with the PABA framework in Example 3.*

The theorem below asserts that Demspter's structure $\mathcal{D}$ is semantically equivalent to the PABA framework $PABA_\mathcal{D}$.

Theorem 1. *Let $\mathcal{D} = (\mathcal{W}, Pr, \Gamma, \Theta)$ be a Demspter's structure. Then for any $X \in 2^\Theta$, $Pl_\mathcal{D}(X) = Prob(PABA_\mathcal{D} \vdash_{cr} X)$ and $Bel_\mathcal{D}(X) = Prob(PABA_\mathcal{D} \vdash_{sk} X)$.*

Now let's switch our attention to the remaining forms of DST data - mass functions and evidence bases. Recall that any mass function, say m, can be specified by a set of ordered pairs $\{(X_i, \mu_i)\}_{i=1}^{|focal(m)|}$ with X_i being a focal element and $\mu_i \in [0,1]$ being the mass of X_i. Obviously m can be translated to PABA in many ways, and the so-called *canonical PABA translation $PABA_m$* defined below uses a set of probabilistic assumptions $\{\phi_i^m \mid i \in \{1, \dots, |focal(m)|\}\}$ where the probability of ϕ_i^m is set to μ_i. To represent the mutually exclusiveness of focal elements, $PABA_m$ uses a Problog annotated disjunction "$\mu_1 :: \phi_1^m; \mu_2 :: \phi_2^m; \dots; \mu_{|focal(m)|} :: \phi_{|focal(m)|}^m$.". Finally the content of each focal element X_i is represented in $PABA_m$ by a set of rules $\{\neg\theta \leftarrow \phi_i^m \mid \theta \in \Theta - X_i\}$, where $\neg\theta \leftarrow \phi_i^m$ says that if the i^{th} focal element of m occurs then any $\theta \in \Theta - X_i$ cannot be the answer.

Definition 8. *Let $m = \{(X_i, \mu_i)\}_{i=1}^{|focal(m)|}$ be a mass function with FoD Θ.* ***The canonical PABA translation of*** *m, denoted $PABA_m$, is the PABA framework $(\mathcal{V}_m, Pr_m, \mathcal{F}_m)$ where*

1. *$\mathcal{V}_m = \{\phi_1^m, \dots, \phi_{|focal(m)|}^m\}$ and Pr_m is a Problog program consisting of only one annotated disjunction: $\mu_1 :: \phi_1^m; \mu_2 :: \phi_2^m; \dots; \mu_{|focal(m)|} :: \phi_{|focal(m)|}^m$.*

2. *$\mathcal{F}_m$ is the ABA framework obtained from the canonical ABA translation $\mathcal{FD}_\Theta$ of Θ by adding a set of rules $\bigcup_{1 \leq i \leq |focal(m)|} \{\neg\theta \leftarrow \phi_i^m \mid \theta \in \Theta - X_i\}$.*

The canonical PABA translation of a DST evidence base $\mathcal{M}$ (a set of mass functions over the same FoD Θ) defined below is simply the set union of the canonical PABA translations of individual mass functions.

Definition 9. ***The canonical PABA translation of a DST evidence base*** *$\mathcal{M}$ is the PABA framework $PABA_{\mathcal{M}} = (\mathcal{V}_{\mathcal{M}}, Pr_{\mathcal{M}}, \mathcal{F}_{\mathcal{M}})$ where $\mathcal{V}_{\mathcal{M}} = \bigcup_{m \in \mathcal{M}} \mathcal{V}_m$, $Pr_{\mathcal{M}} = \bigcup_{m \in \mathcal{M}} Pr_m$ and $\mathcal{F}_{\mathcal{M}} = \bigcup_{m \in \mathcal{M}} \mathcal{F}_m$ (where $\mathcal{V}_m$, Pr_m and $\mathcal{F}_m$ are defined in Def. 8).*

Theorem 2 asserts the semantic equivalence between m and the above defined canonical PABA translation of m.

Theorem 2. *Let m be a mass function over FoD Θ. Then for any $X \subseteq \Theta$, $Bel_m(X) = Prob(PABA_m \vdash_{sk} X)$ and $Pl_m(X) = Prob(PABA_m \vdash_{cr} X)$.*

More generally, for a DST evidence base $\mathcal{M}$, the Smet's semantics of $\mathcal{M}$ and the semantics of $PABA_{\mathcal{M}}$ coincide.

Theorem 3. *Let $\mathcal{M}$ be a DST evidence base over FoD Θ. Then for any $X \subseteq \Theta$, $Bel_{\mathcal{M}}(X) = Prob(PABA_{\mathcal{M}} \vdash_{sk} X)$ and $Pl_{\mathcal{M}}(X) = Prob(PABA_{\mathcal{M}} \vdash_{cr} X)$.*

4. Fusion and generation of DST data

In this section we present two techniques for: 1) fusing DST data with; and 2) generating DST data from existing PABA frameworks.

4.1. DST data fusion with PABA frameworks

Given a DST evidence base $\mathcal{M}$ and a PABA framework $\mathcal{P}$, the so-called *the s-union of $\mathcal{P}$ and $\mathcal{M}$* defined below is a structure obtained by taking set unions of the corresponding components of $\mathcal{P}$ and $PABA_{\mathcal{M}}$.

Definition 10. *Let $\mathcal{M}$ be a DST evidence base and $\mathcal{P} = (\mathcal{V}, Pr, \mathcal{F})$ be a PABA framework.* ***The s-union of $\mathcal{P}$ and $\mathcal{M}$****, denoted $\mathcal{P} \uplus \mathcal{M}$, is simply the triple $(\mathcal{V} \cup \mathcal{V}_{\mathcal{M}}, Pr \cup Pr_{\mathcal{M}}, \mathcal{F} \cup \mathcal{F}_{\mathcal{M}})$ obtained by taking set-unions of the corresponding components of $\mathcal{P}$ and the canonical PABA translation $PABA_{\mathcal{M}} = (\mathcal{V}_{\mathcal{M}}, Pr_{\mathcal{M}}, \mathcal{F}_{\mathcal{M}})$ of $\mathcal{M}$.*

Let's introduce a condition to ensure that $\mathcal{P} \uplus \mathcal{M}$ is a well-formed PABA framework.

Definition 11. *We say that a PABA framework $\mathcal{P} = (\mathcal{V}, Pr, \mathcal{F})$* ***syntactically complies*** *with a FoD Θ iff: 1) all assumptions and rules occurring $\mathcal{FD}_\Theta$ also occur in $\mathcal{F}$; and 2) for any $\theta \in \Theta$, $\neg\theta$ is not an assumption of $\mathcal{F}$.*

Lemma 2. *Let $\mathcal{M}$ be a DST evidence base over FoD Θ and $\mathcal{P}$ be a PABA framework that is syntactically complies with Θ. Then $\mathcal{P} \uplus \mathcal{M}$ is a well-formed PABA framework syntactically complying with Θ.*

Hence $\uplus$ can be viewed as a knowledge fusion operator. Though its applicability is limited (e.g. $\uplus$ cannot fuse two arbitrary PABA frameworks), $\uplus$ suffices for our purpose which is to fuse a DST evidence base $\mathcal{M}$ with an existing PABA framework $\mathcal{P}$. Now let's examine several properties of $\uplus$. Lemma 3 below says that $\uplus$ in fact encapsulates the translation technique from DST data to PABA presented in the previous section.

Lemma 3. *Let $\mathcal{M}$ be a DST evidence base over FoD Θ and $\mathcal{P} = (\emptyset, \emptyset, \mathcal{FD}_{\Theta})$. Then $\mathcal{M} \uplus \mathcal{P}$ is exactly the canonical PABA translation of $\mathcal{M}$.*

As the underlying operation in $\uplus$ is set union, $\uplus$ inherits many desirable properties from $\cup$. For example, $\mathcal{P} \uplus \emptyset = \mathcal{P}$; and $\mathcal{P} \uplus \mathcal{M} = (\mathcal{P} \uplus \mathcal{M}_1) \uplus \mathcal{M}_2$ if $\mathcal{M}_1 \cup \mathcal{M}_2 = \mathcal{M}$.

4.2. DST data generation from PABA frameworks

Obviously a Dempster's structure generated from a PABA framework should share the same probability space with the framework.

Definition 12. *We say that a PABA framework $\mathcal{P} = (\mathcal{V}, Pr, \mathcal{F})$ **generates** a Dempster's structure $\mathcal{D} = (\mathcal{W}, Pr, \Gamma, \Theta)$, written $\mathcal{P} \xrightarrow{\Theta} \mathcal{D}$, if*

1. *$\mathcal{D}$ and $\mathcal{P}$ have the same set of possible worlds $\mathcal{W}$ and the same probability distribution $Pr : \mathcal{W} \rightarrow [0,1]$, and*
2. *for each $\omega \in \mathcal{W}$, $\Gamma(\omega) = \{\theta \in \Theta \mid \mathcal{F}_{\omega} \vdash_{cr} \theta\}$.*

That is, we add θ into $\Gamma(\omega)$ just in case there is a credulously accepted argument for θ in $\mathcal{F}_{\omega}$. Generated Dempster's structures can be converted into mass functions, and so:

Definition 13. *A PABA framework $\mathcal{P}$ **generates** a mass function m, written $\mathcal{P} \xrightarrow{\Theta} m$, if $\mathcal{P} \xrightarrow{\Theta} \mathcal{D}$ and m coincides $m_{\mathcal{D}}$.*

It is easy to see that:

Lemma 4. *Let $PABA_{\mathcal{M}}$ be the canonical PABA translation of DST evidence base $\mathcal{M}$ over FoD Θ. Then $PABA_{\mathcal{M}} \xrightarrow{\Theta} \bigotimes_S \mathcal{M}$ (hence for any $m \in \mathcal{M}$, $PABA_m \xrightarrow{\Theta} m$).*

It is worth noting that the generating PABA framework $\mathcal{P}$ does not have to satisfy any constraint with respect to the FoD Θ of the generated DST data. Concretely:

Lemma 5. *Let $\mathcal{P}$ be a PABA framework. For any FoD Θ, there is an unique Dempster's structure $\mathcal{D}$ (resp. mass function m) such that $\mathcal{P} \xrightarrow{\Theta} \mathcal{D}$ (resp. $\mathcal{P} \xrightarrow{\Theta} m$).*

Hence a PABA framework can generate DST data over any FoD. This flexibility, however, may lead to possible semantic differences between the generated DST data and the generating PABA framework. Let's study conditions for preventing such differences.

4.3. Relationships between generated DST data and generating PABA framework

Theorem 4 below says that the degree of plausibility wrt generated DST data and the credulous semantics of the generating PABA framework always coincide.

Theorem 4. *Suppose $\mathcal{P} \overset{\Theta}{\to} m$. Then for any $\theta \in \Theta$, $Pl_m(\theta) = Prob(\mathcal{P} \vdash_{cr} \theta)$.*

However in general $Bel_m(\theta)$ and $Prob(\mathcal{P} \vdash_{sk} \theta)$ may be different as illustrated by the following example.

Example 6. *Consider $\mathcal{P} = (\emptyset, \emptyset, \mathcal{F})$ with $\mathcal{F} = (\emptyset, \{a \leftarrow, b \leftarrow\})$. For $\Theta = \{a, b\}$, we have $\mathcal{P} \overset{\Theta}{\to} m = \{\{a, b\} \mapsto 1\}$. Hence $Bel_m(a) = Bel_m(b) = 0$. However $Prob(\mathcal{P} \vdash_{sk} a) = Prob(\mathcal{P} \vdash_{sk} b) = 1$.*

Now let's introduce a condition that ensures that $Bel_m(\theta) = Prob(\mathcal{P} \vdash_{sk} \theta)$.

Definition 14. *We say that a PABA framework $\mathcal{P}$* ***semantically complies*** *with a FoD Θ iff for each possible world ω and $\theta \in \Theta$, $\mathcal{F}_\omega \vdash_{sk} \theta$ iff $\{x \in \Theta \mid \mathcal{F}_\omega \vdash_{cr} x\} = \{\theta\}$.*

For example, it is easy to see that for a DST evidence base $\mathcal{M}$ over FoD Θ, the PABA canonical translation $PABA_{\mathcal{M}}$ of $\mathcal{M}$ always semantically complies with Θ.

Theorem 5 given below and the previous Theorem 4 say that semantic compliance is a sufficient condition for ensuring the semantic coincidence between generated DST data and its generating PABA framework.

Theorem 5. *Suppose $\mathcal{P} \overset{\Theta}{\to} m$. If $\mathcal{P}$ semantically complies with Θ then for any $\theta \in \Theta$, $Bel_m(\theta) = Prob(\mathcal{P} \vdash_{sk} \theta)$.*

One might ask whether syntactic compliance (defined in Def. 11) ensures semantic compliance. The following example shows that it does not.

Example 7. *Consider FoD $\Theta = \{\theta_1, \theta_2\}$ and $\mathcal{P} = (\emptyset, \emptyset, \mathcal{F})$ where $\mathcal{F}$ is the ABA framework obtained from $\mathcal{FD}_\Theta$ by adding rules $\{\neg\theta_1 \leftarrow .\ \neg\theta_2 \leftarrow a.\ \overline{a} \leftarrow b.\ \overline{b} \leftarrow a\}$ and assumptions a, b. Clearly $\mathcal{P}$ syntactically complies with Θ. However $\mathcal{P}$ does not semantically comply with Θ. To see this consider the possible world $\omega = \{\}$ (the only possible world of $\mathcal{P}$), clearly $\{x \in \Theta \mid \mathcal{F}_\omega \vdash_{cr} x\} = \{\theta_2\}$ but $\mathcal{F}_\omega \not\vdash_{sk} \theta_2$.*

However, syntactic compliance ensures a weaken version of semantic compliance defined as follows.

Definition 15. *We say that a PABA framework $\mathcal{P}$* ***semantically semi-complies*** *with a FoD Θ iff for each possible world ω and $\theta \in \Theta$, if $\mathcal{F}_\omega \vdash_{sk} \theta$ then $\{x \in \Theta \mid \mathcal{F}_\omega \vdash_{cr} x\} = \{\theta\}$.*

Basically $\mathcal{P}$ semantically semi-complies but not semantically complies with Θ if for some possible world ω and answer $\theta \in \Theta$, $\{x \in \Theta \mid \mathcal{F}_\omega \vdash_{cr} x\} = \{\theta\}$ but $\mathcal{F}_\omega \not\vdash_{sk} \theta$. The PABA framework in Example 7 falls into this case.

Lemma 6. *If a PABA framework $\mathcal{P}$ syntactically complies with Θ, then $\mathcal{P}$ semantically semi-complies with Θ.*

Obviously semantic semi-compliance could not ensure that $Bel_m(\theta) = Prob(\mathcal{P} \vdash_{sk} \theta)$. However it ensures a half of this equality as follows.

Lemma 7. *Suppose $\mathcal{P} \overset{\Theta}{\to} m$. If $\mathcal{P}$ semantically semi-complies with Θ then $\forall\theta \in \Theta$, $Bel_m(\theta) \geq Prob(\mathcal{P} \vdash_{sk} \theta)$.*

So a corollary of the above lemmas is that if $\mathcal{P} \xrightarrow{\Theta} m$ and $\mathcal{P}$ syntactically complies with Θ, then for any $\theta \in \Theta$, $Bel_m(\theta) \geq Prob(\mathcal{P} \vdash_{sk} \theta)$ and $Pl_m(\theta) = Prob(\mathcal{P} \vdash_{cr} \theta)$.

5. Composite PABA frameworks

In this section, we accumulate three presented techniques to propose so-called c-PABA which lends itself to a development tool for composite argumentation systems.

5.1. Structure and Semantics

A c-PABA framework contains components of two kinds: data-consuming and data-generating. The latter provides DST data which is consumed by the former.

Definition 16. *A **composite PABA (c-PABA) framework** is a structure of the form $\mathcal{S} = (\{(\Theta_i, S_i)\}_{i=1}^{k}, \mathcal{P})$ where $\mathcal{P}$, which is referred to as the **data-consuming component** of $\mathcal{S}$, is a PABA framework; $\mathcal{P}_i$, which is referred to as a **data-generating component** of $\mathcal{S}$, is a c-PABA framework; and Θ_i is a FoD.*

*The set of all mass functions $\{m_i \mid i \in \{1,\dots,k\}, S_i \xrightarrow{\Theta_i} m_i\}$ generated by all data-generating components is referred to as **the internal information flow** in $\mathcal{S}$.*

For example, a c-PABA framework $(\{\}, \mathcal{P})$, which shall be written as $\mathcal{P}$ for short, is just a PABA framework. In general, we want to see a c-PABA framework as the combination of its main module and its internal information flow. To ensure that this combination can be computed by the s-fusion operator and always results in a well-formed PABA framework, let's introduce a class of *well-formed* c-PABA frameworks.

Definition 17. *A c-PABA framework $\mathcal{S} = (\{(\Theta_i, S_i)\}_{i=1}^{k}, \mathcal{P})$ is said to be **well-formed** if for each $i, j \in \{1,\dots,k\}$, $\mathcal{P}$ syntactically complies with FoD Θ_i and S_i semantically complies with Θ_i; further either $\Theta_i = \Theta_j$ or $\Theta_i \cap \Theta_j = \emptyset$ for any $j \in \{1,\dots,k\}$.*

For example, c-PABA framework $\mathcal{S} = (\{(\Theta, PABA_{m_i})\}_{i=1}^{k}, (\emptyset, \emptyset, \mathcal{FD}_{\Theta}))$ representing a DST evidence base $\mathcal{M} = \{m_i\}_{i=1}^{k}$ over FoD Θ, is well-formed[6]. The following lemma follows directly from Lemma 2.

Lemma 8. *Let $\mathcal{S} = (\{(\Theta_i, S_i)\}_{i=1}^{k}, \mathcal{P})$ be a well-formed c-PABA framework with internal information flow $\mathcal{M}$, and $\{M_1, M_2, \dots, M_n\}$ be the partition on $\mathcal{M}$ such that M_i is a DST evidence base[7]. Then $\mathcal{P} \uplus M_1 \uplus \cdots \uplus M_n$ is a well-formed PABA framework.*

The above PABA framework $\mathcal{P} \uplus M_1 \uplus \cdots \uplus M_n$ will be referred to as the PABA representation of the given c-PABA framework $\mathcal{S}$ and denoted by $PABA_{\mathcal{S}}$. The semantics of $\mathcal{S}$ is then defined by that of $PABA_{\mathcal{S}}$, concretely:

Definition 18. *Let $\mathcal{S} = (\{(\Theta_i, S_i)\}_{i=1}^{k}, \mathcal{P})$ be a well-formed c-PABA framework and π is a proposition. Define $Prob(\mathcal{S} \vdash_s \pi) \triangleq Prob(PABA_{\mathcal{S}} \vdash_s \pi)$.*

Of course we will say that $\mathcal{S}$ generates a mass function m over FoD Θ if $PABA_{\mathcal{S}} \xrightarrow{\Theta} m$; $\mathcal{S}$ semantically/syntactically complies with Θ if so does $PABA_{\mathcal{S}}$; and so on.

[6] It is easy to see that the internal information flow of $\mathcal{S}$ coincides with $\mathcal{M}$ because $PABA_{m_i} \xrightarrow{\Theta} m_i$

[7] That is, the mass functions in M_i share the same FoD.

5.2. Two sample applications: DST-based data fusion and ML classifier combination

DST-based data fusion can be implemented by a simple c-PABA framework as follows.

Lemma 9. *Let $\mathcal{M} = \{m_i\}_{i=1}^k$ be a DST evidence base over FoD Θ. For any $X \subseteq \Theta$, $Bel_{\mathcal{M}}(X) = Prob(\mathcal{S} \vdash_{sk} X)$ and $Pl_{\mathcal{M}}(X) = Prob(\mathcal{S} \vdash_{cr} X)$ where $\mathcal{S}$ is the c-PABA framework $(\{(\Theta, PABA_{m_i})\}_{i=1}^k, (\emptyset, \emptyset, \mathcal{FD}_\Theta))$.*

ML classifier combination can be implemented by c-PABA as well. Note that a classifier is an algorithm that assigns to each input pattern x a single class from a set of classes $\Theta = \{\theta_1, \ldots, \theta_{|\Theta|}\}$ - which can be viewed as a FoD. In practice, however a classifier built by ML often returns a vector $[s_1, \ldots, s_{|\Theta|}]$ where s_i indicates some kind of confidence degree that x belongs to class θ_i. It is a common practice to test a ML classifier against test datasets, computing various performance indexes such as recognition rate r, substitution rate s and rejection rate $q = 1 - r - s$. Such indexes are then used to interpret what the classifier actually says. For example, in [16] Xu et al argue that $[s_1, \ldots, s_{|\Theta|}]$ should be interpreted as such a mass function m that: if $[s_1, \ldots, s_{|\Theta|}] = [0, \ldots, 0]$, then $m = \{(\Theta, 1)\}$; if $[s_1, \ldots, s_{|\Theta|}] = [0, \ldots, s_i = 1, \ldots, 0]$, then $m = \{(\{\theta_i\}, r), (\Theta - \{\theta_i\}, s), (\Theta, q)\}$. Now as different classifiers potentially offer complementary information about patterns to be classified, one wants to combine the outputs of multiple classifiers for the classification problem at hand. This combination problem is formalized as follows: given classifiers, $f_1, \ldots, f_k$, return a combined classifier f^* that for a given input x, assigns a class $\theta^* \in \Theta$ to x, by: 1) for each output vector $f_i(x)$, construct a mass function m_i; and 2) combine $m_1, \ldots, m_k$ to obtain one mass function m^* which then derives θ^*. Clearly both steps allow choices. Suppose that step (1) uses Xu et al's rule to compute m_i; and step (2) uses Smet's rule to compute m^* then returns $\theta^* = argmax_{\theta \in \Theta} Pl_{m^*}(\theta)$. The lemma below says that f^* can be implemented in c-PABA.

Lemma 10. *Suppose f^* combines classifiers $f_1, \ldots, f_k$ (by using Xu et al's rule, Smet's rule to arrive at a combined mass function m^*) where f_i has recognition/substitution/rejection rates $f_i.r, f_i.s, f_i.q$. Then*

$$f^*(x) \triangleq argmax_{\theta \in \Theta} Pl_{m^*}(\theta) = argmax_{\theta \in \Theta} Prob(\mathcal{S} \vdash_{cr} \theta)$$

where $\mathcal{S} = (\{(\Theta, \mathcal{P}_i)\}_{i=1}^k, (\emptyset, \emptyset, \mathcal{FD}_\Theta))$, with $\mathcal{P}_i$ being any PABA framework that generates m_i - the mass function that Xu et al's rule derives from the output vector $f_i(x) = [s_{i1}, \ldots, s_{i|\Theta|}]$ of f_i.

For example, $\mathcal{P}_i$ may be $(\mathcal{V}, Pr, \mathcal{OI}_\Theta \cup \{op([s_{i1}, \ldots, s_{i|\Theta|}]) \leftarrow)\})$ with $\mathcal{V} = \{reg, sub, rej\}$, $Pr = \{f_i.r : reg; f_i.s :: sub; f_i.q :: rej.\}$ (saying that the probabilities of random variables reg, sub, rej coincide with the recognition/substitution/rejection rates of f_i), and $\mathcal{OI}_\Theta$ is the ABA framework obtained from $\mathcal{FD}_\Theta$ by adding the following rules for each vector $[0, \ldots, s_i = 1, \ldots, 0]$:

- $\neg\theta_j \leftarrow reg, op([0, \ldots, s_i = 1, \ldots, 0])$ where $j \neq i$ representing that θ_i is the right class with probability equal the recognition rate.
- $\neg\theta_i \leftarrow sub, op([0, \ldots, s_i = 1, \ldots, 0])$ representing that $\Theta - \{\theta_i\}$ contains the right class with probability equal the substitution rate.

Note that $op([s_{i1}, \ldots, s_{i|\Theta|}]) \leftarrow$ is a just a fact encoding the output vector of f_i.

6. Conclusion and related work

Against two theoretical backdrops: PABA and DST, we present a development tool called c-PABA for composite argumentation systems with ML components. Demonstratively we use c-PABA to implement two key applications of DST: multi source data fusion and multi-classifier combination. To the best of our knowledge, the only work in the current literature that involves both DST and PABA is [8] which uses PABA to re-construct DST, but does not deal with composite argumentation systems as in the current paper. However there is a rich line of work combining DST and logic-based reasoning (but not necessarily argumentative). For example, in [15,2,14] the authors combine DST with deductive reasoning. [11] associate probability mass with formula and compute measures-like belief degrees of the reasoning with these formula. The notion of arguments of this work, however, is limited to conjunctions of literals. In [12] the authors define argumentation semantics for subjective logic, a logic that incorporates measures from DST. It is argued that reasoning systems in these above reasoning formalisms can be viewed as components in composite argumentation systems that our c-PABA proposal captures.

References

[1] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1):63 – 101, 1997.

[2] P. Chatalic, D. Dubois, and H. Prade. An approach to approximate reasoning based on the dempster rule of combination. *Int. J. Expert Syst.*, 1(1):67–85, September 1987.

[3] A. P. Dempster. Upper and Lower Probabilities Induced by a Multivalued Mapping. *The Annals of Mathematical Statistics*, 38(2):325 – 339, 1967.

[4] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321 – 357, 1995.

[5] Phan Minh Dung and Phan Minh Thang. Towards (probabilistic) argumentation for jury-based dispute resolution. In *COMMA 2010*, pages 171–182, 2010.

[6] Daan Fierens, Guy Van Den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401, 2015.

[7] Nguyen Duy Hung. Inference procedures and engine for probabilistic argumentation. *International Journal of Approximate Reasoning*, 90:163–191, 2017.

[8] Nguyen Duy Hung. Probabilistic assumption-based argumentation with dst evidence. In *IFSA-SCIS*, pages 1–6, 2017.

[9] Nguyen Duy Hung. Progressive inference algorithms for probabilistic argumentation. In *PRIMA 2018*, volume 11224, pages 371–386, 2018.

[10] Anthony Hunter. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximate Reasoning*, 54(1):47 – 81, 2013.

[11] J. Kohlas, D. Berzati, and R. Haenni. Probabilistic argumentation systems and abduction. *Annals of Mathematics and Artificial Intelligence*, 34(1):177–195, 2002.

[12] Nir Oren, Timothy J. Norman, and Alun Preece. Subjective logic and arguing with evidence. *Artificial Intelligence*, 171(10):838 – 854, 2007.

[13] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[14] Philippe Smets. Probability of deductibility and belief functions. In *ECSQARU*, pages 332–340, 1993.

[15] Yuqing Tang, Chung-Wei Hang, Simon Parsons, and Munindar P. Singh. Towards argumentation with symbolic dempster-shafer evidence. In *COMMA 2012*, pages 462–469.

[16] L. Xu, A. Krzyzak, and C.Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418–435, 1992.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220151

Automated Reasoning with Epistemic Graphs Using SAT Solvers

Anthony HUNTER
Department of Computer Science,
University College London, London, UK
(anthony.hunter@ucl.ac.uk)

Abstract. Epistemic graphs have been developed for modelling an agent's degree of belief in an argument and how belief in one argument may influence the belief in other arguments. These beliefs are represented by constraints on probability distributions. In this paper, we present a framework for reasoning with epistemic graphs that allows for beliefs for individual arguments to be determined given beliefs in some of the other arguments. We present and evaluate algorithms based on SAT solvers.

Keywords. Probabilistic argumentation; Argumentation algorithms; Bipolar argumentation.

1. Introduction

Epistemic graphs are a generalization of the epistemic approach to probabilistic argumentation [1]. In epistemic graphs, the graph is augmented with a set of constraints on probability distributions. These constraints restrict the belief we have in each argument and they capture how beliefs in arguments influence each other. The aim is that a set of constraints captures the subjective, and possibly imperfect way, that an agent views the beliefs in the arguments and their interactions. The graphs can model both attack and support (see for example Figure 1) as well as relations that are neither positive nor negative (see for example Figure 3) with the label denoting the type of influence (e.g. positive (supporting), negative (attacking), and mixed). Both the label and the constraints provide information about the argumentation. In this paper, we focus on the constraints.

There are some similarities between epistemic graphs and graded and ranking–based semantics proposed for a number of argumentation frameworks [2,3,4,5,6,7,8,9,10,11,12] but there are also substantial differences. Most assign a value in the unit interval to arguments without further clarification of the meaning of the number. Furthermore, many of the postulates in these approaches are not really applicable in the epistemic approach, even though they can be perfectly suitable in other scenarios (e.g. in the epistemic approach, an increase or decrease in beliefs in attackers (or supporters) does not necessarily invoke an decrease or increase in the belief of the target argument).

Epistemic graphs have some similarities with abstract dialectical frameworks (ADFs) [13] and weighted ADFs (WADFs) [14]. However, differences include epistemic graphs allow for a finer-grained probabilistic evaluation of arguments, allowing unattacked arguments to be disbelieved, and long-distance effects between arguments that do not have

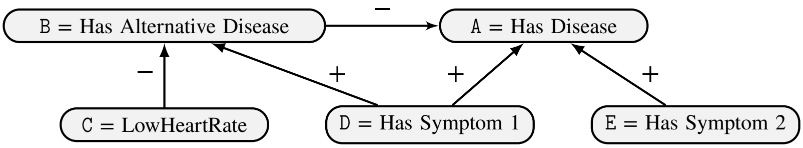


Figure 1. Example of an epistemic graph concerning diagnosis of a disease based on belief in symptoms and a differential diagnosis which in turn is based on a symptom and a test. The + (resp. −) label denote support (resp. attack) relations. Assume that if B is strongly believed, and D or E is strongly disbelieved, then A is strongly disbelieved, whereas if B is believed, and D or E is disbelieved, then A is disbelieved. Furthermore, if D and E are believed, then A is believed. These constraints could be reflected by the following formulae: $\varphi_1 : p(\mathtt{B}) > 0.8 \land p(\mathtt{D} \lor \mathtt{E}) < 0.2 \Rightarrow p(\mathtt{A}) < 0.8$; $\varphi_2 : p(\mathtt{B}) > 0.5 \land p(\mathtt{D} \lor \mathtt{E}) < 0.5 \Rightarrow p(\mathtt{A}) < 0.5$; $\varphi_3 : p(\mathtt{D} \land \mathtt{E}) > 0.5 \Rightarrow p(\mathtt{A}) > 0.5$; $\varphi_4 : p(\mathtt{C}) > 0.5 \Rightarrow p(\mathtt{B}) \leq 0.5$; And $\varphi_5 : p(\mathtt{C}) \leq 0.5 \Rightarrow p(\mathtt{B}) > 0.5$.

an arc connecting them. For more detailed comparison of ADFs with epigraphs, see [1]. Also see [1] for coverage of substantial differences with Bayesian networks.

In previous work, we presented a model-based theorem prover which can be used to check whether one constraint entails another that was based on enumerating all the models [15], and an approach based on calculating probability distributions satisfying the constraints using numerical optimization methods [16]. These methods only work for small numbers of arguments. Yet, there is a need for a scalable theorem prover that allows us to query the constraints of an epistemic graph in order to draw inferences about the belief in specific arguments. To address this need, we present a new proposal in this paper for taking a knowledgebase of constraints and optionally further assumptions, and drawing inferences from them. The approach involves representing the constraints as clauses, and then uses an off-the-shelf SAT solver (see [17] for an introduction to SAT solvers). We do this by defining a set of axioms, which we call a completion, of an epistemic graph which we add to the constraints when querying the SAT solver. By assuming these axioms, we can obtain a sound and complete inferencing algorithm.

2. Epistemic Graphs: A Simplified Version

In this paper, we present a simpler version of epistemic graphs than presented in [1]. Let $\mathcal{G}$ denote a graph where $\mathsf{Nodes}(\mathcal{G})$ be the set of nodes in $\mathcal{G}$, and $\mathsf{Arcs}(\mathcal{G})$ be the set of arcs in $\mathcal{G}$. We consider a probability distribution $P : \wp(\mathsf{Nodes}(G)) \rightarrow [0,1]$ as being a probability assignment to each subset of the set of arguments such that this sums to 1 (i.e. $\sum_{\Gamma \subseteq \mathsf{Nodes}(G)} P(\Gamma) = 1$). We denote the set of all probability distributions on $\mathsf{Nodes}(\mathcal{G})$ by $\mathsf{Dist}(\mathcal{G})$. The constraints restrict the set of probability distributions that satisfy the arguments (as we explain in the rest of this subsection).

Based on a given graph, we can now define the epistemic language. In this paper, we will only consider a sublanguage of that defined in [1]. The **simplified epistemic language** based on graph $\mathcal{G}$ is defined as follows: an **epistemic atom** is of the form $p(A)\#x$ where $\# \in \{<, \leq, =, \geq, >\}$, $x \in [0,1]$ and $A \in \mathsf{Nodes}(\mathcal{G})$; and an **epistemic formula** is a Boolean combination of epistemic atoms. For example, from the epistemic atoms $p(\mathtt{A}) \leq 0.5$ and $p(\mathtt{B}) \geq 0.5$, an epistemic formula is $p(\mathtt{A}) \leq 0.5 \rightarrow p(\mathtt{B}) \geq 0.5$.

The semantics for constraints come from probability distributions $P \in \mathsf{Dist}(\mathcal{G})$. Each $\Gamma \subseteq \mathsf{Nodes}(\mathcal{G})$ corresponds to a possible world where the arguments in Γ are true. The **probability of an argument** being acceptable is defined as the sum of the probabilities of

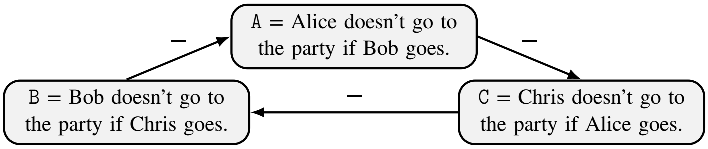

A = Alice doesn't go to the party if Bob goes.
B = Bob doesn't go to the party if Chris goes.
C = Chris doesn't go to the party if Alice goes.
−
−
−

Figure 2. The epistemic graph has constraints $\{p(\mathtt{B}) > 0.5 \Rightarrow p(\mathtt{A}) \leq 0.5,\ p(\mathtt{C}) > 0.5 \Rightarrow p(\mathtt{B}) \leq 0.5,\ p(\mathtt{A}) > 0.5 \Rightarrow p(\mathtt{C}) \leq 0.5\}$. Given these constraints, we see that at most one argument can be believed.

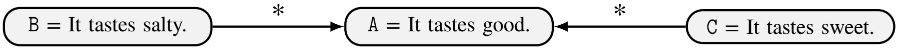


Figure 3. The epistemic graph has constraints $\{p(\mathtt{B}) > 0.5 \wedge p(\mathtt{C}) \leq 0.5 \Rightarrow p(\mathtt{A}) > 0.5,\ p(\mathtt{C}) > 0.5 \wedge p(\mathtt{B}) \leq 0.5 \Rightarrow p(\mathtt{A}) > 0.5,\ p(\mathtt{B}) > 0.5 \wedge p(\mathtt{C}) > 0.5 \Rightarrow p(\mathtt{A}) \leq 0.5\}$. Given these constraints, the influence of B and C on A is not simply a positive or a negative one. Consider some an item of food. If it is believed to be tasting salty and not believed to be tasting sweet, or it is not believed to be tasting salty and believed to be tasting sweet, then it is believed to be good tasting, and if it is believed to be tasting salty and believed to be tasting sweet, then it is not believed to be good tasting.

the worlds containing it: $P(A) = \sum_{\Gamma \subseteq \mathsf{Nodes}(\mathcal{G}) \text{ s.t. } A \in \Gamma} P(\Gamma)$. We say that an agent believes an argument A to be acceptable if $P(A) > 0.5$, disbelieves A to be acceptable if $P(A) < 0.5$, and neither believes nor disbelieves A to be acceptable when $P(A) = 0.5$.

For an epistemic atom $p(A)\#v$, where $\# \in \{<, \leq, =, \geq, >\}$, the **satisfying distributions**, or equivalently **models**, of $p(A)\#v$ are defined as $\mathsf{Sat}(p(A)\#v) = \{P' \in \mathsf{Dist}(\mathcal{G}) \mid P'(A)\#v\}$. The set of satisfying distributions for a given epistemic formula is as follows where ϕ and ψ are epistemic formulae: $\mathsf{Sat}(\phi \wedge \psi) = \mathsf{Sat}(\phi) \cap \mathsf{Sat}(\psi)$; $\mathsf{Sat}(\phi \vee \psi) = \mathsf{Sat}(\phi) \cup \mathsf{Sat}(\psi)$; and $\mathsf{Sat}(\neg\phi) = \mathsf{Sat}(\top) \setminus \mathsf{Sat}(\phi)$. For a set of epistemic formulae $\Phi = \{\phi_1, \ldots, \phi_n\}$, the set of satisfying distributions is $\mathsf{Sat}(\Phi) = \mathsf{Sat}(\phi_1) \cap \ldots \cap \mathsf{Sat}(\phi_n)$. A set of epistemic formulae is consistent iff its set of models is non-empty.

Example 1. *Consider the set of formulae* $\{p(\mathtt{A}) > 0.5 \rightarrow \neg(p(\mathtt{B}) > 0.5), p(\mathtt{A}) = 0 \vee p(\mathtt{A}) = 0.5 \vee p(\mathtt{A}) = 1, p(\mathtt{B}) = 0 \vee p(\mathtt{B}) = 0.5 \vee p(\mathtt{B}) = 1\}$. *Examples of probability distributions that satisfy the set include* P_1 *s.t.* $P_1(\emptyset) = 1$, P_2 *s.t.* $P_2(\emptyset) = P_2(\{\mathtt{A}\}) = 0.5$, P_3 *s.t.* $P_3(\{\mathtt{A}\}) = 1$, *or* P_4 *s.t.* $P_4(\{\mathtt{A}\}) = P_3(\{\mathtt{A},\mathtt{B}\}) = 0.5$ *(omitted sets are assigned* 0*). The probability distribution* P_5 *s.t.* $P_5(\{\mathtt{A},\mathtt{B}\}) = 1$ *does not satisfy the formula.*

For the arguments in graph $\mathcal{G}$, and probability function P, an **epistemic extension** is the set $\{A \in \mathsf{Nodes}(\mathcal{G}) \mid P(A) > 0.5\}$. So the extension is determined from the probability function rather the structure of the graph. For example, for Figure 2, if $P(\mathtt{A}) = 0.1$, $P(\mathtt{B}) = 0.9$, and $P(\mathtt{C}) = 0.1$, then the epistemic extension is $\{\mathtt{B}\}$.

We define an **entailment relation**, denoted $\models$, as follows, where Γ is a set of epistemic formulae, and ϕ is an epistemic formula: $\Gamma \models \phi$ iff $\mathsf{Sat}(\phi) \subseteq \mathsf{Sat}(\Gamma)$

Example 2. *Let* $\Gamma = \{p(\mathtt{C}) > 0.9, p(\mathtt{B}) = 0.3, p(\mathtt{C}) \geq 0.8 \wedge p(\mathtt{B}) < 0.6 \rightarrow p(\mathtt{A}) > 0.5\}$. *Hence,* $\mathsf{Sat}(p(\mathtt{A}) \geq 0.5) \subseteq \mathsf{Sat}(\Gamma)$*, and so* $\Gamma \models p(\mathtt{A}) \geq 0.5$.

The simplified epistemic language does not incorporate features of the full epistemic language (as presented in [1]) such as terms that are Boolean combinations of arguments (e.g. $P(\mathtt{B} \vee \mathtt{C}) > 0.6$ which says that the probability argument B or argument C is greater than 0.6) or summation of probability values (such as $P(\mathtt{A})+P(\mathtt{B}) \leq 1$ which says that the sum of probability A and probability B is less than or equal to 1). Nonetheless, the restricted epistemic language is a useful sublanguage and it simplifies the presentation and evaluation in this paper.

An **epistemic constraint** is an epistemic formula $\psi \in \mathsf{Formulae}(\mathcal{G})$. An **epistemic graph** is a tuple $(\mathcal{G}, \mathcal{L}, C)$ where $(\mathcal{G}, \mathcal{L})$ is a labelled graph, and $C \subseteq \mathsf{Formulae}(\mathcal{G})$ is a set of epistemic constraints associated with the graph.

In general, the graph (and its labellings) is not necessarily induced by the constraints and therefore it contains additional information. The actual direction of the edges in the graph is also not necessarily derivable from C. For example, if we have two arguments A and B connected by an edge, a constraint of the form $p(\mathtt{A}) < 0.5 \vee p(\mathtt{B}) < 0.5$ would not tell us the direction of this edge. The constraints may also involve unrelated arguments, similar to [18], e.g. $\neg p(\mathtt{C}) > 0.5 \vee \neg p(\mathtt{D}) > 0.5$ when there is no arc between C and D. So the assignment of a label to an arc (by the $\mathcal{L}$ function) is an extra piece of information. The assignment is intended to denote the kind of influence of the source node on the target node. If we use the labels $\{+, *, -\}$, then the assignment of $+$ is intended to denote a form of support, the assignment of $-$ is intended to denote a form of attack, and $*$ is intended to denote an influence that is neither support nor attack. So $*$ could denote that under some conditions behaves as an attack and under some conditions behaves as a support as illustrated in the arc in Example 3. As investigated in [1], there are various ways that we can formalize the relationships between labels and constraints. We will not consider labels further in this paper, and we will focus on the constraints.

For this paper, we also require the notion of an **observation** which is an epistemic formula. The difference between constraints and observations is that we assume the constraints always hold, whereas observations only hold in some situations or for some periods. For example, if a debater uses an epistemic graph to model what opponents believe, the observations would be specific beliefs for a specific opponent.

Example 3. *Returning to Figure 2, suppose we have the observation* $p(\mathtt{C}) \geq 0.8$*, then we want to draw the conclusions* $p(\mathtt{B}) \leq 0.5$ *and* $p(\mathtt{A}) \leq 0.5$.

Example 4. *Returning to Figure 3, suppose we have the observations* $p(\mathtt{B}) = 0.7$ *and* $p(\mathtt{C}) = 0.2$*, then we want to draw the conclusion* $p(\mathtt{A}) > 0.5$*. Or suppose we have the observations* $p(\mathtt{B}) > 0.7$ *and* $p(\mathtt{C}) \geq 0.8$*, then we want to draw the conclusion* $p(\mathtt{A}) \leq 0.5$.

In the following, we will use the term **knowledgebase**, denoted $\mathcal{K}$, to refer to the union of a set of constraints and a set of observations.

3. Reasoning with Epistemic Graphs

In this paper, our approach to inference with constraints and observations is to use SAT solvers. So we will need to represent constraints and observations as clauses (i.e. a disjunction of literals). Any formula of propositional logic (and similarly any epistemic formula) can be rewritten in conjunction normal form, and then conjunction elimination applied, to obtain a set of clauses that are logically equivalent to the original epistemic formula. So we do not lose any expressibility if we represent our epistemic formulae as clauses. Note, clauses can be rewritten as implications. So $\beta_1 \vee \ldots \vee \beta_{n-1} \vee \beta_n$ can be represented as $\neg\beta_1 \wedge \ldots \wedge \neg\beta_{n-1} \rightarrow \beta_n$.

We will also restrict the probability values that the formulae can take by using a **restricted value set**, denoted Π, which is a subset of the unit interval such that $0, 1 \in \Pi$, and for all $x, y \in \Pi$, if $x + y \in [0, 1]$, then $x + y \in \Pi$, and if $x - y \in [0, 1]$, then $x - y \in \Pi$. For example, $\{0, 0.5, 1\}$ and $\{0, 0.1, 0.2, \ldots, 0.9, 1\}$ are restricted value sets. In this paper, we will assume $\Pi = \{0, 0.1, 0.2, \ldots, 0.9, 1\}$ unless explicitly stated otherwise.

Definition 1. *The* **restricted language** *based on graph $\mathcal{G}$ and a restricted value set* Π *is defined as follows: a* **restricted atom** *of the form $p(A)\#x$ where $\# \in \{<, \leq, =, \geq, >\}$, $x \in \Pi$ and $A \in \mathsf{Nodes}(\mathcal{G})$; a* **restricted clause** *of the form $\beta_1 \vee \ldots \vee \beta_n \vee \beta_{n+1}$ where each β_i in $\{\beta_1, \ldots, \beta_n, \beta_{n+1}\}$ is a* **restricted literal** *(i.e. a restricted atom, or its negation).*

Example 5. *Let* $\Pi = \{0, 0.5, 1\}$. *In the restricted language w.r.t.* Π, *we can only have atoms of the form $p(\mathtt{A})\#0$, $p(\mathtt{A})\#0.5$, and $p(\mathtt{A})\#1$, where $\mathtt{A} \in \mathsf{Nodes}(\mathcal{G})$ and $\# \in \{<, \leq, =, \geq, >\}$. From these atoms we compose epistemic formulae, using the Boolean connectives, such as $p(\mathtt{A}) \leq 0.5 \to \neg(p(\mathtt{B}) \geq 0.5)$.*

We also require some subsidiary definitions. Literals ϕ and ψ are **logically complementary** iff ϕ is $\neg\psi$ or ψ is $\neg\phi$. (e.g. the literals $P(\mathtt{A}) > 0.8$ and $\neg(P(\mathtt{A}) > 0.8)$ are logically complementary); And the literals ϕ and ψ are **probabilistically complementary** iff $\mathsf{Sat}(\phi) \cap \mathsf{Sat}(\psi) = \emptyset$ and ϕ and ψ are not logically complementary (e.g. $P(\mathtt{A}) > 0.8$ and $P(\mathtt{A}) < 0.8$ are probabilistically complementary, and when $\Pi = \{0, 0.1, \ldots, 0.9, 1.0\}$, $P(\mathtt{A}) > 0.9$ and $\neg(P(\mathtt{A}) = 1)$ are probabilistically complementary).

To reason with a knowledgebase (i.e. a set of constraints and observations), we propose a proof theoretic approach based on adding extra axioms to the knowledgebase to capture the implicit probabilistic information that is required. For this we introduce the notion of equality completion to reduce our knowledgebase and query to disjunctions involving only equality and the restricted value set Π. For example the atom $p(A) > 0.6$ implies $p(A)$ is one of 0.7, 0.8, 0.9, or 1 as captured by the following clause.

$$\neg(p(\mathtt{A}) > 0.6) \vee p(\mathtt{A}) = 0.7 \vee p(\mathtt{A}) = 0.8 \vee p(\mathtt{A}) = 0.9 \vee p(\mathtt{A}) = 1.0$$

In the following definition of completion, we also include the constraint that an argument cannot have two values. So for all arguments $\mathtt{A}$, for all $x, y \in \{0, 0.1, 0.2, \ldots, 0.9, 1\}$, s.t. $x \neq y$, $\neg(p(\mathtt{A}) = x) \vee \neg(p(\mathtt{A}) = y)$.

Definition 2. *For a graph $\mathcal{G}$, the set of* **completion clauses** *is the following set of clauses*

$$\mathsf{Complete}(\mathcal{G}) = \bigcup_{A \in \mathsf{Nodes}(\mathcal{G})} \left(\left(\bigcup_{k \in \{1, \ldots, 8\}} \mathsf{C}_k(A) \right) \cup \mathsf{Exclusion}(A) \right)$$

where $\mathsf{Exclusion}(A) = \{\neg(p(A) = x) \vee \neg(p(A) = y) \mid x \neq y\}$ *and*

$$\begin{aligned}
\mathsf{C}_1(A) &= \{p(A) > x \vee p(A) = y_1 \vee \ldots \vee p(A) = y_n \mid x \leq y_1, \ldots, y_n\}\\
\mathsf{C}_2(A) &= \{\neg(p(A) > x) \vee p(A) = y_1 \vee \ldots \vee p(A) = y_n \mid x > y_1, \ldots, y_n\}\\
\mathsf{C}_3(A) &= \{p(A) < x \vee p(A) = y_1 \vee \ldots \vee p(A) = y_n \mid x \geq y_1, \ldots, y_n\}\\
\mathsf{C}_4(A) &= \{\neg(p(A) < x) \vee p(A) = y_1 \vee \ldots \vee p(A) = y_n \mid x < y_1, \ldots, y_n\}\\
\mathsf{C}_5(A) &= \{\neg(p(A) \leq x) \vee p(A) = y_1 \vee \ldots \vee p(A) = y_n \mid x \leq y_1, \ldots, y_n\}\\
\mathsf{C}_6(A) &= \{p(A) \leq x \vee p(A) = y_1 \vee \ldots \vee p(A) = y_n \mid x > y_1, \ldots, y_n\}\\
\mathsf{C}_7(A) &= \{\neg(p(A) \geq x) \vee p(A) = y_1 \vee \ldots \vee p(A) = y_n \mid x \geq y_1, \ldots, y_n\}\\
\mathsf{C}_8(A) &= \{p(A) \geq x \vee p(A) = y_1 \vee \ldots \vee p(A) = y_n \mid x < y_1, \ldots, y_n\}
\end{aligned}$$

The size of $\mathsf{Complete}(\mathcal{G})$ is a linear function of the number of arguments in the graph $\mathcal{G}$, as there are 198 axioms in $\mathsf{Complete}(\mathcal{G})$ per argument.

Proposition 1. *If* $|\mathsf{Nodes}(\mathcal{G})| = n$, *then* $|\mathsf{Complete}(\mathcal{G})| = 198n$.

Proof. For each argument in $A \in \mathsf{Nodes}(\mathcal{G})$, there is 11 axioms for each of Com_1 to Com_8 (there are 11 axioms since there is one axiom per value of x), and there are 110 exclusion axioms (since, for $\neg(p(A) = x) \vee \neg(p(A) = y)$, there are 11 choices for x and therefore 10 choices for y, which is 110 choices), giving a total of 198 axioms per argument. □

The axioms given in the completion are sound. In other words, they are satisfied by all probability distributions, and are therefore entailed by any knowledgebase.

In the following definition, we present the resolution proof rule as part of the resolution proof relation. This proof rule takes a pair of clauses where one has a disjunct, and the other has a disjunct that is its negation, and returns a clause where the disjuncts are all the disjuncts from the original clauses except the disjunct in the first clauses that is negated in the second clause.

Definition 3. *Let ϕ and ϕ' be clauses where ϕ is of the form $\alpha \vee \beta$ and ϕ' is of the form $\gamma \vee \delta$, and α and γ are logically complementary literals (i.e. α is $\neg\gamma$ or $\neg\alpha$ is γ), then $\beta \vee \delta$ is a* **resolvent** *of ϕ and ϕ'. The* **resolution proof relation**, *denoted $\vdash_{\mathsf{resolution}}$, is defined as follows where Δ is a set of clauses and ψ is a clause where the proof rules are: (1) Resolution; (2) Reflexivity; (3) Associativity; and (4) Contradiction.*

1 $\Delta \vdash_{\mathsf{resolution}} \beta \vee \delta$ *if* $\Delta \vdash_{\mathsf{resolution}} \alpha \vee \beta$ & $\Delta \vdash_{\mathsf{resolution}} \gamma \vee \delta$ & α *is* $\neg\gamma$
2 $\Delta \vdash_{\mathsf{resolution}} \phi$ *if* $\phi \in \Delta$
3 $\Delta \vdash_{\mathsf{resolution}} \alpha_1 \vee \ldots \vee \alpha_m$ *if* $\Delta \vdash_{\mathsf{resolution}} \beta_1 \vee \ldots \vee \beta_n$ & $\{\alpha_1, \ldots, \alpha_m\} = \{\beta_1, \ldots, \beta_n\}$
4 $\Delta \vdash_{\mathsf{resolution}} \bot$ *if* $\Delta \vdash_{\mathsf{resolution}} \phi$ & $\Delta \vdash_{\mathsf{resolution}} \neg\phi$

We now consider resolution with a knowledgebase and completion. Consider two clauses and two literals (one in each clause) that are either logically complementary or probabilistically complementary. For entailment, there is no probability distribution that satisfies both literals, and so the inference follows. In contrast, the resolution proof rule only deals with logically complementary literals, and so the completion is required to treat probabilistically complementary literals as logically complementary literals, and thereby obtain the inference. We illustrate this in the following example.

Example 6. *Consider* $\phi_1 = p(\mathtt{A}) > 0.8 \vee p(\mathtt{B}) > 0.5$ *and* $\phi_2 = p(\mathtt{A}) < 0.2 \vee p(\mathtt{B}) > 0.5$. *Clearly,* $p(\mathtt{A}) > 0.8$ *and* $p(\mathtt{A}) < 0.2$ *are probabilistically complementary literals, and that* $\{\phi_1, \phi_2\} \vDash p(\mathtt{B}) > 0.5$ *holds. The following axioms are from the completion.*

$$\pi_1 = \neg(p(\mathtt{A}) > 0.8) \vee p(\mathtt{A}) = 0.9 \vee p(\mathtt{A}) = 1 \qquad \pi_4 = \neg(p(\mathtt{A}) = 1) \vee \neg(p(\mathtt{A}) = 0)$$
$$\pi_2 = \neg(p(\mathtt{A}) < 0.2) \vee p(\mathtt{A}) = 0 \vee p(\mathtt{A}) = 0.1 \qquad \pi_5 = \neg(p(\mathtt{A}) = 0.9) \vee \neg(p(\mathtt{A}) = 0.1)$$
$$\pi_3 = \neg(p(\mathtt{A}) = 0.9) \vee \neg(p(\mathtt{A}) = 0) \qquad \pi_6 = \neg(p(\mathtt{A}) = 1) \vee \neg(p(\mathtt{A}) = 0.1)$$

We now show that $p(\mathtt{B}) > 0.5$ *can be obtained using the resolution proof relation with the completion of the knowledge. We use the names of clauses rather than the clauses in the premises to save space. The name of each clause generated by resolution is given on the right after the clause.*

1 $\{\phi_1, \pi_1\} \vdash_{\mathsf{resolution}} P(\mathtt{A}) = 0.9 \vee P(\mathtt{A}) = 1 \vee P(\mathtt{B}) > 0.5$ (ω_1)

2 $\{\phi_2, \pi_2\} \vdash_{\mathsf{resolution}} P(\mathtt{A}) = 0 \vee P(\mathtt{A}) = 0.1 \vee P(\mathtt{B}) > 0.5$ (ω_2)

3 $\{\omega_1, \pi_3\} \vdash_{\mathsf{resolution}} \neg(P(\mathtt{A}) = 0) \vee P(\mathtt{A}) = 1 \vee P(\mathtt{B}) > 0.5$ (ω_3)

4 $\{\omega_3, \pi_4\} \vdash_{\mathsf{resolution}} \neg(P(\mathtt{A}) = 0) \vee P(\mathtt{B}) > 0.5$ (ω_4)

5 $\{\omega_1, \pi_5\} \vdash_{\mathsf{resolution}} \neg(P(\mathtt{A}) = 0.1) \vee P(\mathtt{A}) = 1 \vee P(\mathtt{B}) > 0.5$ (ω_5)

6 $\{\omega_5, \pi_6\} \vdash_{\mathsf{resolution}} \neg(P(\mathtt{A}) = 0.1) \vee P(\mathtt{B}) > 0.5$ (ω_6)

7 $\{\omega_2, \omega_4\} \vdash_{\mathsf{resolution}} P(\mathtt{A}) = 0.1 \vee P(\mathtt{B}) > 0.5$ (ω_7)

8 $\{\omega_6, \omega_7\} \vdash_{\mathsf{resolution}} P(\mathtt{B}) > 0.5$ (ω_8)

In the following lemma, we generalize the above example by showing that if a clause is entailed by a pair of clauses, then that inference can be obtained from the completion of the clauses using only the resolution proof rule.

Lemma 1. *For graph $\mathcal{G}$, if ϕ, ϕ', ψ are clauses where ϕ is of the form $\alpha_1 \vee \ldots \vee \alpha_n$, ϕ' is of the form $\beta_1 \vee \ldots \vee \beta_m$, ψ is of the form $\alpha_1 \vee \ldots \vee \alpha_{n-1} \vee \beta_1 \vee \ldots \vee \beta_{m-1}$, and $\{\phi, \phi'\} \models \psi$, then $\{\phi, \phi'\} \cup \mathsf{Complete}(\mathcal{G}) \vdash_{\mathsf{resolution}} \psi$.*

Proof. Assume $\{\phi, \phi'\} \models \psi$. So for all $P \in \mathsf{Sat}(\{\phi, \phi'\})$, $P \not\models \alpha_n$ or $P \not\models \beta_m$. So either α_n and β_m are logically complementary literals (i.e. syntactically, α_n is $\neg\beta_m$ or $\neg\alpha_n$ is β_m) or α_n and β_m are probabilistically complementary literals (i.e. α_n is of the form $p(A_1)\#_1 v_1$ and β_m is of the form $p(A_2)\#_2 v_2$ and there is no assignment for w_1 and w_2 where $P(A_1) = w_1$ and $P(A_2) = w_2$ that would satisfy α_n and β_m). In the case that α_n and β_m are logically complementary literals, then $\{\phi, \phi'\} \vdash_{\mathsf{resolution}} \psi$ holds, and hence $\{\phi, \phi'\} \cup \mathsf{Complete}(\mathcal{G}) \vdash_{\mathsf{resolution}} \psi$ holds. In the case that α_n and β_m are probabilistically complementary literals, then the disjunct α_n in ϕ is resolved with a completion axiom and so exchanged for a disjunction of $p(A_1) = y_1 \vee \ldots \vee p(A_1) = y_n$, and the disjunct β_n in ϕ' is resolved with a completion axiom and so exchanged for a disjunction of $p(A_2) = y'_1 \vee \ldots \vee p(A_2) = y'_n$. So together with the exclusion axioms, there is no assignment for w_1 and w_2 in $p(A_1) = w_1$ and $p(A_2) = w_2$ that would satisfy $p(A_1) = y_1 \vee \ldots \vee p(A_1) = y_n$ and $p(A_2) = y'_1 \vee \ldots \vee p(A_2) = y'_n$. So each of these incompatible assignments is removed by resolution until none of them remain. So via a number of resolution steps, $\{\phi, \phi'\} \cup \mathsf{Complete}(\mathcal{G}) \vdash_{\mathsf{resolution}} \psi$. □

The following correctness result shows that a literal α is entailed if and only if the negation of the query together with the knowledgebase and completion results in a contradiction using the resolution consequence relation

Proposition 2. *For all epistemic graphs $(\mathcal{G}, \mathcal{L}, \mathcal{C})$, and literals α, $\mathcal{C} \models \alpha$ iff $\mathcal{C} \cup \mathsf{Complete}(\mathcal{G}) \cup \{\neg\alpha\} \vdash_{\mathsf{resolution}} \bot$.*

Proof. ($\Rightarrow$) Assume $\mathcal{C} \models \alpha$. Therefore $\mathsf{Sat}(\mathcal{C} \cup \{\neg\alpha\}) = \emptyset$. Therefore there is a subset $\Gamma \subseteq \mathcal{C} \cup \{\neg\alpha\}$ such that $\mathsf{Sat}(\Gamma) = \emptyset$ and for all $\Gamma' \subseteq \Gamma$, $\mathsf{Sat}(\Gamma') \neq \emptyset$. So for all $\phi \in \Gamma$, and for all $\alpha \in \mathsf{Disjuncts}(\phi)$, $\Gamma \setminus \{\phi\} \models \neg\alpha$. Moreover, for all $\phi, \phi' \in \Gamma$, and for all ψ such that ψ is a resolvent of ϕ and ϕ', $\Gamma \vdash \psi$, and by Lemma 1, $\Gamma \cup \mathsf{Complete}(\mathcal{G}) \vdash_{\mathsf{resolution}} \psi$. Since $\mathsf{Sat}(\Gamma) = \emptyset$, $\Gamma \vdash \bot$, and by Lemma 1, $\Gamma \cup \mathsf{Complete}(\mathcal{G}) \vdash_{\mathsf{resolution}} \bot$. So $\mathcal{C} \cup \mathsf{Complete}(\mathcal{G}) \cup \{\neg\alpha\} \vdash_{\mathsf{resolution}} \bot$. ($\Rightarrow$) Assume $\mathcal{C} \cup \mathsf{Complete}(\mathcal{G}) \cup \{\neg\alpha\} \vdash_{\mathsf{resolution}} \bot$. So $\mathsf{Sat}(\mathcal{C} \cup \mathsf{Complete}(\mathcal{G}) \cup \{\neg\alpha\}) = \emptyset$. Since all $\delta \in \mathsf{Complete}(\mathcal{G})$ are satisfied by all $P \in \mathsf{Dist}(\mathcal{G})$ (i.e. for all $P \in \mathsf{Dist}(\mathcal{G})$, $P \models \delta$), we have $\mathsf{Sat}(\mathsf{Complete}(\mathcal{G})) = \mathsf{Dist}(\mathcal{G})$. Therefore, $\mathsf{Sat}(\mathcal{C} \cup \{\neg\alpha\}) = \emptyset$. Hence, $\mathcal{C} \models \alpha$ holds. □

Algorithm 1 Clausal inference for knowledgebase $\mathcal{K}$, query α, and graph $\mathcal{G}$

```
function INFERENCE($\mathcal{K}$,$\alpha$,$\mathcal{G}$ )
    if $\alpha$ is a positive literal then
        return NOT SAT($\mathcal{K} \cup \mathsf{Complete}(\mathcal{G}) \cup \{\neg\alpha\}$)
    else
        return NOT SAT($\mathcal{K} \cup \mathsf{Complete}(\mathcal{G}) \cup \{\beta\}$) where $\alpha$ is of the form $\neg\beta$
```

Algorithm 2 Bounds for argument A w.r.t knowledgebase $\mathcal{K}$, graph $\mathcal{G}$, and increments μ.

```
function TIGHTINFERENCE($\mathcal{K}$,$A$,$\mu$,$\mathcal{G}$)
    $n = 0$
    while INFERENCE($\mathcal{K}, p(A) \geq n), \mathcal{G}$ do
        $n = n + \mu$
    $m = 1$
    while INFERENCE($\mathcal{K}, p(A) \leq m), \mathcal{G}$ do
        $m = m - \mu$
    return $(n, m)$
```

4. Algorithms

The inference algorithm (Algorithm 1) calls the SAT solver with a knowledgebase, and its completion, plus the negation of the query. If the SAT solver returns True, then the set of formulae is consistent, and hence the query does not follow from the premises, whereas if the SAT solver returns False, then the set of formulae is inconsistent, and hence the query does follow from the premises.

Proposition 3. *For a knowledgebase $\mathcal{K}$, and restricted literal α,* INFERENCE$(\mathcal{K}, \alpha, \mathcal{G})$ = True *iff $\mathcal{K} \vDash \alpha$.*

Proof. INFERENCE$(\mathcal{K}, \alpha, \mathcal{G})$ = True iff $\mathcal{K} \cup \mathsf{Complete}(\mathcal{G}) \cup \{\neg\alpha\} \vdash_{SAT} \perp$ iff $\mathcal{K} \vDash \alpha$. □

We also give an algorithm for obtaining bounds on a query (Algorithm 2). It obtains the tightest bounds $n, m \in \Pi$ such that $\mathcal{K} \vDash p(A) \geq n$ and $\mathcal{K} \vDash p(A) \leq m$ hold. The parameter μ specifies the restricted value set. For example, $\mu = 0.5$ when $\Pi = \{0, 0.5, 1\}$ and $\mu = 0.1$ when $\Pi = \{0, 0.1, 0.2, \ldots, 0.9, 1\}$.

Example 7. *Given the constraints $\{p(\mathtt{A}) \geq 0.4), p(\mathtt{A}) < 0.7 \vee p(\mathtt{B}) < 0.5\}$ and the observations $\{p(\mathtt{B}) \geq 0.5\}$, we obtain $(0.4, 0.6)$ from Algorithm 2 (i.e. 0.4 as the lower bound for $p(\mathtt{A})$ and 0.6 as the upper bound for $p(\mathtt{A})$).*

The algorithms (i.e. Algorithms 1 and 2) were implemented on Python. The implementation[1] uses the PySAT implementation [19] that incorporates SAT solvers such as Glucose3. The implementation includes code to randomly generate sets of epistemic constraints and queries. For a given number of arguments, and an upper limit on the number of disjuncts in each clause, the code randomly selects the argument, comparator and probability value for each atom in the clause. Each query is generated in the same way.

[1] http://www0.cs.ucl.ac.uk/staff/a.hunter/papers/episat.zip

	(2,10)	(2,100)	(4,10)	(4,100)	(6,10)	(6,100)
25	0.22	0.27	0.44	0.18	0.18	0.45
50	0.40	0.43	0.87	0.38	0.39	0.82
75	0.65	0.62	0.88	0.63	0.65	0.88
100	0.92	0.87	0.96	0.93	0.90	0.94
125	1.19	1.17	1.17	1.19	1.17	1.28
150	1.56	1.52	1.50	1.42	1.52	1.56
175	1.82	2.15	2.38	1.87	1.81	3.05
200	2.10	1.98	3.76	2.19	2.12	2.67
225	2.46	2.48	2.52	2.52	2.55	2.59
250	3.02	4.17	3.03	3.04	2.82	3.04

Table 1. Experiments with the INFERENCE algorithm (Algorithm 1). Each column is for a pair (d,c) where d is the upper limit of disjuncts (taking the value 2, 4, or 6 disjuncts) and c is cardinality of knowledgebase (taking the value of 10 or 100 clauses). Each row is the number of arguments in the range 25 to 250. For each combination of column and row, we obtained the average time taken (seconds) obtained over 10 runs.

Combination	$(a = 10, c = 10)$	$(a = 20, c = 20)$	$(a = 30, c = 30)$	$(a = 40, c = 40)$
Average time	1.47	3.25	5.61	7.55

Table 2. For each combination, where a is the number of arguments, and c is the number of clauses, we obtained the average time taken (seconds) obtained over 20 runs for each number of arguments.

The main purpose of the evaluation was to determine how the inference algorithm performs with the number of arguments (propositional letters), disjuncts per clauses, and clauses per knowledgebase. We considered the values 2, 4, and 6 for the number of disjuncts as this reflects what might be common values in applications, we considered 10 and 100 for the number of clauses, and similarly between 25 and 250 arguments, as they represent the numbers that might be found in small and larger applications. Table 1 shows that for each row, the time taken was similar for each column. So increasing the number of disjuncts per clause (i.e. d), or increasing the number of clauses (i.e. c), does not substantially affect the time taken. In contrast, the number of arguments does substantially increase the time taken. This can be clearly seen in each column.

The algorithm for bounds involves more computation since repeated queries are made to the inference algorithm. As a result the average time to obtain bounds were slower than for entailment as indicated by the results in Table 2. A simple improvement to the algorithm to decrease the time would be to only form the completion once (rather than form the completion each time the inference algorithm is called) and then use this completion each time the SAT solver is called.

The conclusion that we draw from the evaluations is that by basing the algorithms on off-the-shelf SAT solvers, we are able to have scalable reasoning with epistemic graphs. Given a set of constraints for an epistemic graph together with a set of observations, we are able to quickly determine the belief in any of the arguments. In other words, the belief on some arguments can be efficiently propagated through the graph to determine the belief in the others. We can claim that this is scalable because we see that even with 100s of arguments with clauses of up to 6 disjuncts, and a set of constraints plus observation of 100 clauses, the time taken is a few seconds. For instance, with 200 arguments, a maximum of 5 disjuncts, and 500 clauses in the knowledgebase, the average time is 2.71 seconds.

5. An Application of Automated Reasoning

We now consider an extended example (which has been adapted from [1]) to illustrate how we can use the automated reasoning as part of an automated persuasion system. Assume we have the graph presented in Figure 4 and that through, for instance, crowdsourcing data, we have learned which constraints should be associated with a given user profile. So now we assume we are dealing with a user of an automated persuasion system whose profile leads to the selection of the following constraints in order to predict his or her attitudes.

(1) $p(\mathtt{B}) > 0.5 \land p(\mathtt{C}) < 0.5 \land p(\mathtt{D}) < 0.5 \rightarrow p(\mathtt{A}) > 0.5$
(2) $p(\mathtt{B}) > 0.7 \land p(\mathtt{C}) < 0.5 \land p(\mathtt{D}) < 0.5 \rightarrow p(\mathtt{A}) > 0.8$
(3) $p(\mathtt{B}) > 0.9 \land p(\mathtt{C}) < 0.5 \land p(\mathtt{D}) < 0.5 \rightarrow p(\mathtt{A}) > 0.9$
(4) $p(\mathtt{C}) \geq 0.9 \rightarrow p(\mathtt{A}) < 0.25$
(5) $p(\mathtt{D}) \leq 0.5 \rightarrow p(\mathtt{A}) \geq 0.25$
(6) $p(\mathtt{D}) > 0.75 \rightarrow p(\mathtt{A}) < 0.75$
(7) $p(\mathtt{E}) > 0.9 \rightarrow p(\mathtt{B}) < 0.5$
(8) $p(\mathtt{E}) \leq 0.5 \land p(\mathtt{F}) > 0.5 \rightarrow p(\mathtt{B}) > 0.5$
(9) $p(\mathtt{G}) > 0.5 \rightarrow p(\mathtt{C}) < 0.5$
(10) $p(\mathtt{H}) > 0.5 \rightarrow p(\mathtt{D}) < 0.5$
(11) $p(\mathtt{I}) > 0.75 \rightarrow p(\mathtt{D}) \leq 0.5$
(12) $p(\mathtt{J}) > 0.5 \rightarrow p(\mathtt{E}) < 0.5$
(13) $p(\mathtt{J}) > 0.5 \rightarrow p(\mathtt{B}) > 0.5$
(14) $p(\mathtt{J}) > 0.5 \land p(\mathtt{F}) > 0.5 \rightarrow p(\mathtt{B}) > 0.9$

We explain these constraints as follows: (1) If B is believed, and C and D are disbelieved, then A is believed; (2) This refines above so if B is strongly believed, then A is strongly believed; (3) This refines above so if B is very strongly believed, then A is very strongly believed; (4) If C is very strongly believed, then A is strongly disbelieved; (5) If D is not believed, then A is not strongly disbelieved; (6) If D is strongly believed, then A is not strongly believed; (7) If E is strongly believed, then B is disbelieved; (8) If E is not believed, and F is believed, then B is believed; (9) If G is believed, then C is disbelieved; (10) If H is believed, then D is disbelieved; (11) If I is strongly believed, then D is not believed; (12) If J is believed, then E is disbelieved; (13) If J is believed, then B is believed; And (14) If J is believed, and F is believed, then B is strongly believed;

We can use these constraints together with any specific observations we have about an individual (perhaps a lapsed patient at a dental surgery) to predict the belief in the persuasion goal (i.e. argument A)). For instance, if we know that a given individual strongly believes F and G, e.g. $p(\mathtt{F}) = 0.8$ and $p(\mathtt{G}) = 0.8$, then we can infer that C is disbelieved (i.e. $p(\mathtt{C}) < 0.5$). However, it is not possible to infer whether the individual believes or disbelieves the persuasion goal.

Next, we could consider presenting an argument to the individual in order to see whether (according to the epistemic graph) the persuasion goal is believed or even strongly believed. For instance, if we present H and J, we may assume that the patient believes the arguments (i.e. $p(\mathtt{H}) > 0.5$ and $p(\mathtt{J}) > 0.5$). This assumption could be based on analyzing the crowdsourced data to see which arguments are believed after being presented. Then from $p(\mathtt{H}) > 0.5$ and $p(\mathtt{J}) > 0.5$, together with the original information about

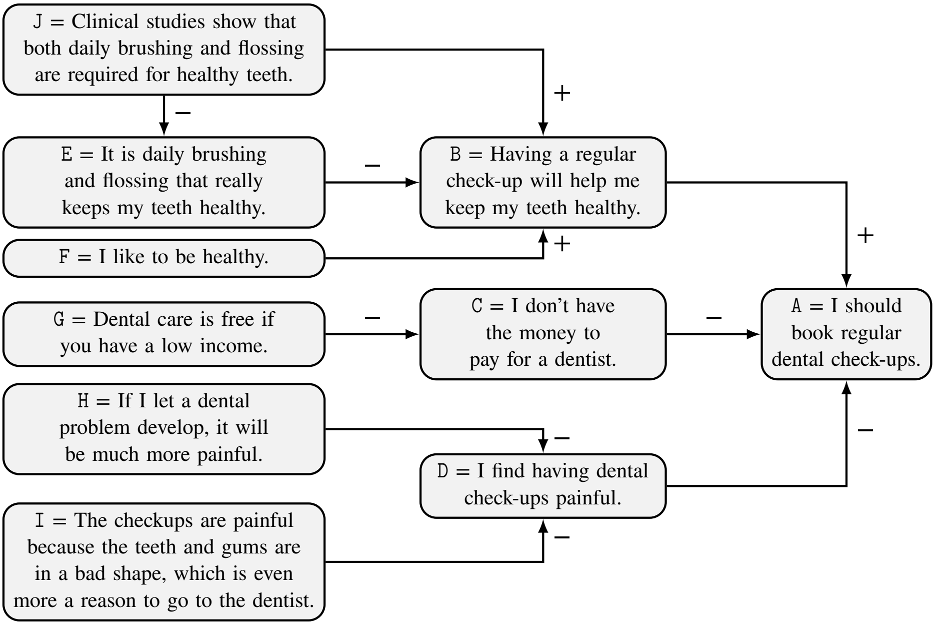

J = Clinical studies show that both daily brushing and flossing are required for healthy teeth.
E = It is daily brushing and flossing that really keeps my teeth healthy.
F = I like to be healthy.
G = Dental care is free if you have a low income.
H = If I let a dental problem develop, it will be much more painful.
I = The checkups are painful because the teeth and gums are in a bad shape, which is even more a reason to go to the dentist.
B = Having a regular check-up will help me keep my teeth healthy.
C = I don't have the money to pay for a dentist.
D = I find having dental check-ups painful.
A = I should book regular dental check-ups.
+
−
−
+
+
−
−
−
−
−

Figure 4. Epistemic graph (adapted from [1]) for the domain model for a case study on encouraging people to take regular dental check-ups.

the patient (i.e. $p(\mathtt{F}) = 0.8$ and $p(\mathtt{G}) = 0.8$), we can infer B and A are very strongly believed (i.e. $p(\mathtt{B}) > 0.9$ and $p(\mathtt{A}) > 0.9$).

Since it is possible to acquire substantial amounts of crowdsourced data, and apply machine learning to generate constraints [20], we can easily acquire large numbers of constraints on a topic that can be harnessed for user models in automated persuasion. The above example only involved 14 constraints, and so the inferences can be made by hand, but if we have 100s of constraints (which can easily arise if we have an argument graph with 100 arguments), then we need automated reasoning such as the approach presented in this paper (which was shown in the previous section to scale to 100s of clauses with 200 arguments) to be able to identify the implications of specific options for presenting arguments.

6. Discussion

Epistemic graphs offer a rich and flexible formalism for modelling argumentation. The approach provides subjective reasoning by allowing different agents to be modelled by a different set of constraints (which can be useful in complex problem analysis where different perspectives and the associated unncertainty is captured). This may be useful for modelling how different decision makers make their decisions based on their beliefs in the relevant arguments by each presenting an epistemic graph. Epistemic graphs also allow for better modelling of imperfect agents, which can be important in multi–agent application with dialogical argumentation (e.g. persuasion, negotiation, etc.).

The benefit of the work presented in this paper is that we can use the automated reasoning system to allow us to draw inferences about a situation modelled by an epistemic graph, or about what inferences another agent would draw based on what we assume about their epistemic graph. Off-the-shelf SAT solvers (which are available for a range of programming languages) allow the reasoning to scale to large epistemic graphs, and this allows us to deal with much larger numbers of arguments than possible with previous proposals for automated reasoning with epistemic graphs [15,16]. The approach of using the completion clauses can be adapted to a range of automated reasoning tasks. We will explore these in future work. We will also consider generalizing the algorithms to handle the general version of epistemic graphs that was presented in [1].

References

[1] Hunter A, Polberg S, Thimm M. Epistemic graphs for representing and reasoning with positive and negative influences of arguments. Artificial Intelligence. 2020;281:103236.

[2] Amgoud L, Ben-Naim J. Ranking-Based Semantics for Argumentation Frameworks. In: Proceedings of SUM'13. vol. 8078 of LNCS. Springer; 2013. p. 134-47.

[3] Amgoud L, Ben-Naim J, Doder D, Vesic S. Acceptability Semantics for Weighted Argumentation Frameworks. In: Proceedings of IJCAI'17. IJCAI; 2017. p. 56-62.

[4] Bonzon E, Delobelle J, Konieczny S, Maudet N. A Comparative Study of Ranking-Based Semantics for Abstract Argumentation. In: Proceedings of AAAI'16. AAAI Press; 2016. p. 914-20.

[5] Cayrol C, Lagasquie-Schiex M. Graduality in Argumentation. Journal of Artificial Intelligence Research. 2005;23:245-97.

[6] Leite J, Martins J. Social Abstract Argumentation. In: Proceedings of IJCAI'11. AAAI Press; 2011. p. 2287-92.

[7] Rago A, Toni F, Aurisicchio M, Baroni P. Discontinuity-Free Decision Support with Quantitative Argumentation Debates. In: Proceedings of KR'16. AAAI Press; 2016. p. 63-73.

[8] da Costa Pereira C, Tettamanzi A, Villata S. Changing One's Mind: Erase or Rewind? Possibilistic Belief Revision with Fuzzy Argumentation Based on Trust. In: Proceedings of IJCAI'11. AAAI Press; 2011. p. 164-71.

[9] Baroni P, Romano M, Toni F, Aurisicchio M, Bertanza G. Automatic evaluation of design alternatives with quantitative argumentation. Argument & Computation. 2015;6(1):24-49.

[10] Potyka N. Continuous Dynamical Systems for Weighted Bipolar Argumentation. In: Proceedings of KR'18. AAAI Press; 2018. p. 148-57.

[11] Pu F, Luo J, Zhang Y, Luo G. Argument Ranking with Categoriser Function. In: Proceedings of KSEM'14. vol. 8793 of LNCS. Springer; 2014. p. 290-301.

[12] Pu F, Luo J, Zhang Y, Luo G. Attacker and Defender Counting Approach for Abstract Argumentation. In: Proceedings of CogSci'15. cognitivesciencesociety.org; 2015. p. 1.

[13] Brewka G, Woltran S. Abstract Dialectical Frameworks. In: Proceedings of KR'10. AAAI Press; 2010. p. 102-11.

[14] Brewka G, Strass H, Wallner J, Woltran S. Weighted Abstract Dialectical Frameworks. In: Proceedings of AAAI'18. AAAI Press; 2018. p. 1779-86.

[15] Hunter A, Polberg S. A Model-Based Theorem Prover for Epistemic Graphs for Argumentation. In: Proceedings of ECSQARU'19. vol. 11726 of LNCSe. Springer; 2019. p. 50-61.

[16] Hunter A, Polberg S, Potyka N. Updating Belief in Arguments in Epistemic Graphs. In: Proceedings of KR'18. AAAI Press; 2018. p. 138-47.

[17] Vizel Y, Weissenbacher G, Malik S. Boolean Satisfiability Solvers and Their Applications in Model Checking. Proceedings of the IEEE. 2015;103(11):2021-35.

[18] Coste-Marquis S, Devred C, Marquis P. Constrained Argumentation Frameworks. In: Proceedings of KR'06. AAAI Press; 2006. p. 112-22.

[19] Ignatiev A, Morgado A, Marques-Silva J. PySAT: A Python Toolkit for Prototyping with SAT Oracles. In: Proceedings of SAT'18. vol. 10929 of LNCS. Springer; 2018. p. 428-37.

[20] Hunter A. Learning Constraints for the Epistemic Graphs Approach to Argumentation. In: Proceedings of COMMA'20. vol. 326. IOS Press; 2020. p. 239-50.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220152

Explaining Change in Quantitative Bipolar Argumentation[1]

Timotheus KAMPIK [a,b], Kristijonas ČYRAS [c]
[a] *Umeå University, Umeå, Sweden*
[b] *SAP Signavio, Berlin, Germany*
[c] *Ericsson Research, Stockholm, Sweden*
ORCiD ID: Timotheus Kampik https://orcid.org/0000-0002-6458-2252, Kristijonas Čyras https://orcid.org/0000-0002-4353-8121

Abstract. This paper presents a formal approach to explaining change of inference in Quantitative Bipolar Argumentation Frameworks (QBAFs). When drawing conclusions from a QBAF and updating the QBAF to then again draw conclusions (and so on), our approach traces changes – which we call *strength inconsistencies* – in the partial order that a semantics establishes on the arguments in the QBAFs. We trace the strength inconsistencies to specific arguments, which then serve as explanations. We identify both sufficient and counterfactual explanations for strength inconsistencies and show that our approach guarantees that explanation arguments exist if and only if an update leads to strength inconsistency.

Keywords. quantitative argumentation, explainable AI, non-monotonic reasoning

1. Introduction

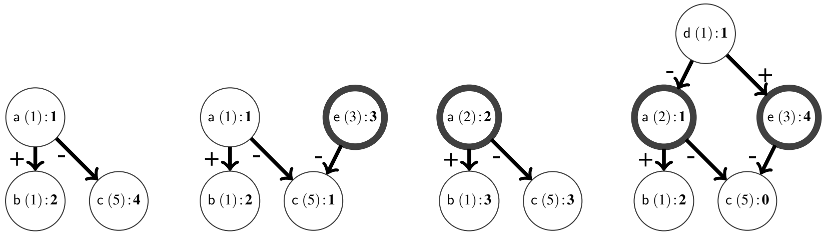
A key challenge in the domain of eXplainable Artificial Intelligence (XAI) is the explanation of an agent's *change of mind*: if the agent has inferred (or decided) A at time t_0, why does she infer A' at t_1? This challenge is reflected in fundamental approaches to decision-making and reasoning. From the perspective of microeconomic decision theory (see, e.g. [1]), a basic assumption is that the agent has *consistent preferences*, i.e. assuming two independent choices A and A', the agent must not decide A and then A' if A' has been available as a decision option all along and also A is still available, as long as no relevant change in circumstances has occurred. If an agent's preferences on the available decision options are not consistent, one would expect an explanation that highlights this relevant change in circumstances that violates the *ceteris paribus*[2] condition. From an automated reasoning perspective, one would expect that an explanation is provided if monotony of entailment is violated, i.e. if the agent first infers A and then A', such that $A \not\subseteq A'$, an explanation of why previously inferred statements are to be rejected should be provided. In this paper, we define such explanations in the setting of evolving Quantitative Bipolar Argumentation Frameworks (QBAFs) [2].

[1] This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

[2] Translates to: "all else unchanged" and is a crucial assumption in classical models of economic rationality.

Specifically, our goal is to explain, in a QBAF that was updated by changing arguments, their initial credences and/or relationships, the relative change in acceptability of specified arguments. We strive for explanations of changes in arguments' relative strengths that pertain to in some sense minimal information causing those changes. We adopt the notions of (attributive) sufficient explanations and counterfactual explanations (see [3] for an excellent overview of counterfactual explanations) to the setting of explaining changes in the partial ordering of argument strengths in evolving QBAFs. In the example below, we give intuitive readings of the introduced concepts; rigorous definitions follow later.

Example 1
*We start with the QBAF depicted in Figure 1.1, which we denote by QBF. We have the nodes (*arguments*) a (with initial strength $\tau(\mathsf{a}) = 1$), b (with $\tau(\mathsf{b}) = 1$), and c (with $\tau(\mathsf{c}) = 5$); a supports b and attacks c. Here, b and c are topic arguments, i.e. arguments that we want to weigh against each other: the topic argument with the highest Final Strength (FS) can be considered the most* promising. *a is a support argument, i.e. the final strength $\sigma(\mathsf{a})$ is not directly relevant to the decision but impacts the FS of (some) topic arguments. Typically, we determine $\sigma(\mathsf{x})$ of an argument x by aggregating the FS of its supporters and attackers. For instance, we can add to $\tau(\mathsf{x})$ the FS of supporters of x and subtract the FS of attackers of x, iteratively, starting with the neither attacked nor supported leaf arguments (whose FS equals initial strength). Here, we get $\sigma(\mathsf{b})$ by adding $\sigma(\mathsf{a}) = \tau(\mathsf{a})$ to $\tau(\mathsf{b})$: $1 + 1 = 2$; and $\sigma(\mathsf{c})$ by subtracting $\sigma(\mathsf{a})$ from $\tau(\mathsf{c})$: $5 - 1 = 4$. Consequently, c is the is the topic argument with the highest FS (and hence our recommendation).*



1.1: *QBF* 1.2: *QBF*$^{\mathsf{e}}$ 1.3: *QBF*$^{\mathsf{a}}$ 1.4: *QBF*$'$

Figure 1. *QBF* and different updates thereof. Here and henceforth, a node labelled x (i) : **f** carries argument x with initial strength $\tau(\mathsf{x}) = i$ and final strength $\sigma(\mathsf{x}) = \mathbf{f}$. Edges labelled $+$ and $-$ respectively represent attack and support. Arguments with bold borders are strength inconsistency explanation arguments.

Later, our knowledge base receives an update. The update can be of different forms of changes to the QBAF: we give examples of the different resulting situations in Figures 1.2, 1.3, 1.4. As we will spell out shortly, we determine the FSs of b and c (using the same approach as before) in each situation and find that, after any of the updates, b, rather than c, is the highest-ranking topic argument. We are then interested in explaining why the ranking of b relative to c has changed.
(i) In Figure 1.2, the final strength of b is 2 and the final strength of c is 1. Here, the new argument e directly decreases the final strength of c. Intuitively, (the addition of) e explains the change in the relative ordering of the final strengths for $\{\mathsf{b},\mathsf{c}\}$.
(ii) In Figure 1.3, the final strengths of b and c are equal to 3. Here, the change in the initial strength of a from 1 to 2 leads to changes to the final strengths of b and c. Intuitively, (the change in the initial strength of) a explains the change in the relative ordering of the final strengths for $\{\mathsf{b},\mathsf{c}\}$.

(iii) In Figure 1.4, the final strength of b *is* 2 *and that of* c *is* 0. *Here, we have the addition of new arguments* d *and* e, *as well as a change to the initial strength of* a, *that both influence the final strengths of* b *and* c. *Now, one could say that here all the changes collectively explain the change in the relative ordering of the final strengths for* $\{b, c\}$. *However, let us search for in some sense* minimal *explanations.*

For instance, the addition of only e sufficient *to make* b *stronger than* c, *in the absence of other changes: this is the situation in Figure 1.2. Additionally, since without adding* e *and in the absence of the other changes we would just have QBF we started with as in Figure 1.1, we conclude that* $\{e\}$ *is a minimal explanation of the change in the relative ordering of the final strengths for* $\{b, c\}$.

Similarly, absent the addition of d *and* e, *with only the change to* a, *we would be in the situation in Figure 1.3, where* c *is not stronger than* b. *Hence,* $\{a\}$ *is also a minimal explanation of the change in relative ordering of the final strengths for* $\{b, c\}$.

How about other combinations? Absent the change to a *(but with the addition of* e *and* d*), we would find* $\sigma(a) = 0$ *and thus* $\sigma(b) = 1 = \sigma(c)$. *I.e. the relative strengths of* b *and* c *would change from QBF. So, intuitively,* $\{d, e\}$ *also explains the change. But it is not a minimal explanation, because* $\{e\}$ *is a smaller one. On the other hand, absent the addition of* e, *we would find* $\sigma(a) = \tau(a) - \sigma(d) = 2 - \tau(d) = 1$, *and the final strengths of* b *and* c *would be* $\sigma(b) = \tau(b) + \sigma(a) = 2$ *and* $\sigma(c) = \tau(c) - \sigma(a) = 4$, *just as in QBF to begin with. So* $\{a, d\}$ *is not an explanation, for there is no change in the relative strengths of* b *and* c. *Similarly, if the addition of* d *was the only change, we would find* $\sigma(a) = 0$ *and the final strengths of* b *and* c *equal to their initial strengths. So* d *alone is not an explanation, either.*

In the end, we have two $\subset$*-minimal* sufficient *explanations, namely* $\{a\}$ *and* $\{e\}$, *of the change in the relative ordering of the final strengths for* $\{b, c\}$. *Note, however, that the absence of the changes to* a *does not* counterfactually *restore strength consistency. That is, as shown in the above paragraph, if the initial strength of* a *were* 1 *in* QBF' *(as it is in QBF), we would have* $\sigma(b) = \sigma(c) = 1$ *in* QBF', *whereas* $\sigma(b) < \sigma(c)$ *in QBF. On the other hand, as shown in the above paragraph, were* e *absent from* QBF', *we would have* $\sigma(b) = 2 < 4 = \sigma(c)$ *and strength consistency would be restored: so* $\{e\}$ *is not only a sufficient explanation, but also a counterfactual one. In fact,* $\{e\}$ *is a* $\subset$*-minimal counterfactual explanation: any counterfactual explanation entails* e, *because in order to restore strength consistency of* b *and* c *we need to revert (the addition of)* e.

The above explanations satisfy the following properties: i) it is sufficient to apply changes to only these arguments (and to ignore the other changes) in the QBAF for the partial order of final strengths to coincide with the one obtained after the actual update (sufficient explanations); ii) in addition to i), reverting the changes made to these arguments only (and keeping all the other changes) restores the original partial order (counterfactual explanations); iii) the set of explanation arguments is $\subset$-minimal among the sets that satisfy i) or ii). These explanations achieve our objective of explaining any change in the partial order that the assignment of the final strengths establishes on a set of arguments of interest, by identifying arguments whose *change* (addition, removal, or change of initial strength) leads to the change in the partial order of the final strengths.

In what follows we formalise the intuition given above by defining and analysing novel forms of explanations in QBAFs. We provide the formal preliminaries in Section 2. We introduce in Section 3 our formal framework for explaining change of inference in

QBAFs. We analyse the properties of our explanations in Section 4. Finally, in Section 5 we discuss our work in the context of related research.

2. Preliminaries

This section introduces the formal preliminaries of our work. Let $\mathbb{I}$ be a set of elements and let $\preceq$ be a preorder on $\mathbb{I}$. Typically, $\mathbb{I} = [0,1]$ is the unit interval[3] and $\preceq = \leqslant$ is the standard less-than-equal ordering. A *quantitative bipolar argumentation framework* contains a set of arguments related by binary *attack* and *support* relations, and assigns an *initial strength* in $\mathbb{I}$ to the arguments. The initial strength can be thought of as initial credence in, or importance of, arguments. Typically, the greater the strength in say the unit interval, the more credible or important the argument is.

Definition 1 (Quantitative Bipolar Argumentation Framework (QBAF) [4,2])
A Quantitative Bipolar Argumentation Framework (QBAF) *is a quadruple* $(Args, \tau, Att, Supp)$ *consisting of a set of arguments Args, an* attack *relation* $Att \subseteq Args \times Args$, *a* support *relation* $Supp \subseteq Args \times Args$ *and a total function* $\tau : Args \to \mathbb{I}$ *that assigns the* initial strength $\tau(\mathsf{a})$ *to every* $\mathsf{a} \in Args$.

Henceforth, we assume as given a fixed but otherwise arbitrary QBAF $QBF = (Args, \tau, Att, Supp)$, unless specified otherwise. We also assume that *Args* is finite.

Given $\mathsf{a} \in Args$, the set $Att_{QBF}(\mathsf{a}) := \{\mathsf{b} \mid \mathsf{b} \in Args, (\mathsf{b}, \mathsf{a}) \in Att\}$ is the set of attackers of a and each $\mathsf{b} \in Att_{QBF}(\mathsf{a})$ is an *attacker* of a; the set $Supp_{QBF}(\mathsf{a}) := \{\mathsf{c} \mid \mathsf{c} \in Args, (\mathsf{c}, \mathsf{a}) \in Supp\}$ is the set of supporters of a and each $\mathsf{c} \in Supp_{QBF}(\mathsf{a})$ is a *supporter* of a. We may drop the subscript $_{QBF}$ when the context is clear.

Reasoning in QBAFs amounts to updating the initial strengths of arguments to their final strengths, taking into account the strengths of attackers and supporters. Specifically, given a QBAF, a strength function assigns final strengths to arguments in the QBAF. Different ways of defining a strength function are called gradual semantics [2,4].

Definition 2 (QBAF Semantics and Strength Functions)
A gradual semantics σ *defines for* $QBF = (Args, \tau, Att, Supp)$ *a* strength function $\sigma_{QBF} : Args \to \mathbb{I}$ *that assigns the* final strength $\sigma_{QBF}(\mathsf{a})$ *to each argument* $\mathsf{a} \in Args$.

For the sake of conciseness, we do not consider the case of a gradual semantics as a partial function that may leave the final strength value of an argument undefined. We may abuse the notation and drop the subscript $_{QBF}$ so that σ denotes the strength function, whenever the context is clear. The (final) strength of an argument can be thought of as its (final) credence or importance. Typically, the greater the strength in $\mathbb{I}$, the more credible or important the argument is. In our examples, we use $\mathbb{I} = \mathbb{R}$.

A gradual semantics can define a strength function as a composition of multivariate real-valued functions that determines the strength of a given argument by aggregating the strengths of its attackers and supporters, taking into account the initial strengths [4]. A strength function so defined is recursive and generally takes iterated updates to produce a sequence of strength vectors, whence the final strengths are defined as the limits (or fixed points) if they exist. However, for *acyclic* QBAFs (without directed cycles) defining a

[3] However, in our examples we use a simplistic semantics and hence a different interval.

semantics and computing the final strengths can be more straightforward: in the topological order of an acyclic QBAF as a graph, start with the leaves,[4] set their final strengths to equal their initial strengths, and then iteratively update the strengths of parents whose all children already have final strengths defined. For instance, in Figure 1.4 from Example 1, we can use the function $\sigma(\mathsf{x}) = \tau(\mathsf{x}) + \left(\sum_{\mathsf{y} \in Supp(\mathsf{x})} \sigma(\mathsf{y}) - \sum_{\mathsf{z} \in Att(\mathsf{x})} \sigma(\mathsf{z})\right)$ defined as a composition, namely sum, of the initial strength ($\tau(\mathsf{x})$) and the difference between the added strengths of the supporters and the added strengths of the attackers ($\sum_{\mathsf{x} \in Supp(\mathsf{x})} \sigma(\mathsf{y}) - \sum_{\mathsf{z} \in Att(\mathsf{x})} \sigma(\mathsf{z})$). It gives final strengths of arguments in the topological order of QBF': first $\sigma(\mathsf{d}) = \tau(\mathsf{d}) = 1$, then $\sigma(\mathsf{a}) = \tau(\mathsf{a}) - \sigma(\mathsf{d}) = 1$ and $\sigma(\mathsf{e}) = \tau(\mathsf{e}) + \sigma(\mathsf{d}) = 4$, and then $\sigma(\mathsf{b}) = \tau(\mathsf{b}) + \sigma(\mathsf{a}) = 2$ and $\sigma(\mathsf{c}) = \tau(\mathsf{c}) - \sigma(\mathsf{a}) - \sigma(\mathsf{e}) = 0$.

While many gradual semantics can be defined for QBAFs in general, their convergence is not always guaranteed in a particular QBAF. For several well-studied semantics, convergence is however always guaranteed in acyclic QBAFs. (See e.g. [4] for a neat exposition of convergence results under various semantics.) In what follows, we restrict our attention to QBAFs for which a fixed but otherwise arbitrary gradual semantics is well-defined. In other words, our study applies to the setting where a gradual semantics σ defines a total strength function σ_{QBF} assigning the final strengths to all arguments of a given QBF. Specifically for illustration purposes to avoid dealing with the sometimes demanding definitions of strength functions, we use acyclic QBAFs and the above strength function σ (in accordance with a topological ordering of an acyclic QBAF). We however note that both the formal definitions and theoretical analysis given in the paper apply to the general setting of well-defined gradual semantics giving total strength functions.

3. Change Explainability in QBAFs

In this section, we introduce our formal approach to change explainability in QBAFs. We start by introducing the notion of *strength consistency*. Henceforth in this section, unless stated otherwise, we let $QBF = (Args, \tau, Att, Supp)$ and $QBF' = (Args', \tau', Att', Supp')$ be QBAFs, let $\mathsf{a}, \mathsf{b}, \mathsf{x}, \mathsf{y} \in Args \cap Args'$, let σ be a strength function, and let $S \subseteq Args \cup Args'$. Let us highlight here that we do not formalise the change operation; instead, we merely assume that we have two QBAFs that have at least two arguments in common, and the second QBAF can be considered a revised (or: *updated*) version of the first one.

Definition 3 (Strength Consistency)
We say that a *is* strength-consistent *w.r.t.* b*, denoted by* $\mathsf{a} \sim_{\sigma, QBF, QBF'} \mathsf{b}$*, iff the following statements hold true:*

- *If* $\sigma_{QBF}(\mathsf{a}) > \sigma_{QBF}(\mathsf{b})$ *then* $\sigma_{QBF'}(\mathsf{a}) > \sigma_{QBF'}(\mathsf{b})$*;*
- *If* $\sigma_{QBF}(\mathsf{a}) < \sigma_{QBF}(\mathsf{b})$ *then* $\sigma_{QBF'}(\mathsf{a}) < \sigma_{QBF'}(\mathsf{b})$*;*
- *If* $\sigma_{QBF}(\mathsf{a}) = \sigma_{QBF}(\mathsf{b})$ *then* $\sigma_{QBF'}(\mathsf{a}) = \sigma_{QBF'}(\mathsf{b})$.

Intuitively, two arguments are strength-consistent only if their relative strengths correspond between the two QBAFs. In an obvious way, $\mathsf{a} \not\sim_{\sigma, QBF, QBF'} \mathsf{b}$ denotes the negation of $\mathsf{a} \sim_{\sigma, QBF, QBF'} \mathsf{b}$ and we say that a and b are *strength-inconsistent*. When there is no ambiguity, we drop the subscripts and write $\mathsf{a} \sim \mathsf{b}$ to denote that a is strength-consistent w.r.t. b, and similarly for the derived notions.

[4] Here, leaves are nodes without incoming edges.

In this work we aim to provide a formal approach to supplying answers to questions regarding changes in arguments' relative strengths in an evolving QBAF. The main objective of this paper is to define explanations as to why, given two QBAFs, any two arguments are strength-inconsistent (or strength-consistent).

As a prerequisite for generating our explanations, we introduce the notion of a QBAF *reversal* with respect to a set of arguments, where such sets of arguments will later play the role of explanations. Colloquially speaking, given QBAFs QBF and its update QBF', a reversal of QBF' to QBF w.r.t. a set of arguments S updates the properties of every argument from S in QBF' so that they reflect the properties of the same argument in QBF: arguments from S that are not in QBF are deleted and arguments from S that are in QBF but not in QBF' are restored.

Definition 4 (QBAF Reversal)
We define the reversal *of QBF' to QBF w.r.t. $S \subseteq Args \cup Args'$, denoted by $QBF_{\leftarrow QBF'}(S)$, as a QBAF $(Args^*, \tau^*, Att^*, Supp^*)$, where:*

- $Args^* = (Args' \cup S) \setminus (S \setminus Args)$;
- $Att^* = \underbrace{(Att' \setminus (S \times Args))}_{\textit{Attacks in } QBF' \textit{ that are not from } S \textit{ to } Args} \cup \underbrace{(S \times Args^* \cap Att)}_{\textit{Attacks in } QBF \textit{ from } S \textit{ to } Args^*}$;
- $Supp^* = (Supp' \setminus (S \times Args)) \cup (S \times Args^* \cap Supp)$;
- $\tau^* : Args^* \to \mathbb{I}$ *and* $\forall \mathsf{x} \in Args^*$ *the following statement holds true:*

$$\tau^*(\mathsf{x}) = \begin{cases} \tau(\mathsf{x}), & \textit{if } \mathsf{x} \in Args \cap S; \\ \tau'(\mathsf{x}), & \textit{otherwise} . \end{cases}$$

Intuitively: for arguments that were removed (i.e. arguments from $Args \setminus Args'$), those from S are added back; for arguments that were added (i.e. arguments from $Args' \setminus Args$), those from S are removed. The arguments are restored with the associated initial strengths, attacks and supports: in the reversal, we restore "old" attacks and supports from S; we leave "new" attacks and supports unless they are from S to the "old" arguments. For visual intuition, a Venn diagram of the set $Args^*$ is given in Figure 2.

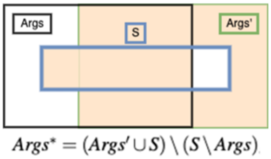


Figure 2. Venn diagram for $Args^*$ (shaded in light and weakly saturated reddish yellow 'sand' colour) in the reversal $QBF_{\leftarrow QBF'}(S) = (Args^*, \tau^*, Att^*, Supp^*)$ of QBF' to QBF w.r.t. $S \subseteq Args' \cup Args$. (Args, Args' and S in small highlighted rectangles are labels of the enclosures highlighted in corresponding colours.)

Using the notion of a QBAF reversal, we introduce different notions of *strength inconsistency explanations*, that are sets of arguments intuitively described as follows:

- $\emptyset$ is both a *sufficient* and a *counterfactual* explanation if we do not find strength inconsistency after the update from QBF to QBF'.
- $S \neq \emptyset$ is a sufficient explanation of strength inconsistency after the update from QBF to QBF' if the inconsistency persists when we reverse everything *except* S back – so changes to S are sufficient for the inconsistency.

- $S \neq \emptyset$ is a counterfactual explanation of strength inconsistency after the update from QBF to QBF' if the inconsistency persists when we reverse everything except S back, but does not persist when we reverse back only S itself – so the absence of changes to S would restore consistency.
- For both sufficient and counterfactual explanations, we define $\subset$-minimal versions.

Definition 5 (Strength Inconsistency Explanations)
We say that $S \subseteq Args' \cup Args$ is a:

- Sufficient Strength Inconsistency (SSI) explanation *of* x *and* y *w.r.t.* σ*,* QBF*, and* QBF' *iff the following statement holds true:*

$$\textbf{\textit{either}} \qquad \underbrace{\left(S = \emptyset \text{ and } \mathsf{x} \sim_{\sigma,QBF,QBF'} \mathsf{y}\right)}_{\mathsf{x} \text{ and } \mathsf{y} \text{ are strength-consistent, so empty explanation}}$$

$$\textbf{\textit{or}} \quad \underbrace{\left(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y} \text{ and } \mathsf{x} \not\sim_{\sigma,QBF,QBF_{\leftarrow QBF'}((Args \cup Args')\setminus S)} \mathsf{y}\right)}_{\mathsf{x} \text{ and } \mathsf{y} \text{ are strength-inconsistent and remain so after reversing everything but } S \text{ back}}$$

  $SX(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$ *denotes all SSI explanations of* x *and* y *w.r.t.* σ*,* QBF*, and* QBF' *and* $SX_{\subset\min}(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$ *denotes all* $\subset$*-minimal SSI explanations of* x *and* y *w.r.t.* σ*,* QBF*, and* QBF'*.*
- Counterfactual Strength Inconsistency (CSI) explanation *of* x *and* y *w.r.t.* σ*,* QBF*, and* QBF' *iff the following statement holds true:*

$$\underbrace{S \in SX(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})}_{S \text{ is an SSI of } \mathsf{x} \text{ and } \mathsf{y}} \qquad \textit{and} \qquad \underbrace{\mathsf{x} \sim_{\sigma,QBF,QBF_{\leftarrow QBF'}(S)} \mathsf{y}}_{\mathsf{x} \text{ and } \mathsf{y} \text{ become strength-consistent after reversing } S}$$

  $CX(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$ *denotes all CSI explanations of* x *and* y *w.r.t.* σ*,* QBF*, and* QBF' *and* $CX_{\subset\min}(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$ *denotes all* $\subset$*-minimal CSI explanations of* x *and* y *w.r.t.* σ*,* QBF*, and* QBF'*.*

Analogously to the case of strength consistency, when there is no ambiguity, we may drop the subscripts and write simply $SX(\mathsf{x} \not\sim \mathsf{y})$ *to denote all SSI explanations of* x *and* y *(w.r.t. the implicit* σ*,* QBF *and* QBF'*), and similarly for the derived notions.*

Intuitively, a *sufficient* strength inconsistency explanation identifies changes that explain why the relative strengths between two arguments are inconsistent, given an initial QBAF and an update thereof; the changes that a *counterfactual* explanation identifies are – in addition – *counterfactual*, i.e. their absence would restore the initial relative strengths between two arguments. Let us revisit the example from the *Introduction* section to illustrate how strength inconsistency explanations explain change of inference in QBAFs, this time with the formal notation.

Example 2 (Example 1 revisited)
Figures 3.1 and 3.2 depict again the QBAFs $QBF = (\{\mathsf{a},\mathsf{b},\mathsf{c}\}, \tau, \{(\mathsf{a},\mathsf{c})\}, \{(\mathsf{a},\mathsf{b})\})$ *and* $QBF' = (\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e}\}, \tau', \{(\mathsf{a},\mathsf{c}),(\mathsf{e},\mathsf{c}),(\mathsf{d},\mathsf{a})\}, \{(\mathsf{a},\mathsf{b}),(\mathsf{d},\mathsf{e})\})$ *from Example 1, where:*

- $\tau(\mathsf{a}) = \tau(\mathsf{b}) = 1$ *and* $\tau(\mathsf{c}) = 5$*;*
- $\tau'(\mathsf{a}) = 2$*,* $\tau'(\mathsf{b}) = \tau'(\mathsf{d}) = 1$*,* $\tau'(\mathsf{c}) = 5$ *and* $\tau'(\mathsf{e}) = 3$*.*

Consider the gradual semantics σ *defined using the illustrative strength function* $\sigma(\mathsf{x}) = \tau(\mathsf{x}) + \left(\sum_{\mathsf{y} \in Supp(\mathsf{x})} \sigma(\mathsf{y}) - \sum_{\mathsf{z} \in Att(\mathsf{x})} \sigma(\mathsf{z})\right)$ *that updates the strengths of arguments in an acyclic QBAF according to its topological ordering, as previously discussed. Denote*

σ_{QBF} and $\sigma_{QBF'}$ by σ and σ', respectively. Assume we are primarily interested in the final strengths of the arguments b and c: $\sigma(\mathsf{b}) = 2 < 4 = \sigma(\mathsf{c})$. In contrast, $\sigma'(\mathsf{b}) = 2 > 0 = \sigma'(\mathsf{c})$. Hence, b *is strength-inconsistent w.r.t.* c *(*$\mathsf{b} \not\sim \mathsf{c}$*), for which we have the following explanations: (i) $SX_{\subset_{\min}}(\mathsf{b} \not\sim \mathsf{c}) = \{\{\mathsf{a}\},\{\mathsf{e}\}\}$, (ii) $CX_{\subset_{\min}}(\mathsf{b} \not\sim \mathsf{c}) = \{\{\mathsf{e}\}\}$.*

Indeed, for $\{\mathsf{a}\}$, its relative complement is $S_{\{\mathsf{a}\}} := (Args \cup Args') \setminus \{\mathsf{a}\} = \{\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e}\}$, so that the reversal $QBF_{\leftarrow QBF'}(S_{\{\mathsf{a}\}})$ of QBF' to QBF w.r.t. to $S_{\{\mathsf{a}\}}$ has the arguments

$$\big(Args' \cup S_{\{\mathsf{a}\}}\big) \setminus \big(S_{\{\mathsf{a}\}} \setminus Args\big) =$$

$$(\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e}\} \cup \{\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e}\}) \setminus (\{\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e}\} \setminus \{\mathsf{a},\mathsf{b},\mathsf{c}\}) = \{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e}\} \setminus \{\mathsf{d},\mathsf{e}\} = Args.$$

Since $Args \cap S_{\{\mathsf{a}\}} = \{\mathsf{b},\mathsf{c}\}$, it follows that reversing w.r.t. all arguments except a *yields*

$$QBF^{\mathsf{a}} := QBF_{\leftarrow QBF'}(S_{\{\mathsf{a}\}}) = (Args^{\mathsf{a}}, \tau^{\mathsf{a}}, Att^{\mathsf{a}}, Supp^{\mathsf{a}}) =$$

$$\big((Args' \cup S_{\{\mathsf{a}\}}) \setminus (S_{\{\mathsf{a}\}} \setminus Args), \tau^{\mathsf{a}}, \big(Att' \setminus (S_{\{\mathsf{a}\}} \times Args)\big) \cup \big((S_{\{\mathsf{a}\}} \times Args^{\mathsf{a}}) \cap Att\big), Supp^{\mathsf{a}}\big) =$$

$$\big(Args, \{(\mathsf{a}, \tau'(\mathsf{a})), (\mathsf{b}, \tau(\mathsf{b})), (\mathsf{c}, \tau(\mathsf{c}))\}, Att' \cup \emptyset, Supp' \cup \emptyset\big) =$$

$$(Args, \{(\mathsf{a},2),(\mathsf{b},1),(\mathsf{c},5)\}, \{(\mathsf{a},\mathsf{c})\}, \{(\mathsf{a},\mathsf{b})\}).$$

So QBF^{a} is like QBF but with a*'s initial strength changed to* 2 *(as depicted in Figure 1.3 and discussed in Example 1), thus giving $\sigma_{QBF^{\mathsf{a}}}(\mathsf{b}) = 3 = \sigma_{QBF^{\mathsf{a}}}(\mathsf{c})$. So,* b *and* c *are strength-inconsistent (when updating from QBF to QBF') and remain so after reversing everything but $\{\mathsf{a}\}$ back. Hence, $\{\mathsf{a}\}$ is a $\subset$-minimal SSI, by Definition 5.*

Now observe that reversing w.r.t. a *yields*

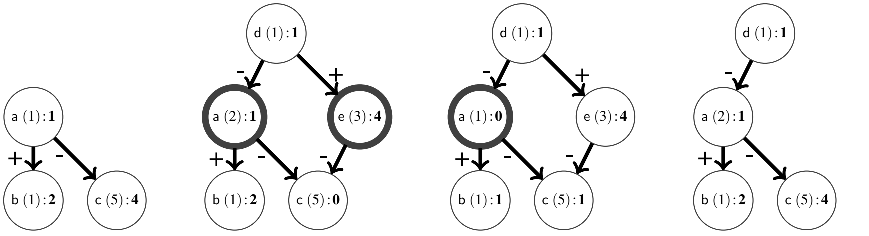
$$QBF^{*} := QBF_{\leftarrow QBF'}(\{\mathsf{a}\}) = (Args^{*}, \tau^{*}, Att^{*}, Supp^{*}) =$$

$$\big((Args' \cup \{\mathsf{a}\}) \setminus (\{\mathsf{a}\} \setminus Args), \tau^{*}, \big(Att' \setminus (\{\mathsf{a}\} \times Args)\big) \cup ((\{\mathsf{a}\} \times Args^{*}) \cap Att), Supp^{*}\big) =$$

$$\big(Args', \{(\mathsf{a}, \tau(\mathsf{a})), (\mathsf{b}, \tau'(\mathsf{b})), (\mathsf{c}, \tau'(\mathsf{c})), (\mathsf{d}, \tau'(\mathsf{d})), (\mathsf{e}, \tau'(\mathsf{e}))\}, Att', Supp'\big) =$$

$$\big(Args', \{(\mathsf{a},1),(\mathsf{b},1),(\mathsf{c},5),(\mathsf{d},1),(\mathsf{e},3)\}, Att', Supp'\big).$$

So QBF^{*} is like QBF' but with a*'s initial strength unchanged from* 1 *(depicted in Figure 3.3), thus giving $\sigma_{QBF^{*}}(\mathsf{b}) = 1 = \sigma_{QBF^{*}}(\mathsf{c})$. That is,* b *and* c *do **not** become strength-consistent after reversing $\{\mathsf{a}\}$ (i.e. $\mathsf{b} \not\sim_{\sigma, QBF, QBF_{\leftarrow QBF'}(\{\mathsf{a}\})} \mathsf{c}$), whence $\{\mathsf{a}\}$ is **not** a CSI.*



3.1: *QBF* 3.2: *QBF'* 3.3: $QBF_{\leftarrow QBF'}(\{\mathsf{e}\})$. 3.4: $QBF_{\leftarrow QBF'}(\{\mathsf{e}\})$.

Figure 3. QBAFs for explanations from Example 1.

For $\{\mathsf{e}\}$, with $S_{\{\mathsf{e}\}} := (Args \cup Args') \setminus \{\mathsf{e}\} = \{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d}\}$ we have that $\big(Args' \cup S_{\{\mathsf{e}\}}\big) \setminus \big(S_{\{\mathsf{e}\}} \setminus Args\big) = \{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e}\} \setminus (\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d}\} \setminus \{\mathsf{a},\mathsf{b},\mathsf{c}\}) = \{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e}\} \setminus \{\mathsf{d}\} = \{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{e}\}$.

It follows that reversing w.r.t. all arguments except e *yields*

$$QBF^{\mathsf{e}} := QBF_{\leftarrow QBF'}(S_{\{\mathsf{e}\}}) =$$
$$\big(\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{e}\},\{(\mathsf{a},\tau(\mathsf{a})),(\mathsf{b},\tau(\mathsf{b})),(\mathsf{c},\tau(\mathsf{c})),(\mathsf{e},\tau'(\mathsf{e}))\},\{(\mathsf{a},\mathsf{c}),(\mathsf{e},\mathsf{c})\},\{(\mathsf{a},\mathsf{b})\}\big) =$$
$$\big(\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{e}\},\{(\mathsf{a},1),(\mathsf{b},1),(\mathsf{c},5),(\mathsf{e},3)\},\{(\mathsf{a},\mathsf{c}),(\mathsf{e},\mathsf{c})\},\{(\mathsf{a},\mathsf{b})\}\big).$$

So QBF^{e} *is* QBF *with* e *and the attack* (e,c) *added (as depicted in Figure 1.2 and discussed in Example 1), thus giving* $\sigma_{QBF^{\mathsf{e}}}(\mathsf{b}) = 2$ *and* $\sigma_{QBF^{\mathsf{e}}}(\mathsf{c}) = 1$. *That is,* b *and* c *remain strength-inconsistent after reversing everything but* $\{\mathsf{e}\}$ *back, and so* $\{\mathsf{e}\}$ *is a* $\subset$*-minimal SSI. Further, reversing w.r.t.* e *yields*

$$QBF^{**} := QBF_{\leftarrow QBF'}(\{\mathsf{e}\}) = (Args^{**},\tau^{**},Att^{**},Supp^{**})$$
$$\big((Args' \cup \{\mathsf{e}\}) \setminus (\{\mathsf{e}\} \setminus Args),\tau^{**},\big(Att' \setminus (\{\mathsf{e}\} \times Args)\big) \cup ((\{\mathsf{e}\} \times Args^{**}) \cap Att),Supp^{**}\big) =$$
$$\big(\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d}\},\{(\mathsf{a},\tau'(\mathsf{a})),(\mathsf{b},\tau'(\mathsf{b})),(\mathsf{c},\tau'(\mathsf{c})),(\mathsf{d},\tau'(\mathsf{d}))\},\{(\mathsf{a},\mathsf{c}),(\mathsf{d},\mathsf{a})\},\{(\mathsf{a},\mathsf{b})\}\big).$$

QBF^{**} *is thus like* QBF' *but without* e *(depicted in Figure 3.4), giving* $\sigma_{QBF^{**}}(\mathsf{b}) = 2$ *and* $\sigma_{QBF^{**}}(\mathsf{c}) = 4$. *So* b *and* c ***do*** *become strength-consistent after reversing* $\{\mathsf{e}\}$*, whence* $\{\mathsf{e}\}$ ***is*** *a CSI. Clearly, reversing w.r.t.* $\emptyset$ *yields* QBF'*, so that* $\emptyset$ *is not a CSI, and hence* $\{\mathsf{e}\}$ *is also a* $\subset$*-mininmal CSI.*

Lastly, one can check that $\{\mathsf{d}\}$ *is not an SSI and hence cannot be a CSI: as mentioned in Example 1, adding only* d *leaves the final strengths of* b *and* c *unchanged from their initial strengths, so does not explain anything. Thus,* $\{\mathsf{e}\}$ *is the only* $\subset$*-mininmal CSI.*

4. Theoretical Analysis

In this section, we let $QBF = (Args,\tau,Att,Supp)$ and $QBF' = (Args',\tau',Att',Supp')$ be QBAFs, $\mathsf{x},\mathsf{y} \in Args \cap Args'$, and σ be a strength function. We show that both minimal sufficient and counterfactual explanations are sound and complete: either we have strength inconsistency and at least one non-empty set (and no empty set) of explanation arguments or we have strength consistency explained by the empty set (and only by the empty set).

First, if two arguments are strength-consistent given two QBAFs in which they occur and a gradual semantics, then there is no strength inconsistency to explain and the only explanation is the empty set ($SX_{\subset_{\min}}$-soundness).

Proposition 1 ($SX_{\subset_{\min}}$-Soundness)
If $\mathsf{x} \sim \mathsf{y}$*, then* $SX_{\subset_{\min}}(\mathsf{x} \not\sim \mathsf{y}) = \{\emptyset\}$.

Proof. Let $\mathsf{x} \sim \mathsf{y}$. Then $\emptyset$ is an SSI directly by Definition 5. It is clearly $\subset$-minimal, so $\{\emptyset\} \subseteq SX_{\subset_{\min}}(\mathsf{x} \not\sim \mathsf{y})$. On the other hand, no $S \neq \emptyset$ can be an SSI, by definition, precisely because $\mathsf{x} \sim \mathsf{y}$. So $SX_{\subset_{\min}}(\mathsf{x} \not\sim \mathsf{y}) \subseteq \{\emptyset\}$. Hence, $SX_{\subset_{\min}}(\mathsf{x} \not\sim \mathsf{y}) = \{\emptyset\}$ as required. □

If arguments are strength-inconsistent though, then there exists an explanation, but no empty explanation ($SX_{\subset_{\min}}$-completeness).

Proposition 2 ($SX_{\subset_{\min}}$-Completeness)
If $\mathsf{x} \not\sim \mathsf{y}$*, then* $|SX_{\subset_{\min}}(\mathsf{x} \not\sim \mathsf{y})| \geq 1$ *and* $\emptyset \notin SX_{\subset_{\min}}(\mathsf{x} \not\sim \mathsf{y})$.

Proof. Let $\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y}$.
Proof of $|SX_{\subset_{\min}}(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})| \geq 1$. By definition of an SSI, since $\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y}$, any $S \subseteq Args \cup Args'$ is an SSI of x and y (w.r.t. σ, QBF, and QBF') iff $\mathsf{x} \not\sim_{\sigma,QBF,QBF_{\leftarrow QBF'}((Args \cup Args')\setminus S)} \mathsf{y}$. Suppose for a contradiction that such a set S does not exist: $\forall S \subseteq Args \cup Args'$, $\mathsf{x} \sim_{\sigma,QBF,QBF_{\leftarrow QBF'}((Args \cup Args')\setminus S)} \mathsf{y}$. Trivially then, $\mathsf{x} \sim_{\sigma,QBF,QBF_{\leftarrow QBF'}((Args \cup Args')\setminus(Args \cup Args'))} \mathsf{y}$. Since $QBF_{\leftarrow QBF'}(\emptyset) = QBF'$ by definition of QBAF reversal (Definition 4), it follows that $\mathsf{x} \sim_{\sigma,QBF,QBF'} \mathsf{y}$, contradicting $\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y}$. By contradiction, there is at least one $S \in SX_{\subset_{\min}}(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$.
Proof of $\emptyset \notin SX_{\subset_{\min}}(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$. Suppose $\emptyset \in SX(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$ for a contradiction. Since $\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y}$, we have $\mathsf{x} \not\sim_{\sigma,QBF,QBF_{\leftarrow QBF'}((Args \cup Args')\setminus \emptyset)} \mathsf{y}$, by definition of an SSI. As $QBF_{\leftarrow QBF'}(Args \cup Args') = QBF$ by definition of QBAF reversal, it follows that $\mathsf{x} \not\sim_{\sigma,QBF,QBF} \mathsf{y}$. But this is in direct contradiction to the definition of strength consistency (Definition 3). Thus, $\emptyset \notin SX(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$, and hence $\emptyset \notin SX_{\subset_{\min}}(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$. □

We can prove analogous properties for $\subset$-minimal CSIs.

Proposition 3 ($CX_{\subset_{\min}}$-soundness)
If $\mathsf{x} \sim \mathsf{y}$, *then* $CX_{\subset_{\min}}(\mathsf{x} \not\sim \mathsf{y}) = \{\emptyset\}$.

Proof. Let $\mathsf{x} \sim_{\sigma,QBF,QBF'} \mathsf{y}$. By definition, a CSI is an SSI S for which $\mathsf{x} \sim_{\sigma,QBF,QBF_{\leftarrow QBF'}(S)} \mathsf{y}$. Since $QBF_{\leftarrow QBF'}(\emptyset) = QBF'$ and $\mathsf{x} \sim_{\sigma,QBF,QBF'} \mathsf{y}$, we find $\mathsf{x} \sim_{\sigma,QBF,QBF_{\leftarrow QBF'}(\emptyset)} \mathsf{y}$, whence $\emptyset$ is a CSI. Clearly, it is a unique $\subset$-minimal CSI. □

Proposition 4 ($CX_{\subset_{\min}}$-completeness)
If $\mathsf{x} \not\sim \mathsf{y}$, *then* $|CX_{\subset_{\min}}(\mathsf{x} \not\sim \mathsf{y})| \geq 1$ *and* $\emptyset \notin CX_{\subset_{\min}}(\mathsf{x} \not\sim \mathsf{y})$.

Proof. Let $\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y}$.
Proof of $|CX_{\subset_{\min}}(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})| \geq 1$. Consider $S = Args \cup Args'$. First note that $QBF_{\leftarrow QBF'}((Args \cup Args') \setminus (Args \cup Args')) = QBF_{\leftarrow QBF'}(\emptyset) = QBF'$, and so $\mathsf{x} \not\sim_{\sigma,QBF,QBF_{\leftarrow QBF'}((Args \cup Args')\setminus(Args \cup Args'))} \mathsf{y}$. Thus, $Args \cup Args' \in SX(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$. Now, since $QBF_{\leftarrow QBF'}(Args \cup Args') = QBF$ and $\mathsf{x} \sim_{\sigma,QBF,QBF} \mathsf{y}$ holds true by definition, we have that $\mathsf{x} \sim_{\sigma,QBF,QBF_{\leftarrow QBF'}(Args \cup Args')} \mathsf{y}$. Thus, by definition, $Args \cup Args' \in CX(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$, so that the non-empty $CX(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$ must have at least one $\subset$-minimal element. Therefore, $|CX_{\subset_{\min}}(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})| \geq 1$.
Proof of $\emptyset \notin CX_{\subset_{\min}}(\mathsf{x} \not\sim_{\sigma,QBF,QBF'} \mathsf{y})$. Since a $\subset$-minimal CSI is an SSI, if $\emptyset$ were an SSI, then $\emptyset$ would be a $\subset$-minimal SSI, contradicting Proposition 2. □

The above results show that there are non-trivial (i.e. non-empty) sufficient and counterfactual strength inconsistency explanations if and only if a strength inconsistency results between two arguments after an update to a given QBAF. We deem this a desirable property: one needs to explain only if a change in the relative strengths of arguments actually happens after an update; and if there are explanations of changes in the relative strengths of arguments, then they should correctly refer to such changes.

5. Discussion

In this paper, we introduced explanations for changes in the relative strengths of two arguments after a QBAF update; explanations are in the form of sets of arguments that have been changed (added, removed, changed in their initial score or outgoing attacks and supports). Intuitively, a change by means of a set of arguments E provides a sufficient explanation of an alteration in the relative strengths of some arguments of interest if it suffices to change E without making other changes to obtain the alteration in question. Additionally, E is a counterfactual explanation if the absence of change to E would revert back the alteration in the relative strengths of the arguments of interest, even with all the other changes present. Our approach helps to answer a key explainability question – "why b and no longer a?" – in dynamic quantitative bipolar argumentation.

To our knowledge, this is the first paper on explainability in quantitative bipolar argumentation. Our explanations are immediately applicable to quantitative (non-bipolar) argumentation, where explainability has not been researched either, with the exception of [5]. There, the authors formalise a notion of *impact* of an argument on the final strength of another argument, roughly as a difference between the final strengths of the latter argument with and without the former argument being present. We instead consider as explanations the changes to arguments that guarantee alterations in the relative strengths of other arguments after a given update to the quantitative argumentation framework.

More generally, our work is positioned at the intersection of argumentation dynamics and explainable argumentation, both of which have been studied in depth: see [6] for a survey on argumentation dynamics, as well as [7] and [8] for surveys on argumentation and explainability. Few works study the intersection of dynamics and explainability *explicitly*. A notable exception is [9], where we studied, in the context of (admissibility-based) abstract argumentation, how the violation of monotony of entailment can be explained in so-called *normal expansion* scenarios, in which new arguments are added to an argumentation framework, but the relation among previously existing arguments remains unchanged. The present work is different in that it i) addresses QBAFs, and ii) explains strength inconsistency (i.e. change in preferences from a decision-theoretical perspective) rather than the violation of monotony of entailment.

However, several argumentation explainability approaches consider dynamics *implicitly*. For instance, assuming some space of modifications in a given argumentation framework, the modifications that would change some topic argument's acceptability status (or strength) can be seen as explanations of such a change [10,11,12]. In particular, a collection of additions or removals of arguments or attacks in an abstract argumentation framework in a way that changes the acceptability of a specific argument is an explanation in e.g. [13,12]. Relatedly, though not directly concerning changes, [14,15,16] define explanations, roughly speaking, as sets of arguments (in non-quantitative argumentation frameworks) that are sufficient for acceptance or rejection of some target argument(s).

Our work is on QBAFs instead, concerning gradual semantics and changes to numerical argument strengths. We also defined counterfactual explanations, rather than necessary ones: for comparison, in Example 1, neither a nor e could be said to be necessary explanations, because changing neither one alone is needed for strength inconsistency; rather, e is counterfactual in that the absence of its change guarantees strength consistency back, given all other changes. Collectively, $\{\mathsf{a}, \mathsf{e}\}$ could be said to be necessary, as changing at least one element therein is needed in any combination of changes that leads to strength inconsistency. We leave formal investigations of this for future work. In the future we can also expand the current perspective on QBAF (change) explainability by in addition providing sub-graphs to trace sets of explanation arguments to topic arguments.

References

[1] M.J. Osborne and A. Rubinstein, *Models in Microeconomic Theory*, Open Book Publishers, 2020. doi:10.11647/OBP.0204.

[2] P. Baroni, A. Rago and F. Toni, From Fine-Grained Properties to Broad Principles for Gradual Argumentation: A Principled Spectrum, *International Journal of Approximate Reasoning* **105** (2019), 252–286. doi:10.1016/j.ijar.2018.11.019.

[3] I. Stepin, J.M. Alonso, A. Catala and M. Pereira-Farina, A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence, *IEEE Access* **9** (2021), 11974–12001. doi:10.1109/ACCESS.2021.3051315.

[4] N. Potyka, Extending Modular Semantics for Bipolar Weighted Argumentation, in: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2019, pp. 1722–1730–. ISBN ISBN 9781450363099.

[5] J. Delobelle and S. Villata, Interpretability of Gradual Semantics in Abstract Argumentation, in: *15th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Vol. 11726 LNAI, G. Kern-Isberner and Z. Ognjanovic, eds, Springer, Belgrade, 2019, pp. 27–38. ISSN 16113349. ISBN ISBN 9783030297640.

[6] S. Doutre and J.-G. Mailly, Constraints and changes: A survey of abstract argumentation dynamics, *Argument & Computation* **9** (2018), 223–248. ISBN ISBN 1946-2174. doi:10.3233/AAC-180425.

[7] A. Vassiliades, N. Bassiliades and T. Patkos, Argumentation and explainable artificial intelligence: a survey, *The Knowledge Engineering Review* **36** (2021), e5. doi:10.1017/S0269888921000011.

[8] K. Čyras, A. Rago, E. Albini, P. Baroni and F. Toni, Argumentative XAI: A Survey, in: *30th International Joint Conference on Artificial Intelligence*, Z.-H. Zhou, ed., IJCAI, Montreal, 2021, pp. 4392–4399. ISBN ISBN 978-0-9992411-9-6. doi:10.24963/ijcai.2021/600.

[9] T. Kampik and K. Čyras, Explanations of Non-Monotonic Inference in Admissibility-based Abstract Argumentation, in: *Logic and Argumentation (to appear)*, P. Baroni, C. Benzmüller and Y.N. Wáng, eds, Springer International Publishing, Cham, 2020.

[10] T. Wakaki, K. Nitta and H. Sawamura, Computing Abductive Argumentation in Answer Set Programming, in: *6th International Workshop on Argumentation in Multi-Agent Systems*, P. McBurney, I. Rahwan, S. Parsons and N. Maudet, eds, Springer, Budapest, 2009, pp. 195–215. doi:10.1007/978-3-642-12805-9_12.

[11] R. Booth, D.M. Gabbay, S. Kaci, T. Rienstra and L. van der Torre, Abduction and Dialogical Proof in Argumentation and Logic Programming, in: *21st European Conference on Artificial Intelligence*, T. Schaub, G. Friedrich and B. O'Sullivan, eds, Frontiers in Artificial Intelligence and Applications, Vol. 263, IOS Press, Prague, 2014, pp. 117–122. doi:10.3233/978-1-61499-419-0-117.

[12] C. Sakama, Abduction in Argumentation Frameworks, *Journal of Applied Non-Classical Logics* **28**(2–3) (2018), 218–239. doi:10.1080/11663081.2018.1487241.

[13] X. Fan and F. Toni, On Computing Explanations for Non-Acceptable Arguments, in: *Theory and Applications of Formal Argumentation - 3rd International Workshop*, E. Black, S. Modgil and N. Oren, eds, Lecture Notes in Computer Science, Vol. 9524, Springer, Buenos Aires, 2015, pp. 112–127. doi:10.1007/978-3-319-28460-6_7.

[14] Z.G. Saribatur, J.P. Wallner and S. Woltran, Explaining Non-Acceptability in Abstract Argumentation, in: *24th European Conference on Artificial Intelligence*, G.D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín and J. Lang, eds, IOS Press, Santiago de Compostela, 2020, pp. 881–888. doi:10.3233/FAIA200179.

[15] M. Ulbricht and J.P. Wallner, Strong Explanations in Abstract Argumentation, in: *35th Conference on Artificial Intelligence*, AAAI, 2021, pp. 6496–6504.

[16] A. Borg and F. Bex, Necessary and Sufficient Explanations for Argumentation-Based Conclusions, in: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, J. Vejnarová and N. Wilson, eds, Springer International Publishing, Cham, 2021, pp. 45–58. ISBN ISBN 978-3-030-86772-0.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220153

How Complex Is the Strong Admissibility Semantics for Abstract Dialectical Frameworks?

Atefeh KESHAVARZI ZAFARGHANDI [b,a,1], Wolfgang DVOŘÁK [c],
Rineke VERBRUGGE [b] and Bart VERHEIJ [b]

[a] *Human-Centered Data Analytics, Centrum Wiskunde & Informatica, The Netherlands*
[b] *Department of Artificial Intelligence, Bernoulli Institute,*
University of Groningen, The Netherlands
[c] *Institute of Logic and Computation, TU Wien, Austria*

Abstract. Abstract dialectical frameworks (ADFs) have been introduced as a formalism for modeling and evaluating argumentation allowing general logical satisfaction conditions. Different criteria used to settle the acceptance of arguments are called semantics. Semantics of ADFs have so far mainly been defined based on the concept of admissibility. Recently, the notion of strong admissibility has been introduced for ADFs. In the current work we study the computational complexity of the following reasoning tasks under strong admissibility semantics. We address 1. the credulous/skeptical decision problem; 2. the verification problem; 3. the strong justification problem; and 4. the problem of finding a smallest witness of strong justification of a queried argument.

Keywords. argumentation, abstract dialectical frameworks, complexity

1. Introduction

Despite the fact that Dung's abstract argumentation frameworks [1] (AFs for short) are widely used and studied within AI, in certain scenarios AFs are too limited to properly model the complex relations between arguments. Thus, several generalizations of AFs have been introduced [2], e.g., SETAFs and Bipolar AFs. Abstract dialectical frameworks (ADFs) [3,4,5] are an expressive generalization of AFs that can represent logical relations among arguments and subsume many popular generalizations of AFs. Semantics of AFs and ADFs single out coherent subsets of arguments that fit together, according to specific criteria [6].

There are several established semantics for AFs and ADFs. In this work we consider strong admissibility semantics and grounded semantics, which are the most skeptical types of semantics. Characteristics of *grounded* semantics for AFs include that 1. each AF has a unique grounded extension; 2. the grounded extension collects all the arguments about which no one doubts their acceptance; 3. the grounded extension is often a subset

[1] Corresponding Author, E-mail: a.keshavarzi.zafarghandi@rug.nl

of the set of extensions of other types of AF semantics. Thus, it is important to investigate whether an argument belongs to the grounded extension of a given AF. The notion of *strong admissibility* is introduced for AFs to answer the query ‘Why does an argument belong to the grounded extension?’.

While the grounded extension collects all the arguments of a given AF that can be accepted without any doubt, a strongly admissible extension provides a (minimal) justification why specific arguments can be accepted without any doubt, i.e., belong to the grounded extension. Thus, the strong admissibility semantics can be the basis for an algorithm that can be used not only for answering the credulous decision problem but also for human-machine interaction that requires an explainable outcome (cf. [7,8]).

In AFs, strong admissibility semantics were first defined in the work of Baroni and Giacomin [9], and later in [10]. Furthermore, in [11], Caminada and Dunne presented a labelling account of strong admissibility to answer the decision problems of AFs under grounded semantics. Moreover, Caminada showed in [12,10] that strong admissibility plays a crucial role in discussion games for AFs under grounded semantics. This motivated the study of the computational complexity of strong admissibility of AFs in general and in particular of the problem of computing small strongly admissible sets that justify the acceptance of an argument [13,14].

In previous work, we generalized the concept of strong admissibility to ADFs [15]. This concept fulfils properties that are related to those of the strong admissibility semantics for AFs, as follows: 1. Each ADF has at least one strongly admissible interpretation. 2. The set of strongly admissible interpretations of ADFs forms a lattice with as least element the trivial interpretation and as maximum element the grounded interpretation. 3. The strong admissibility semantics can be used to answer whether an argument is justifiable under grounded semantics. 4. The strong admissibility semantics for ADFs is a proper generalization of the strong admissibility semantics for AFs.

Whereas several fundamental properties of strong admissibility semantics for ADFs have been established, the computational complexity under strong admissibility semantics has not previously been studied. The current work closes this gap by studying the complexity of the central reasoning tasks under the strong admissibility semantics of ADFs. The paper is organised as follows: In Section 2 we recall the basic definitions of ADFs and strong admissibility. In Section 3 we provide exact complexity classifications for the different decision problems for strong admissibility semantics. We consider standard decision problems, i.e., the credulous and skeptial decision problems and the verification problem, the strong justification problem, i.e., deciding whether an argument is strongly justified in an interpretation, and the problem of finding a small witness of strong justification of an argument, i.e, whether there exists a strongly admissible interpretation that satisfies a queried argument and is smaller than a given bound. Finally, we conclude in Section 4. [2]

[2] This paper is based on an earlier presentation at the non-archival workshop NMR 2021. Proofs of all theorems are available in the dissertation [16] (Chapter 4), see `https://research.rug.nl/nl/publications/abstract-dialectical-frameworks-semantics-discussion-games-and-va`.

2. Formal Background

We recall the basics of ADFs [5]. Also we recall the definition of strong admissibility for ADFs, presented in [17].

2.1. Abstract Dialectical Frameworks

We summarize key concepts of abstract dialectical frameworks [3,5].

Definition 1. An abstract dialectical framework (ADF) is a tuple $D = (A, L, C)$ where: 1. A is a finite set of arguments (statements, positions); 2. $L \subseteq A \times A$ is a set of links among arguments; 3. $C = \{\varphi_a\}_{a \in A}$ is a collection of propositional formulas over arguments, called acceptance conditions.

An ADF can be represented by a graph in which nodes indicate arguments and links show the relations between arguments. Each argument a in an ADF is labelled by a propositional formula, called acceptance condition, φ_a over $par(a)$, where $par(a) = \{b \mid (b,a) \in L\}$. The acceptance condition of each argument clarifies under which condition the argument can be accepted.

A *three-valued interpretation* v (for D) is a function $v : A \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ that maps arguments to one of the three truth values true ($\mathbf{t}$), false ($\mathbf{f}$), or undecided ($\mathbf{u}$). For reasons of brevity, we will shorten the notation of three-valued interpretation $v = \{a_1 \mapsto t_1, \ldots, a_m \mapsto t_m\}$ as follows: $v = \{a_i \mid v(a_i) = \mathbf{t}\} \cup \{\neg a_i \mid v(a_i) = \mathbf{f}\}$. For instance, $v = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}\} = \{\neg a, b\}$. Interpretation v is called *trivial*, and v is denoted by $v_{\mathbf{u}}$, if $v(a) = \mathbf{u}$ for each $a \in A$. Furthermore, v is called a *two-valued interpretation* if for each $a \in A$ either $v(a) = \mathbf{t}$ or $v(a) = \mathbf{f}$.

Truth values can be ordered via the information ordering relation $<_i$ given by $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ and no other pair of truth values are related by $<_i$. Relation $\leq_i$ is the reflexive closure of $<_i$. Meet operator $\sqcap_i$ is defined over the truth values such that $\mathbf{t} \sqcap_i \mathbf{t} = \mathbf{t}$ and $\mathbf{f} \sqcap_i \mathbf{f} = \mathbf{f}$, while it returns $\mathbf{u}$ otherwise. The meet of two interpretations v and w is then defined as $(v \sqcap_i w)(a) = v(a) \sqcap_i w(a)$ for all $a \in A$.

Given an interpretation v (for D), the partial valuation of φ_a by v is $v(\varphi_a) = \varphi_a^v = \varphi_a[b/\top : v(b) = \mathbf{t}][b/\bot : v(b) = \mathbf{f}]$, for $b \in par(a)$. Note that in this work we assume that $D = (A, L, C)$ is a finite ADF and v is an interpretation of D. Semantics for ADFs can be defined via the *characteristic operator* Γ_D, presented in Definition 2.

Definition 2. Let D be an ADF and let v be an interpretation of D. Applying Γ_D on v leads to v' such that for each $a \in A$, $v'(a) = \mathbf{t}$ if φ_a^v is irrefutable, $v'(a) = \mathbf{f}$ if φ_a^v is unsatisfiable, and $v'(a) = \mathbf{u}$, otherwise.

Most types of semantics for ADFs are based on the concept of admissibility. An interpretation v for a given ADF F is called *admissible* iff $v \leq_i \Gamma_F(v)$; it is *preferred* iff v is $\leq_i$-maximal admissible; it is the *grounded* interpretation of D iff v is the least fixed point of Γ_D. The set of all σ interpretations for an ADF D is denoted by $\sigma(D)$, where $\sigma \in \{adm, grd, prf\}$ abbreviates the different semantics in the obvious manner. Given an interpretation v and an argument $a \in A$, a is called *acceptable* with respect to v if φ_a^v is irrefutable and a is called *deniable* with respect to v if φ_a^v is unsatisfiable.

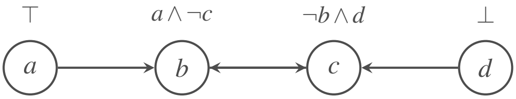


Figure 1. ADF of Example 1

2.2. The Strong Admissibility Semantics for ADFs

In this section, we rephrase the concept of strong admissibility semantics for ADFs from [15], which is defined based on the notion of strongly justifiable arguments (i.e., strongly acceptable/deniable arguments). Below, the interpretation $v_{|_P}$ is equal to $v(p)$ for any $p \in P$, and returns $\mathbf{u}$ otherwise, i.e., $v_{|_P} = v_{\mathbf{u}}|^{p \in P}_{v(p)}$.

Definition 3. Let D be an ADF. Argument a is a *strongly justified* argument in interpretation v with respect to set E if one of the following conditions holds:

- $v(a) = \mathbf{t}$ and there exists a subset P of parents of a excluding E, namely $P \subseteq par(a) \setminus E$, such that (a) a is acceptable with respect to $v_{|_P}$ and (b) all $p \in P$ are strongly justified in v w.r.t. set $E \cup \{p\}$.
- $v(a) = \mathbf{f}$ and there exists a subset P of parents of a excluding E, namely $P \subseteq par(a) \setminus E$, such that (a) a is deniable with respect to $v_{|_P}$ and (b) all $p \in P$ are strongly justified in v w.r.t. set $E \cup \{p\}$.

An argument a is *strongly acceptable*, respectively *strongly deniable*, in v if $v(a) = \mathbf{t}$, resp. $v(a) = \mathbf{f}$, and a is strongly justified in v with respect to set $\{a\}$. We say that a is *strongly justified* in v if it is either strongly acceptable or strongly deniable in v.

Note that in Definition 3, E is used to keep track of the arguments that cannot be used to justify a. We say that *a is not strongly justified in an interpretation v* if there is no set of parents of a that satisfies the conditions of Definition 3 for a. Example 1 presents the notion of strongly justified arguments in an interpretation.

Example 1. *Let D be the ADF depicted in Figure 1. Let $v = \{b, \neg c, \neg d\}$. First, since $\varphi_d^{v_{\mathbf{u}}} \equiv \bot$, it holds that d is strongly deniable in v. We show that c is strongly deniable in v with respect to $E = \{c\}$. Let $P = \{d\}$; it is clear that $\varphi_c^{v_{|_P}}$ is unsatisfiable. That is, c is deniable w.r.t. $v_{|_d}$. Then, since $d \in P$, $v(d) = \mathbf{f}$ and d is strongly justified in v with respect to $E = \{d\}$, c is strongly deniable in v. We show that b is not strongly acceptable in v. Let $P = par(b)$. Since $\varphi_b^{v_{|_P}} \not\equiv \top$, there is no subset of $par(b)$ that satisfies the conditions of Definition 3 for b. Thus, b is not strongly acceptable in v. It will turn out that v is not a strongly admissible interpretation, see Definition 4.*

Definition 4. Let D be an ADF. An interpretation v is a *strongly admissible* interpretation if for each a such that $v(a) = \mathbf{t}/\mathbf{f}$, it holds that a is a strongly justified argument in v. The set of all strongly admissible interpretations of D is denoted by $sadm(D)$.

Consider again the ADF of Example 1. Let $v = \{b, \neg c, \neg d\}$. As shown in Example 1, c and d are strongly justified in v. However, b is not strongly justified in v. Thus, $v \notin$

$sadm(D)$. However, for instance, $v_1 = \{a\}$, $v_2 = \{\neg c, \neg d\}$ and $v_3 = \{a, b, \neg c, \neg d\}$ are strongly admissible interpretations of D. Furthermore, $v_3 \in grd(D)$.

Algorithms in Section 5 of [17] answer the verification problem under strong admissibility semantics and the strong justification problem. To present the algorithms, Definition 28 in [17] introduces a variant of the characteristic operator restricted to a given interpretation v; we rewrite it in Definition 5.

Definition 5. Let D be an ADF and let v, w be interpretations of D. We define $\Gamma^0_{D,v}(w) = w$ and $\Gamma_{D,v}(w) = \Gamma_D(w) \sqcap_i v$, where $\Gamma^j_{D,v}(w) = \Gamma_{D,v}(\Gamma^{j-1}_{D,v}(w))$ for j with $j \geq 1$.

The sequence of interpretations $\Gamma^j_{D,v}(v_{\mathbf{u}})$ as defined in Definition 5 is named the sequence of strongly admissible interpretations constructed based on v in D. Theorems 28 and 29 in [17] show that one can use iterative fixed-point computations of $\Gamma_{D,v}$ operators to decide (a) verification of a given strongly admissible interpretation and (b) whether an argument is strongly acceptable/deniable within a given interpretation. However, because testing whether an argument is acceptable in Γ_D is already NP/coNP-hard [18], these procedures are in P^{NP}. As we will show, both problems allow for algorithms of significantly lower complexity.

3. Computational Complexity

We analyse the complexity under strong admissibility semantics for (a) the standard reasoning tasks of ADFs [18] and (b) two problems specific to strong admissibility semantics: (i) the small witness problem introduced for AFs [14,13] in order to minimize the length of the corresponding discussion games; and (ii) the strong justification problem. For a given ADF D, argument a and the truth value $x \in \{\mathbf{t}, \mathbf{f}\}$, we consider the following problems:

1. *The credulous decision problem*: whether a is credulously justifiable w.r.t. the strong admissibility semantics of D, denoted as $Cred_{sadm}(a,x,D)$, where $Cred_{sadm}(a,x,D) =$ yes if there exists $v \in sadm(D)$ s.t. $v(a) = x$, and it returns *no* otherwise.
2. *The skeptical decision problem*: whether a is skeptically justified w.r.t. the strong admissibility semantics of D, denoted as $Skept_{sadm}(a,x,D)$, where $Skept_{sadm}(a,x,D)$ $=$ yes if for each $v \in sadm(D)$ it holds that $v(a) = x$, and it returns *no* otherwise.
3. *The verification problem*: whether $v \in sadm(D)$ denoted by $Ver_{sadm}(v,D)$, where $Ver_{sadm}(v,D) =$ yes if $v \in sadm(D)$, and it returns *no* otherwise.
4. *The strong justification problem:* The problem whether a given argument a is strongly justified in a given interpretation v, denoted as $StrJust(a,x,v,D)$, where $StrJust(a,x,v,D) =$ yes if a is strongly justified in v, and it returns *no* otherwise.
5. *The small witness problem:* We are interested in computing a strongly admissible interpretation that has the least information of the ancestors of a given argument, namely a, where $v(a) = x$. The decision version of this problem is the k-Witness problem, denoted by k-$Witness_{sadm}$, indicating whether a given argument is strongly justified in at least one v such that $v \in sadm(D)$ and $|v^{\mathbf{t}} \cup v^{\mathbf{f}}| \leq k$. Note that k is part of the input of this problem. This decision problem is presented formally as follows: k-$Witness_{sadm}(a,x,D) =$ yes if there exists $v \in sadm(D)$ such that $v(a) = x$ and $|v^{\mathbf{t}} \cup v^{\mathbf{f}}| \leq k$, and it returns *no* otherwise.

3.1. The Credulous/Skeptical Decision Problems

In this section we show the complexity of deciding whether an argument in question is credulously/skeptically justifiable in at least one/all strongly admissible interpretation(s) of a given ADF. We show that $Cred_{sadm}$ is coNP-complete and $Skept_{sadm}$ is trivial. To this end, we use the fact, presented in [17], that the set of strongly admissible interpretations of a given ADF D forms a lattice with respect to the $\leq_i$-ordering, with the maximum element being $grd(D)$. Thus, any strongly admissible interpretation of D has at most an amount of information equal to $grd(D)$. Thus, answering the credulous decision problem under the strong admissibility semantics coincides with answering the credulous decision problem under the grounded semantics.

Theorem 1. *$Cred_{sadm}$ is coNP-complete.*

Proof. We have that $Cred_{sadm}(a,x,D) = Cred_{grd}(a,x,D)$ and the latter has been shown to be coNP-complete in [19, Proposition 4.1.3.]. □

Concerning skeptical acceptance, notice that the trivial interpretation is the least strongly admissible interpretation in each ADF. Thus, $Skept_{sadm}(a,x,D)$ is trivially *no*.

Theorem 2. *$Skept_{sadm}$ is a trivial problem.*

3.2. The Verification Problem

In this section, we settle the complexity of $Ver_{sadm}(v,D)$. We have mentioned at the end of Section 2.2 that this problem can be solved in P^{NP}; in the sequel, we will show that its complexity is in fact lower. We first sketch a simple translation-based approach that reduces the verification problem of strongly admissible semantics to the verification problem of grounded semantics. In order to reduce $Ver_{sadm}(v,D)$ to $Ver_{grd}(v,D')$, we modify the acceptance conditions φ_a of D to $\varphi'_a = \neg a$ if $v(a) = \mathbf{u}$ and $\varphi'_a = \varphi_a$ otherwise. We then have that $v \in sadm(D)$ iff $v \in grd(D')$, so that we can use the DP procedure for $Ver_{grd}(v,D')$ [19, Theorem 4.1.4]. However, as we will discuss next, $Ver_{sadm}(v,D)$ can even be solved within coNP.

Intuitively, since the grounded interpretation is the maximum element of the lattice of strongly admissible interpretations and the credulous decision problem under grounded semantics is coNP-complete, it seems that the verification problem under the strong admissibility semantics has to be coNP-complete. However, having the positive answer for $Cred_{grd}(a,x,D)$ for each a with $v(a) = \mathbf{t}/\mathbf{f}$ does not lead to the positive answer of $Ver_{sadm}(v,D)$. This is because $v \leq_i grd(D)$ does not imply that v is a strongly admissible interpretation of D (see Example 2 below).

Example 2. *Let $D = (\{a,b\},\{\varphi_a : \top, \varphi_b : a \vee b\})$. The grounded interpretation of D is $\{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}\}$. Furthermore, the interpretation $v = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{t}\}$ is an admissible interpretation of D such that $v \leq_i grd(D)$. However, v is not a strongly admissible interpretation of D. As we know, the answer of $Cred_{grd}(b,\mathbf{t},D)$ is* yes*, but b is not strongly acceptable in v. Thus, the answer to $Ver_{sadm}(v,D)$ is* no.

To show that Ver_{sadm} is coNP-complete, we modify and combine both the fixed-point iteration from [17] and the grounded algorithm from [19]. To this end, we need some auxiliary results that are shown in Lemmas 1 and 2.

Lemma 1. *Given an ADF D with $|A| = n$, the following statements are equivalent:*

1. *v is a strongly admissible interpretation of D;*
2. *$v = \Gamma^n_{D,v}(v_{\mathbf{u}})$;*
3. *for each $w \leq_i v$, it holds that $v = \Gamma^n_{D,v}(w)$.*

In the following, let $v^* = v^{\mathbf{t}} \cup v^{\mathbf{f}}$. The notions of completion of an interpretation and model are presented in Definition 6; they are used in Lemma 2.

Definition 6. Let w be an interpretation. We define the *completion* of w, denoted by $[w]_2$, as follows: $[w]_2 = \{u \mid w \leq_i u \textit{ and } u \textit{ is a two-valued interpretation}\}$.

Furthermore, a two-valued interpretation u is said to be a *model* of formula φ, if $u(\varphi) = \mathbf{t}$, denoted by $u \models \varphi$.

Lemma 2. *Let D be an ADF and let v be an interpretation of D. Then $v \notin sadm(D)$ iff there exists an interpretation w of D that satisfies all the following conditions:*

1. *$w <_i v$;*
2. *For each $a \in w^{\mathbf{u}} \cap v^{\mathbf{t}}$ there exists $u_a \in [w]_2$ s.t. $u_a \not\models \varphi_a$;*
3. *For each $a \in w^{\mathbf{u}} \cap v^{\mathbf{f}}$ there exists $u_a \in [w]_2$ s.t. $u_a \models \varphi_a$.*

Proof. $\Leftarrow$: Assume that v and w are interpretations of D that satisfy all of the items 1, 2, 3 presented in the lemma. We show that $v \notin sadm(D)$. Toward a contradiction, assume that $v \in sadm(v)$. Let a be an argument such that $a \in w^{\mathbf{u}} \cap v^{\mathbf{t}}$; thus, since w satisfies the conditions of the lemma, it holds that there exists $u_a \in [w]_2$ such that $u_a \not\models \varphi_a$, i.e., $u_a(a) = \mathbf{f}$. Furthermore, since $v(a) = \mathbf{t}$ and $v \in sadm(D)$, for any $j \in [v]_2$ it holds that $j \models \varphi_a$. Since $w <_i v$, it holds that $j \in [w]_2$, i.e., $\Gamma_D(w)(a) = \mathbf{u}$. The proof method for the case that $a \in w^{\mathbf{u}} \cap v^{\mathbf{f}}$ is similar, i.e., if $a \in w^{\mathbf{u}} \cap (v^{\mathbf{t}} \cup v^{\mathbf{f}})$, then $\Gamma_D(w)(a) = \mathbf{u}$. Thus, for $a \in w^{\mathbf{u}} \cap v^*$ we have $\Gamma_{D,v}(w)(a) = (\Gamma_D(w) \sqcap v)(a) = \mathbf{u}$. In other words, $\Gamma_{D,v}(w)(a) \leq_i w$ and thus, by the monotonicity of $\Gamma_{D,v}(w)$, also $\Gamma^n_{D,v}(w)(a) \leq_i w <_i v$. Thus, since $\Gamma^n_{D,v}(w) \not\prec_i v$, the third item of Lemma 1 does not hold for w with $w <_i v$. Therefore, $v \notin sadm(D)$.
$\Rightarrow$: Assume that $v \notin sadm(D)$. That is, for the fixed point $w = \Gamma^n_{D,v}(v_{\mathbf{u}})$ we have $w <_i v$. Consider $a \in w^{\mathbf{u}} \cap v^{\mathbf{t}}$. Because w is a fixed point, we have that $\Gamma_{D,v}(w)(a) \neq \mathbf{t}$ and thus $\Gamma_D(w) \neq \mathbf{t}$. That is, there is a $u_a \in [w]_2$ such that $u_a \not\models \varphi_a$. Similar reasoning applies to $a \in w^{\mathbf{u}} \cap v^{\mathbf{f}}$. □

Theorem 3 shows that Ver_{sadm} is coNP-complete for ADFs.

Theorem 3. *Ver_{sadm} is* coNP*-complete for ADFs.*

Proof sketch. We first show that $Ver_{sadm} \in$ coNP for ADFs. Let D be an ADF and let v be an interpretation of D. For membership, consider the co-problem. By Lemma 2, if there exists an interpretation of w that satisfies the condition of Lemma 2, then $v \notin sadm(D)$. Thus, guess an interpretation w, together with an interpretation $u_a \in [w]_2$ for each $a \in v^*$, and check whether they satisfy the conditions of Lemma 2. Note that since $w <_i v$, we have to check the second and the third items of Lemma 2 a total of $|v^* \setminus w^{\mathbf{u}}|$ number of times. That is, this checking has to be done at most $|v^*|$ number of times when w is the trivial interpretation. Thus, this checking step is linear in the size of v^*.

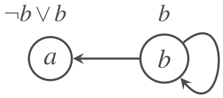


Figure 2. Reduction used in Theorems 3 and 5, for $\psi = \neg b \vee b$.

Therefore, the procedure of guessing of w and checking whether it satisfies $1,2,3$ of Lemma 2 is an NP-problem. Thus, if a w satisfies the items of Lemma 2, then the answer to $Ver_{sadm}(v,D)$ is *no*. Otherwise, if we check all interpretations w such that $w <_i v$ and none of them satisfies the conditions of Lemma 2, then the answer to $Ver_{sadm}(v,D)$ is *yes*. Thus, $Ver_{sadm}(v,D) \in$ coNP.

Now let us show that Ver_{sadm} is coNP-hard. For hardness of Ver_{sadm}, we consider the standard propositional logic problem of VALIDITY. Let ψ be an arbitrary Boolean formula and let $X = atom(\psi)$ be the set of atoms in ψ. Let a be a new atom, i.e., $a \notin X$. Construct ADF $D = (\{X \cup \{a\}\}, L, C)$ where $\varphi_x : x$ for each $x \in X$ and $\varphi_a : \psi$. We show that ψ is valid if and only if $v = v_{\mathbf{u}}|^a_{\mathbf{t}}$ is a strongly admissible interpretation of D. An illustration of the reduction for the formula $\psi = \neg b \vee b$ to the ADF $D = (\{a,b\}, L, \varphi_a : \psi, \varphi_b : b)$ is shown in Figure 2. One can show that ψ is a valid formula iff v is the grounded interpretation of D. □

3.3. Strong Justification of an Argument

Note that it is possible that an interpretation v contains some strongly justified arguments but v is not strongly admissible itself. For instance, in interpretation $v = \{b, \neg c, \neg d\}$, presented in Example 1, arguments c and d are strongly deniable in v, however, argument b is not strongly acceptable in v. Thus, v is not a strongly admissible interpretation of D. However, there exists a strongly admissible interpretation of D in which c and d are strongly acceptable and that has less information than v, namely, $v' = \{c,d\}$. Thus, the problem $StrJust(a,x,v,D)$ of deciding whether an argument is strongly justified in a given interpretation of an ADF is different from the previously discussed decision problems. We show that *StrJust* is coNP-complete.

Algorithm 2 in [17] presents a direct method of deciding whether a is strongly justified in an interpretation v. That is, a is strongly acceptable/deniable in v iff it is acceptable/deniable by the least fixed point of the operator $\Gamma_{D,v}$, which is equal to $\Gamma^n_{D,v}(v_{\mathbf{u}})$ for sufficiently large n.

However, the repeated evaluation of Γ_D is a costly part of this algorithm and results in a P^{NP} algorithm. We will next discuss a more efficient method to answer this reasoning task. To this end, we translate a given ADF D to ADF D', presented in Definition 7, such that the queried argument is strongly justifiable in a given interpretation of D if and only if it is credulously justifiable in the grounded interpretation of D'. As shown in Proposition 4.1.3 in [19], $Cred_{grd} \in$ coNP. Thus, verifying whether a given argument is strongly justified in an interpretation is a coNP-problem, since the translation can be done in polynomial time with respect to the size of D. In the following, assume that v is an interpretation of D.

Definition 7. Let $D = (A,L,C)$ be an ADF and let v be an interpretation of D. The translation of D under v is $D' = (A', L', C')$ such that $A' = A \cup \{x,y\}$ where $x,y \notin A$.

Furthermore, for each $a \in A'$ we define the acceptance condition of a in D', namely φ'_a, as follows: 1. $\varphi'_x : x$; 2. $\varphi'_y : y$; 3. if $v(a) = \mathbf{u}$, then $\varphi'_a : \neg a$; 4. if $v(a) = \mathbf{t}$, then $\varphi'_a = \varphi_a \vee x$; 5. if $v(a) = \mathbf{f}$, then $\varphi'_a = \varphi_a \wedge y$.

Notice that we introduced arguments x, y to ensure that arguments in v^* are not assigned to the opposite truth value during the iteration of $\Gamma_{D'}$ that leads to $grd(D')$. Theorem 4 shows the correctness of the reduction.

Theorem 4. *Let D be an ADF, let v be an interpretation of D, and let D' be the translation of D, via Definition 7. Then, $StrJust(a,x,v,D) = yes$, iff $Cred_{grd}(a,x,D') = yes$.*

Proof. We assume that $StrJust(a,\mathbf{t},v,D)$ = yes, and we show that $Cred_{grd}(a,\mathbf{t},D') =$ yes. The proof for the case that $StrJust(a,\mathbf{f},v,D)$ = yes is similar. Assume that $v_{\mathbf{u}}$ is the trivial interpretation of D and $v'_{\mathbf{u}}$ is the trivial interpretation of D'. Assume that $\Gamma^i_{D,v}(v_{\mathbf{u}})$ is a sequence of strongly admissible interpretations constructed based on v in D, as in Definition 5. Let w be the limit of the sequence of $\Gamma^i_{D,v}(v_{\mathbf{u}})$.

$StrJust(a,\mathbf{t},v,D) = yes$ implies that $w(a) = \mathbf{t}$. Since $w \in sadm(D)$, it holds that $g(a) = \mathbf{t}$ where $g \in grd(D)$, i.e., there exists a natural number n such that $\Gamma^n_D(v_{\mathbf{u}})(a) = \mathbf{t}$. By induction on n, it is easy to show that $\Gamma^n_{D'}(v'_{\mathbf{u}})(a) = \mathbf{t}$. That is, $g'(a) = \mathbf{t}$ where $g' \in grd(D')$. Thus, $Cred_{grd}(a,\mathbf{t},D') =$ yes.

We assume that $Cred_{grd}(a,x,D') = yes$, and we show that $StrJust(a,x,v,D) = yes$. Assume that a is justified in the grounded interpretation of D', namely w. Thus, there exists a j such that $w = \Gamma^j_{D'}(w_{\mathbf{u}})$ for $j \geq 0$, where $w_{\mathbf{u}}$ is the trivial interpretation of D'. By induction we prove the claim that for all i, if $a \mapsto \mathbf{t}/\mathbf{f} \in \Gamma^i_{D'}(w_{\mathbf{u}})$, then a is strongly justified in v.

Base case: Assume that $\Gamma^1_{D'}(w_{\mathbf{u}})(a) \in \{\mathbf{t},\mathbf{f}\}$. By the acceptance conditions of x and y in D', both of them are assigned to $\mathbf{u}$ in w. Then it has to be the case that either $\varphi'_a = \varphi_a \vee x$ or $\varphi'_a = \varphi_a \wedge y$ in D'. Thus, $\Gamma^1_{D'}(w_{\mathbf{u}})(a) \in \{\mathbf{t},\mathbf{f}\}$ implies that ${\varphi'_a}^{w_{\mathbf{u}}} \equiv \top/\bot$. Thus, $w(x/y) = \mathbf{u}$, $\varphi'_a = \varphi_a \vee x/\varphi_a \wedge y$ and ${\varphi'_a}^{w_{\mathbf{u}}} \equiv \top/\bot$ together imply that ${\varphi_a}^{w_{\mathbf{u}}} \equiv \top/\bot$. Hence, $\varphi_a^{v_{\mathbf{u}}} \equiv \top/\bot$ where $v_{\mathbf{u}}$ is the trivial interpretation of D. That is, a is strongly justified in v.

Induction hypothesis: Assume that for all j with $1 \leq j \leq i$, if $a \mapsto \mathbf{t}/\mathbf{f} \in \Gamma^j_{D'}(w_{\mathbf{u}})$, then a is strongly justified in v.

Inductive step: We show that if $a \mapsto \mathbf{t}/\mathbf{f} \in \Gamma^{i+1}_{D'}(w_{\mathbf{u}})$, then a is strongly justified in v. Because $x/y \mapsto \mathbf{u} \in w$, we have that $\varphi_a^w \equiv \top/\bot$ implies that $\varphi_a^v \equiv \top/\bot$. Furthermore, $a \mapsto \mathbf{t}/\mathbf{f} \in \Gamma^{i+1}_{D'}(w_{\mathbf{u}})$ says that there exists a set of parents of a, namely P, where $P \subseteq w^{\mathbf{t}} \cup w^{\mathbf{f}}$, such that, $\varphi_a^{w|_P} \equiv \top/\bot$. Thus, $\varphi_a^{v|_P} \equiv \top/\bot$. By induction hypothesis, each $p \in P$ is strongly justified in v. Thus, a is strongly justified in v. □

We use the auxiliary Theorem 4 to present the main result of this section, i.e., to show that $StrJust$ is coNP-complete.

Theorem 5. *Let D be an ADF, let a be an argument, and let v be an interpretation of D. Deciding whether a is strongly justified in v, i.e., whether $StrJust(a,x,v,D)$, is* coNP-*complete.*

Proof sketch. First we show that $StrJust(a,x,v,D) \in$ coNP. It is shown in [19, Proposition 4.1.3] that $Cred_{grd}(a,x,D) \in$ coNP. Furthermore, the translation of a given ADF D to D' via Definition 7 can be done in polynomial time. By Theorem 4, it holds that

$Cred_{grd}(a,x,D) = yes$ iff $StrJust(a,x,v,D) = yes$. Thus, deciding whether a given argument is strongly justified in interpretation v is a coNP-problem.

Next we show that $StrJust(a,x,v,D)$ is coNP-hard. Let ψ be any Boolean formula and let $X = atom(\psi)$ be the set of atoms in ψ. Let a be a new variable. Construct $D = (\{X \cup \{a\}\}, L, C)$, s.t. $\varphi_x : x$ for each $x \in X$ and $\varphi_a : \psi$. ADF D can be constructed in polynomial time w.r.t. the size of ψ. One can check that a is strongly acceptable in any v where $v(a) = \mathbf{t}$ iff ψ is a valid formula. An illustration of the reduction for the formula $\psi = \neg b \vee b$ to the ADF $D = (\{a,b\}, L, \varphi_a : \psi, \varphi_b : b)$ is depicted in Figure 2.

For credulous denial of a, it is enough to present the acceptance condition of a equal to the negation of ψ in D, i.e., $\varphi_a : \neg\psi$, and follow a similar method. That is, a is strongly deniable in v, where $v(a) = \mathbf{f}$, iff ψ is a valid formula. □

3.4. Smallest Witness of Strong Justification

Assume that an argument a, its truth value x, and a natural number k are given. We are eager to know whether there exists a strongly admissible interpretation v such that $v(a) = x$ and $|v^{\mathbf{t}} \cup v^{\mathbf{f}}| < k$. This reasoning task is denoted by $k\text{-}Witness_{sadm}(a,x,D)$. We show that $k\text{-}Witness_{sadm}$ is Σ_2^{P}-complete. Lemma 3 shows that this problem is a Σ_2^{P}-problem and Lemma 4 indicates the hardness of this reasoning task.

Lemma 3. *Let D be an ADF, let a be an argument, let $x \in \{\mathbf{t},\mathbf{f}\}$, and let k be a natural number. Deciding whether there exists an interpretation v such that $v \in sadm(D)$, $v(a) = x$, and $|v^{\mathbf{t}} \cup v^{\mathbf{f}}| < k$ is a Σ_2^{P}-problem, i.e., $k\text{-}Witness_{sadm} \in \Sigma_2^{\mathsf{P}}$.*

Proof. For membership in Σ_2^{P}, non-deterministically guess an interpretation v and verify whether this interpretation satisfies the following items: 1. $v \in sadm(D)$; 2. $v(a) = x$; 3. $|v^{\mathbf{t}} \cup v^{\mathbf{f}}| < k$. If v satisfies all the items, then the answer to the decision problem is *yes*, i.e., $k\text{-}Witness_{sadm}(a,x,D) = yes$. Notice that we have shown in Section 3.2 that testing (1) is coNP-complete and testing (2) and (3) can clearly be done in polynomial time. That is, the algorithm first non-deterministically guesses an interpretation v and then performs checks that are in coNP to verify that v satisfies the requirements of the decision problem. Thus, this gives an $\mathsf{NP}^{\mathsf{coNP}} = \Sigma_2^{\mathsf{P}}$ procedure. □

Lemma 4. *Let D be an ADF, let a be an argument, let $x \in \{\mathbf{t},\mathbf{f}\}$, and let k be a natural number. Deciding whether there exists a strongly admissible interpretation v of D where $v(a) = x$ and $|v^{\mathbf{t}} \cup v^{\mathbf{f}}| < k$ is Σ_2^{P}-hard, i.e., $k\text{-}Witness_{sadm}$ is Σ_2^{P}-hard.*

Proof sketch. Consider the following well-known problem on quantified Boolean formulas. Given a formula $\Theta = \exists Y \forall Z\, \theta(Y,Z)$ with atoms $X = Y \cup Z$ (and $Y \cap Z = \emptyset$) and propositional formula θ. Deciding whether Θ is valid is Σ_2^{P}-complete (see e.g. [20]). We can assume that θ is of the form $\psi \wedge \bigwedge_{y \in Y}(y \vee \neg y)$, where ψ is an arbitrary propositional formula over atoms X, and that θ is satisfiable. Moreover, we can assume that the formula θ only uses $\wedge$, $\vee$, $\neg$ operations and that negations only appear in literals. Let $\bar{Y} = \{\bar{y} : y \in Y\}$, i.e., for each $y \in Y$ we introduce a new argument $\bar{y}$.

We construct an ADF $D_\Theta = (A,L,C)$ with $A = Y \cup \bar{Y} \cup Z \cup \{\theta\}$ and $C = \{\varphi_y : \top \mid y \in Y\} \cup \{\varphi_{\bar{y}} : \top \mid y \in Y\} \cup \{\varphi_z : \neg z \mid z \in Z\} \cup \{\varphi_\theta : \theta[\neg y/\bar{y}]\}$.

It is easy to verify that $g \in grd(D_\Theta)$ sets all arguments $Y \cup \bar{Y}$ to $\mathbf{t}$ and all arguments Z to $\mathbf{u}$. Moreover, $g(\theta) \in \{\mathbf{t},\mathbf{u}\}$. An illustration of the reduction for the formula $\theta = ((y_1 \wedge$

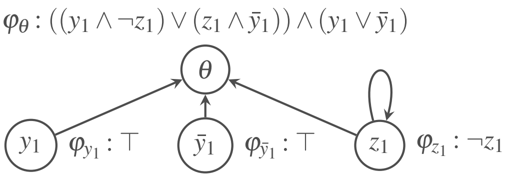

$\varphi_\theta : ((y_1 \wedge \neg z_1) \vee (z_1 \wedge \bar{y}_1)) \wedge (y_1 \vee \bar{y}_1)$
θ
y_1
$\varphi_{y_1} : \top$
$\bar{y}_1$
$\varphi_{\bar{y}_1} : \top$
z_1
$\varphi_{z_1} : \neg z_1$

Figure 3. Illustration of the reduction from the proof of Lemma 4 for $\Theta = \exists y_1 \forall z_1 ((y_1 \wedge \neg z_1) \vee (z_1 \wedge \neg y_1)) \wedge (y_1 \vee \neg y_1)$.

$\neg z_1) \vee (z_1 \wedge \neg y_1)) \wedge (y_1 \vee \neg y_1)$ to the ADF $D = (A, L, C)$ is shown in Figure 3, where: $A = \{y_1, \bar{y}_1, z_1, \theta\}$, $\varphi_{y_1} : \top, \varphi_{\bar{y}_1} : \top, \varphi_{z_1} : \neg z$ and $\varphi_\theta : ((y_1 \wedge \neg z_1) \vee (z_1 \wedge \bar{y}_1)) \wedge (y_1 \vee \bar{y}_1)$. One can check that there is an interpretation v with $v \in sadm(D_\Theta)$, $v(\theta) = \mathbf{t}$, and $|S| = |Y| + 1$ where $S = v^{\mathbf{t}} \cup v^{\mathbf{f}}$ iff Θ is a valid formula. □

Theorem 6 is a direct result of Lemmas 3 and 4.

Theorem 6. *k-Witness$_{sadm}$ is Σ_2^{P}-complete.*

4. Conclusion

We studied the computational properties of the strong admissibility semantics of ADFs. When compared to AFs, computational complexity for ADFs typically increases by one step in the polynomial hierarchy for the non-trivial reasoning tasks [21,18]. We have shown that, similarly, ADFs have higher computational complexity under the strong admissibility semantics when compared to AFs.

We next highlight an interesting difference in the complexity landscapes of AFs and ADFs. When relating the complexity of grounded and strong admissibility semantics, we have that for AFs the verification problems can be (log-space) reduced to each other, while for ADFs there is a gap between the coNP-complete Ver_{sadm} problem and the DP-complete Ver_{grd} problem. That is, on the ADF level the step of proving arguments to be **u** in the grounded interpretation adds an NP part to the complexity; a similar effect can be observed for admissible and complete semantics.

Our complexity analysis for ADFs paves the way to investigate the complexity of strong admissibility for generalizations of Dung AFs that form subclasses of ADFs, e.g., different types of bipolar ADFs [3].

Acknowledgments This research has been supported by the Center of Data Science & Systems Complexity (DSSC) Doctoral Programme, at the University of Groningen. The first researcher is currently embedded in Centrum Wiskunde & Informatica (CWI) and Supported by the Netherlands eScience Center project “The Eye of the Beholder” (project nr. 027.020.G15).

References

[1] P. M. Dung, “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games,” *Artificial Intelligence*, vol. 77, pp. 321–357, 1995.

[2] G. Brewka, S. Polberg, and S. Woltran, "Generalizations of Dung frameworks and their role in formal argumentation," *IEEE Intelligent Systems*, vol. 29, no. 1, pp. 30–38, 2014.

[3] G. Brewka and S. Woltran, "Abstract dialectical frameworks," in *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*, pp. 102–111, 2010.

[4] G. Brewka, H. Strass, S. Ellmauthaler, J. P. Wallner, and S. Woltran, "Abstract dialectical frameworks revisited," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pp. 803–809, 2013.

[5] G. Brewka, S. Ellmauthaler, H. Strass, J. P. Wallner, and S. Woltran, "Abstract dialectical frameworks: An overview," in *Handbook of Formal Argumentation* (P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, eds.), pp. 237–285, College Publications, London, February 2018.

[6] P. Baroni, M. Caminada, and M. Giacomin, "An introduction to argumentation semantics," *Knowledge Engineering Review*, vol. 26, no. 4, pp. 365–410, 2011.

[7] M. Caminada and S. Uebis, "An implementation of argument-based discussion using ASPIC-," in *COMMA*, vol. 326 of *Frontiers in Artificial Intelligence and Applications*, pp. 455–456, IOS Press, Amsterdam, 2020.

[8] R. Booth, M. Caminada, and B. Marshall, "DISCO: A web-based implementation of discussion games for grounded and preferred semantics," in *Proceedings of Computational Models of Argument COMMA* (S. Modgil, K. Budzynska, and J. Lawrence, eds.), pp. 453–454, IOS Press, Amsterdam, 2018.

[9] P. Baroni and M. Giacomin, "On principle-based evaluation of extension-based argumentation semantics," *Artificial Intelligence*, vol. 171, no. 10-15, pp. 675–700, 2007.

[10] M. Caminada, "Strong admissibility revisited," in *Proceedings of Computational Models of Argument COMMA*, vol. 266 of *Frontiers in Artificial Intelligence and Applications*, pp. 197–208, IOS Press, Amsterdam, 2014.

[11] M. Caminada and P. E. Dunne, "Strong admissibility revisited: Theory and applications," *Argument & Computation*, vol. 10, no. 3, pp. 277–300, 2019.

[12] M. Caminada, "Argumentation semantics as formal discussion," in *Handbook of Formal Argumentation* (P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, eds.), pp. 487–518, College Publications, London, 2018.

[13] M. Caminada and P. E. Dunne, "Minimal strong admissibility: A complexity analysis," in *Proceedings of Computational Models of Argument COMMA*, vol. 326 of *Frontiers in Artificial Intelligence and Applications*, pp. 135–146, IOS Press, Amsterdam, 2020.

[14] W. Dvořák and J. P. Wallner, "Computing strongly admissible sets," in *Proceedings of Computational Models of Argument COMMA 2020*, pp. 179–190, IOS Press, Amsterdam, 2020.

[15] A. Keshavarzi Zafarghandi, R. Verbrugge, and B. Verheij, "Strong admissibility for abstract dialectical frameworks," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing SAC '21*, pp. 873–880, 2021.

[16] A. Keshavarzi Zafarghandi, *Abstract Dialectical Frameworks: Semantics, Discussion Games, and Variations*. PhD thesis, University of Groningen, Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, 2022.

[17] A. Keshavarzi Zafarghandi, R. Verbrugge, and B. Verheij, "Strong admissibility for abstract dialectical frameworks," *Argument & Computation*, p. online first, 2021.

[18] W. Dvořák and P. E. Dunne, "Computational problems in formal argumentation and their complexity," in *Handbook of Formal Argumentation* (P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, eds.), pp. 631–687, College Publications, London, 2018.

[19] J. P. Wallner, *Complexity Results and Algorithms for Argumentation: Dung's Frameworks and Beyond*. PhD thesis, Vienna University of Technology, Institute of Information Systems, 2014.

[20] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, Cambridge, 2009.

[21] H. Strass and J. P. Wallner, "Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory," *Artificial Intelligence*, vol. 226, pp. 34–74, 2015.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220154

Just a Matter of Perspective

Intertranslating Expressive Argumentation Formalisms

Matthias KÖNIG [a], Anna RAPBERGER [a], and Markus ULBRICHT [b]
[a] *TU Wien, Institute of Logic and Computation*
[b] *Leipzig University, Department of Computer Science*

Abstract. Many structured argumentation approaches proceed by constructing a Dung-style argumentation framework (AF) corresponding to a given knowledge base. While a main strength of AFs is their simplicity, instantiating a knowledge base oftentimes requires exponentially many arguments or additional functions in order to establish the connection. In this paper we make use of more expressive argumentation formalisms. We provide several novel translations by utilizing claim-augmented AFs (CAFs) and AFs with collective attacks (SETAFs). We use these frameworks to translate assumption-based argumentation (ABA) frameworks as well as logic programs (LPs) into the realm of graph-based argumentation.

Keywords. Argumentation, Translations, ABA, CAF, Logic Programs, SETAF

1. Introduction

Argumentation structures often arise from instantiating knowledge bases and identifying their relevant conflicts. The representation of knowledge bases in terms of graph-based argumentation formalisms has several advantages. First, they provide an intuitive and user-friendly way for conflict-representation due to their graphical design. Second, the uniform representation allows to compare different, seemingly unrelated knowledge bases and helps to identify their similarities. Various kinds of knowledge bases and applications lead to the invention of several tailor-made argumentation formalisms, each with their own advantages and disadvantages. In formal argumentation, *Abstract Argumentation* due to Dung [1] serves as a common denominator for many of these formalisms. Popular extensions of Dung's original framework incorporate for example propositional acceptance conditions [2], assumptions [3], claims [4], or collective attacks [5]. At first glance, these formalisms seem incompatible due to their focus on seemingly entirely different features. In an effort to relate selected formalisms, researchers singled out pairs of formalisms and provided translations for the respective cases. For the classical Dung semantics, i.e., for complete, preferred, stable, and grounded semantics (*com*, *pref*, *stb*, *grd*), semantics-preserving translations have been successfully established in many cases.

In this work, we take a step back and compare a variety of argumentation formalisms, namely Assumption Based Argumentation (ABA) [3], Claim-Augmented Frameworks (CAF) [4], and Argumentation Frameworks with Collective Attacks (SETAF) [5]. Moreover we consider the closely related Normal Logic Programs (LP) and the restricted atomic LPs [6] (we expect readers to enjoy this work the most if they are al-

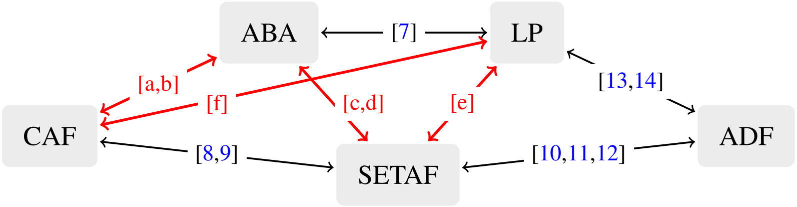


Figure 1. Overview of existing and novel transformations. Novel translations between ABA and CAFs are given in [a] Def. 3.6 and [b] 3.9; we present two translations relating ABA and SETAFs, cf. [c] Def. 3.13 and [d] 3.17; translations between [e] SETAFs and LP are in Section 4.2; and for [f] CAFs and LPs by Def. 4.3.

ready familiar with some of these formalisms). There already exist semantics-preserving translations between several classes of the aforementioned formalisms. Caminada and Schulz [7] provide a translation between ABA and LP and vice versa. In [8,9], the correspondence between well-formed CAFs and SETAFs has been settled. All of these mentioned translations preserve complete, stable, and preferred models (extensions).

If we furthermore take the well-investigated relation between Abstract Dialectical Frameworks (ADF) [2] and LPs [13,14] as well as to SETAFs, respectively [10,11,12], into account and collect all available results, we obtain the following insight: (classes of) ABA frameworks, LPs, ADFs, SETAFs, and CAFs can all be viewed, to some extent, as different sides of the same (pentagonal) coin. We summarize this insight in Figure 1. We note that not all translations consider all instances of the domain; e.g., the translation from CAFs to SETAFs restricts to so-called well-formed CAFs; also, Dvořák et. al [10] as well as Alcântara and Sá [11] focus on attacking (support-free) ADFs. Likewise, the *image* of the translation often do not cover all instances of the target formalism, e.g., Polberg [12] translates SETAFs into attacking ADFs and Caminada and Schulz [7] map LPs to a sub-class of ABA frameworks. As one can verify by following the directed arrows, there exists semantics-preserving rewriting methods between (classes of) all of these formalisms. While this existential statement suffices to establish a theoretical correspondence it is hardly of practical use for translating, e.g., ABA instances to CAFs (this concrete example would require the application of four different translations). From a theoretical point of view, one would have to comprehend several steps through various different formalisms, thereby missing the observation that there are immediate translations which preserve the structure quite well, as we will establish in this paper. For example the CAF obtained from an ABA framework is natural and can be constructed directly, and the role of the additional claims becomes clear immediately.

The paper is organized as follows. In Section 3 we focus on the intertranslatability of ABA, CAFs, and SETAFs. We show how an ABA framework naturally induces a CAF which preserves the structure of the knowledge base due to the flexible handling of claims. Moreover, we explore the advantageous features of SETAFs which yield a representation that requires fewer arguments. We will show that if one is solely interested in the underlying assumptions, SETAFs yield impressively concise representations. In Section 4 we discuss the close relation between atomic LPs, CAFs, and SETAFs, provide natural pairwise translations and demonstrate their compatibility. Along the way, we show that the instantiation procedure [15] (i.e. constructing arguments from a general LPs) can be bridged by first making the LP atomic.

We omit proofs in the present paper; full proofs are made available at https://www.dbai.tuwien.ac.at/research/report/dbai-tr-2022-123.pdf.

2. Background

We recall the necessary background for AFs since they constitute our main underlying formalism. The other formalisms will be introduced on the fly. An argumentation framework (AF) [1] is a directed graph (A,R) where A is a finite set of arguments and $R \subseteq A \times A$ the attack relation. An argument x (set $E \subseteq A$) *attacks* y if $(x,y) \in R$ (some $z \in E$ attacks y). We write $E_R^+ = \{a \in A \mid E \text{ attacks } a\}$ and $E_R^- = \{a \in A \mid (a,b) \in R, b \in E\}$, and for short $x_R^+ = \{x\}_R^+$, $x_R^- = \{x\}_R^-$; we omit subscript R if it is clear from the context.

A set $E \subseteq A$ is *conflict-free* in $F = (A,R)$ iff $(x,y) \notin R$ for all $x,y \in E$; E *defends* an argument x if E attacks each attacker of x. A conflict-free set E is *admissible* in F ($E \in adm(F)$) iff it defends all its elements. A *semantics* σ is a function which returns a set of subsets of A. These subsets are called σ-*extensions*. In this paper we consider so-called *complete*, *grounded*, *preferred*, and *stable* semantics (abbr. *com*, *grd*, *pref*, *stb*).

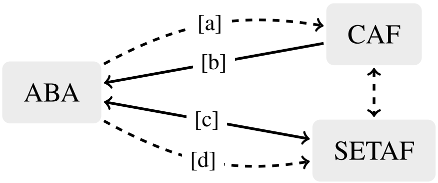
Definition 2.1. Let $F = (A,R)$ be an AF and $E \in adm(F)$. We let $E \in com(F)$ iff E contains all arguments it defends; $E \in grd(F)$ iff E is $\subseteq$-minimal in $com(F)$; $E \in pref(F)$ iff E is $\subseteq$-maximal in $com(F)$; $E \in stb(F)$ iff $E^+ = A \setminus E$.

Throughout the paper we will frequently use the notion of a hitting set: Let $\mathscr{M}$ be a set of sets. We call $\mathscr{H}$ a *hitting set* of $\mathscr{M}$ if $\mathscr{H} \cap M \neq \emptyset$ for each $M \in \mathscr{M}$. By $HS_{min}(\mathscr{M})$ we denote the $\subseteq$-minimal hitting sets of $\mathscr{M}$. We will make use of the following result.

Lemma 2.2 ([16]). *Let $X = \{X_1, \ldots, X_n\}$ be a set of sets with $X_i \not\subseteq X_j$ for $i \neq j$. Then $HS_{min}(HS_{min}(X)) = X$.*

3. Intertranslatability of ABA Frameworks, CAFs, and SETAFs

In this section, we consider the relation between ABA frameworks, well-formed CAFs, and SETAFs. Semantics for ABA can be equivalently formulated in terms of *assumptions* or in terms of *arguments* via attacks based on their *claims*. There are different representations that put the focus on either preserving assumption-sets or extensions in terms of conclusions. Figure 2 shows the different translations and directions we consider in this section: while the CAF representation focuses on extensions in terms of conclusions but also preserves assumption-extension under projection (cf. translation [a] in Figure 2), there are several possibilities to represent ABA frameworks as SETAFs. Translation [c] relates assumptions in the ABA framework with arguments in the SETAF while Translation [d] relates conclusions with arguments. We also consider the reversed direction, i.e., constructing ABA frameworks from CAFs and SETAFs (cf. [b] and [c], respectively). In Section 3.1, we consider the relation of ABA and CAFs; in Section 3.2 we examine the relation between ABA and SETAFs. First, we provide necessary background for ABA.



Translations [a,d] from ABA to CAFs and SETAFs preserve conclusions (cf. Def. 3.6 and 3.17); Translation [b] from CAFs to ABA preserves proper conclusion-extensions (cf. Def. 3.9); Translation [c] between ABA and SETAFs preserves assumption-sets (cf. Def. 3.13). The diagram commutes w.r.t. dashed lines (cf. Prop. 3.21).

Figure 2. Semantics-preserving translations between ABA frameworks, CAFs, and SETAFs.

Assumption-based Argumentation. We assume a deductive system $(\mathcal{L},\mathcal{R})$, where $\mathcal{L}$ is a formal language and $\mathcal{R}$ is a set of inference rules of the form $r : a_0 \leftarrow a_1,\dots,a_n$, $a_i \in \mathcal{L}$; $head(r) = a_0$ denotes the head and $body(r) = \{a_1,\dots,a_n\}$ the body of rule r.

Definition 3.1. An ABA framework is a tuple $(\mathcal{L},\mathcal{R},\mathcal{A},\overline{\ \ })$, where $(\mathcal{L},\mathcal{R})$ is a deductive system, $\mathcal{A} \subseteq \mathcal{L}$, $\mathcal{A} \neq \emptyset$ a set of assumptions, and a contrary function $\overline{\ \ } : \mathcal{A} \to \mathcal{L}$.

We focus on ABA frameworks which are *flat*, i.e., for each rule $r \in \mathcal{R}$, $head(r) \notin \mathcal{A}$, and *finite*, i.e., $\mathcal{L}$, $\mathcal{R}$, $\mathcal{A}$ are finite. Furthermore, we assume $\mathcal{L}$ to be a set of atoms.

An atom $p \in \mathcal{L}$ in an ABA framework $D = (\mathcal{L},\mathcal{R},\mathcal{A},\overline{\ \ })$ is *tree-derivable* from assumptions $S \subseteq \mathcal{A}$ and rules $R \subseteq \mathcal{R}$, denoted by $S \vdash_R p$, if there is a finite rooted labeled tree such that the root is labeled with p, the set of labels for the leaves is equal to S or $S \cup \{\top\}$, and there is a surjective mapping from the set of internal nodes to R s.t. each internal note v is labeled with $head(r)$ for some $r \in R$ and the set of all successor nodes corresponds to $body(r)$ or $\top$ if $body(r) = \emptyset$. We write $S \vdash p$ if there exists $R \subseteq \mathcal{R}$ with $S \vdash_R p$. Derivability for a set of assumptions $S \subseteq \mathcal{A}$ is defined via $Th_D(S) = \{p \mid S \vdash p\}$.

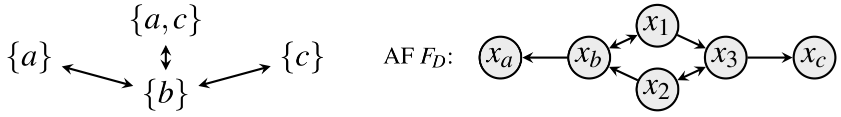
A set $S \subseteq \mathcal{A}$ *attacks* $a \in \mathcal{A}$ if there is $S' \subseteq S$ such that $S' \vdash \overline{a}$; S attacks $T \subseteq \mathcal{A}$ if it attacks some $a \in T$. S is conflict-free if it does not attack itself; S is admissible if it is conflict-free and counter-attacks each attacker (we say: S defends itself). We recall grounded, complete, preferred, and stable ABA semantics (abbr. *grd*, *com*, *pref*, *stb*).

Definition 3.2. For an ABA $D = (\mathcal{L},\mathcal{R},\mathcal{A},\overline{\ \ })$ and an admissible set $S \subseteq \mathcal{A}$, $S \in com(D)$ iff S contains every assumption it defends; $S \in grd(D)$ iff S is $\subseteq$-minimal in $com(D)$; $S \in pref(D)$ iff S is $\subseteq$-maximal in $com(D)$; $S \in stb(D)$ iff S attacks each $\{x\} \subseteq \mathcal{A} \setminus S$. Given $\sigma \in \{com, grd, pref, stb\}$, the *$\sigma$-conclusion-extensions* of D are $\sigma_{Th}(D) = \{Th_D(S) \mid S \in \sigma(D)\}$, the *proper* σ-conclusion-extensions of D are given by $\{C \setminus \mathcal{A} \mid C \in \sigma_{Th}(D)\}$.

ABA frameworks and AFs are closely related (see, e.g., [17]). Viewing tree derivations as arguments, an ABA framework induces a corresponding AF as follows.

Definition 3.3. The associated AF $F_D = (A,R)$ of an ABA $D = (\mathcal{L},\mathcal{R},\mathcal{A},\overline{\ \ })$ is given by $A = \{S \vdash p \mid \exists R \subseteq \mathcal{R} : S \vdash_R p\}$ and attack relation $(S_1 \vdash p, S_2 \vdash q) \in R$ iff $p \in \{\overline{s} \mid s \in S_2\}$.

Example 3.4. Consider the ABA D with assumptions $\mathcal{A} = \{a,b,c\}$ and rules $r_1 : p \leftarrow a$, $r_2 : p \leftarrow c$, and $r_3 : q \leftarrow b$. Moreover, $\overline{a} = b$, $\overline{b} = p$, and $\overline{c} = q$. Below we depict the attacks between the assumption-sets (left, we omit $\emptyset$, $\{a,b\}$, $\{b,c\}$, and $\mathcal{A}$) and the AF F_D (right) with arguments x_i (induced by rules r_i) and arguments x_a, x_b, x_c for the assumptions.



The ABA D has two stable assumption-sets: $S_1 = \{b\}$ and $S_2 = \{a,c\}$ with $Th_D(S_1) = \{b,q\}$ and $Th_D(S_2) = \{a,c,p\}$. The stable extensions in F_D are $\{x_3,x_b\}$ and $\{x_1,x_2,x_a,x_c\}$.

For an argument $x = S \vdash p$, we consider functions $cl(x) = p$ and $asms(x) = S$; moreover, $cl(E) = \{cl(x) \mid x \in E\}$ and $asms(E) = \bigcup_{x \in E} asms(x)$ for a set of arguments E.

Proposition 3.5 ([17]). *For an ABA D, its associated AF F, $\sigma \in \{grd, com, pref, stb\}$; if $E \in \sigma(F)$ then $asms(E) \in \sigma(D)$; and if $S \in \sigma(D)$ then $\{S' \vdash p \mid \exists S' \subseteq S : S' \vdash p\} \in \sigma(F)$.*

3.1. Assumption-based Argumentation and Claims

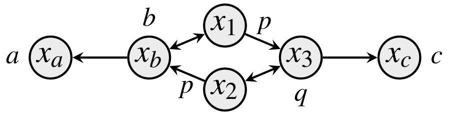
Claim-augmented Argumentation Frameworks. A *claim-augmented argumentation framework (CAF)* [4] is a triple $\mathcal{F} = (A, R, cl)$ where $F = (A, R)$ is an AF and a function cl which assigns a claim to each argument in A. The claim-function is extended to sets in the natural way, i.e. for a set $E \subseteq A$, we let $cl(E) = \{cl(a) \mid a \in E\}$. For a CAF $\mathcal{F} = (A, R, cl)$, $F = (A, R)$, and an AF semantics σ, we define $\sigma_c(\mathcal{F}) = \{cl(E) \mid E \in \sigma(F)\}$. In this work, we focus on CAFs that are *well-formed*; i.e. CAFs satisfying $a_R^+ = b_R^+$ for all $a, b \in A$ with $cl(a) = cl(b)$. Whenever we write CAF, we mean well-formed CAF.

ABA-CAF Translations. There is a natural adaption of the AF instantiation given in Definition 3.3 to CAFs by assigning each argument $S \vdash p$ its claim p:

Definition 3.6. The associated CAF $\mathcal{F}_D = (A, R, cl)$ for an ABA $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ is obtained by constructing (A, R) from Definition 3.3 and $cl(S \vdash p) = p$ for all $S \vdash p \in A$.

Example 3.7. Instantiating ABA D from Example 3.4 yields the following CAF:

CAF $\mathcal{F}_D$:



The CAF $\mathcal{F}_D$ is well-formed since attacks depend on the conclusion of the attacking argument: an argument x attacks argument y if $cl(x) = \overline{a}$ for some $a \in asms(y)$. Due to Proposition 3.5, the translation preserves the σ-conclusion-extensions of an ABA D; assumption-extensions can be obtained by restricting the conclusion-sets to $\mathcal{A}$.

Proposition 3.8. *For an ABA $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$, its associated CAF $\mathcal{F}_D$ and $\sigma \in \{grd, com, pref, stb\}$, it holds that $\sigma_{Th}(D) = \sigma_c(\mathcal{F}_D)$ and $\sigma(D) = \{C \cap \mathcal{A} \mid C \in \sigma_c(\mathcal{F}_D)\}$.*

For the other direction, we identify each claim c in a given well-formed CAF as contrary of some hidden assumption a_c; moreover, each argument which is attacked by claim c is derived from assumption a_c (i.e., a_c is attacked by all arguments with claim c).

Definition 3.9. The associated ABA $D_{\mathcal{F}} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ of a CAF $\mathcal{F} = (A, R, cl)$ is given by $\mathcal{A} = \{a_c \mid c \in cl(A)\}$, $\mathcal{L} = \mathcal{A} \cup cl(A)$, contrary function $\overline{a_c} = c$ for all $c \in cl(A)$, and $\mathcal{R} = \{cl(x) \leftarrow \{a_{cl(y)} \mid y \in x^-\} \mid x \in A\}$.

We obtain a translation which relates claim-sets of the CAF with the *proper* conclusion-extensions of the obtained ABA. Note the restriction to the proper conclusion-extensions is necessary since the translation treats assumptions as implicit information.

Proposition 3.10. *For a CAF $\mathcal{F} = (A, R, cl)$, its corresponding ABA $D_{\mathcal{F}}$ and a semantics $\sigma \in \{grd, com, pref, stb\}$, it holds that $\sigma_c(\mathcal{F}) = \{C \setminus \mathcal{A} \mid C \in \sigma_{Th}(D_{\mathcal{F}})\}$.*

Example 3.11. Consider the CAF $\mathcal{F}_D$ from Example 3.7. We construct an ABA $D_{\mathcal{F}_D} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ with $\mathcal{A} = \{a_p, a_q, a_a, a_b, a_c\}$, contrary function $\overline{a_x} = x$ for each claim in $\mathcal{F}_D$ and rules $p \leftarrow a_b$, $p \leftarrow a_q$, $q \leftarrow a_p$, $a \leftarrow a_b$, $b \leftarrow a_p$, and $c \leftarrow a_q$. The ABA $D_{\mathcal{F}_D}$ has two stable assumption-sets $S_1 = \{a_a, a_p, a_c\}$ and $S_2 = \{a_b, a_q\}$ with $Th_{D_{\mathcal{F}_D}}(S_1) = \{a_a, a_p, a_c, b, q\}$ and $Th_{D_{\mathcal{F}_D}}(S_2) = \{a_b, a_q, a, c, p\}$. The proper conclusion-extensions of $D_{\mathcal{F}_D}$ are $\{b, q\}$ and $\{a, c, p\}$ which correspond to the conclusion-extensions of D.

3.2. Assumption-based Argumentation and Collective Attacks

Argumentation frameworks with collective attacks. A SETAF [18] is a pair $SF = (A,R)$ where A is a finite set of arguments and $R \subseteq (2^A \setminus \{\emptyset\}) \times A$ is the attack relation. For an attack $(T,h) \in R$ we call T the *tail* and h the *head* of the attack. SETAFs (A,R) where $|T| = 1$ for all $(T,h) \in R$ amount to AFs. In that case, we write (t,h) to denote $(\{t\},h)$.

A set $T_1 \subseteq A$ attacks $h \in A$ (the set $T_2 \subseteq A$) if there is $T_1' \subseteq T_1$ (and $h \in T_2$, resp.) such that $(T_1',h) \in R$. We write $h_R^- = \{T \mid (T,h) \in R\}$ to denote the set of attackers of the argument h (in R). For $S \subseteq A$, we use S_R^+ to denote the set of arguments attacked by S (in R). S is *conflict-free* in SF if it does not attack itself; S defends argument $a \in A$ if it attacks each attacker of a; likewise, S defends $T \subseteq A$ iff it defends each $a \in T$. A set S is called *admissible* if it defends itself ($adm(SF)$ denotes the set of all admissible sets in SF). AF semantics generalize to SETAFs in the following way [19,5].

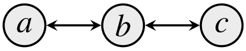
Definition 3.12. Given a SETAF $SF = (A,R)$ and a set $S \in adm(SF)$. Then, $S \in com(SF)$ iff S contains each argument it defends; $S \in grd(SF)$ iff S is $\subseteq$-minimal in $com(SF)$; $S \in pref(SF)$ iff S is $\subseteq$-maximal in $com(SF)$; $S \in stb(SF)$ iff S attacks all $a \in A \setminus S$.

ABA-SETAF-translations: relating assumptions with arguments. When inspecting the definitions of attacks for ABA frameworks and SETAFs we find the following natural correspondence: a set of arguments T attacks an argument h in the SETAF iff T derives the contrary of h in the corresponding ABA. We obtain an ABA framework from a given SETAF by introducing a rule $\overline{h} \leftarrow T$ for each attack $(T,h) \in R$. For the other direction, we identify conflicts between assumption-sets. Below, we give the resulting translations.

Definition 3.13. For an ABA $D = (\mathscr{L},\mathscr{R},\mathscr{A},\overline{})$, we define the corresponding SETAF $SF_D = (A_D,R_D)$ with $A_D = \mathscr{A}$ and $(S,a) \in R_D$ iff $S \vdash \overline{a}$. For a SETAF $SF = (A,R)$, we define the corresponding ABA $D_{SF} = (\mathscr{L}_{SF},\mathscr{R}_{SF},\mathscr{A}_{SF},\overline{})$ with $\mathscr{L}_{SF} = A \cup \{p_x \mid x \in A\}$, $\mathscr{A}_{SF} = A$, $\overline{x} = p_x$ for all $x \in A$, and for each $(T,h) \in R$, we add a rule $p_h \leftarrow T$ to $\mathscr{R}_{SF}$.

Example 3.14. Instantiating ABA D from Example 3.4 yields the following SETAF:

SETAF SF_D:



The translations indeed preserve the (assumption-based) semantics.

Proposition 3.15. *Given a semantics $\sigma \in \{grd, com, pref, stb\}$. For an ABA D and its associated SETAF SF_D, it holds that $\sigma(D) = \sigma(SF_D)$. For a SETAF SF and its associated ABA D_{SF}, it holds that $\sigma(SF) = \sigma(D_{SF})$.*

We obtain the following strong intertranslatibility result using the correspondence $(S,a) \in R$ in SF iff $\overline{a} \leftarrow S$ in D_{SF} iff $S \vdash \overline{a}$ in D_{SF} iff $(S,a) \in R$ in $SF_{D_{SF}} = SF$.

Proposition 3.16. *Given a SETAF SF, it holds that $SF_{D_{SF}} = SF$.*

This result shows that no information is lost in the SETAF when representing it in terms of ABA. The other direction, i.e., translating ABA frameworks to SETAFs, however, comes with a cost: given an ABA framework D, it is impossible to extract the σ-conclusion-extensions from SETAF SF_D. This means that the conclusions of a given ABA instance are lost when applying the translation. In the following, we present a translation that preserves conclusions of an ABA instance.

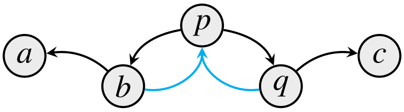
ABA-SETAF-translations: relating conclusions and arguments. In order to establish a translation from ABA frameworks to SETAFs that preserves the conclusions of the original instance, we proceed as follows: For a given ABA instance $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, \overline{})$, we construct a corresponding SETAF $SF_D = (A, R)$ with

1. $A = \{p \mid \exists S \subseteq \mathscr{A} : S \vdash p\}$, i.e., conclusions in D correspond to arguments of our resulting SETAF (observe that each assumption $a \in \mathscr{A}$ is a conclusion of D); and
2. a set of conclusions C attacks a conclusion p in SF_D, i.e., $(C, p) \in R$, iff C contains a contrary for each set of assumptions S with $S \vdash p$, and C is $\subseteq$-minimal among all such sets (i.e., C is a *minimal hitting set* of the set $\{\{\overline{a} \mid a \in S\} \mid S \vdash p\}$).

Definition 3.17. For a given ABA instance $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, \overline{})$, let $\mathscr{S}_p = \{S \mid S \vdash p\}$ and $\overline{\mathscr{S}}_p = \{\{\overline{a} \mid a \in S\} \mid S \vdash p\}$ for each $p \in \mathscr{L}$. We construct the SETAF $SF_D^c = (A, R)$ with $A = \{p \mid \exists S \subseteq \mathscr{A} : S \vdash p\}$ and $R = \{(C, p) \mid p \in A, C \in HS_{min}(\overline{\mathscr{S}}_p)\}$.

Example 3.18. We construct SETAF SF_D^c from the ABA D from Example 3.4. The arguments in SF_D^c correspond to the conclusions in D, i.e., $A = \{a, b, c, p, q\}$. We determine the attackers of $p \in A$: first, we identify the set $\mathscr{S}_p = \{\{a\}, \{c\}\}$ that contains all assumption-sets that derive p (in D); the set $\overline{\mathscr{S}}_p = \{\{b\}, \{q\}\}$ contains the respective contraries. The unique hitting set of $\overline{\mathscr{S}}_p$ is $\{b, q\}$, thus $\{b, q\}$ attacks p. We depict the resulting SETAF below (the joint arcs from $\{b, q\}$ to p (in blue) represent the set-attack):

SETAF SF_D^c:



The construction indeed preserves the σ-conclusion-extensions for the considered semantics; moreover, we obtain the assumption-extensions of the original instance by projecting the conclusion-extensions to the assumptions $\mathscr{A}$.

Proposition 3.19. *For an ABA $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, \overline{})$, its associated SETAF SF_D^c and $\sigma \in \{grd, com, pref, stb\}$, it holds that $\sigma_{Th}(D) = \sigma(SF_D^c)$ and $\sigma(D) = \{C \cap \mathscr{A} \mid C \in \sigma(SF_D^c)\}$.*

3.3. Summary & Compatibility

We presented several different translations from ABA to CAFs and SETAFs and vice versa. For CAFs, we related claims with conclusions; for SETAFs, we considered two translations by relating arguments with assumptions and with conclusions, respectively.

When comparing the ABA instances when starting from a CAF or a SETAF (cf. Definition 3.9 and 3.13, respectively), we observe the following similarities: in both cases, the resulting ABA is flat, also, each rule contains only assumptions in its body, furthermore, no contrary of an assumption is an assumption. We furthermore observe the following notable difference between the two translations: while the translation from ABA to CAF potentially causes an exponential blow-up as the argument-construction can be exponential in the number of assumptions, we observe that the resulting SETAF is linear in the number of assumptions, i.e., the exponential blow-up can be avoided. We note, however, that the computation of the SETAF might be exponential—the computational effort is shifted to the construction of the attack relation which requires to identify tree-derivations $S \vdash \overline{a}$ in the ABA framework to define attacks (S, a) in the SETAF.

We end this section by presenting a strong intertranslatability result for our considered formalisms. For this, we make use of the translation from well-formed CAFs to SETAFs [8]. To fit our setting, we reformulate the translation in terms of hitting sets instead of CNF and DNF-formulas to capture the attack-structure of the frameworks.

Definition 3.20 (cf. [8])**.** For a well-formed CAF $\mathscr{F} = (A, R, cl)$, we define the corresponding SETAF $SF_{\mathscr{F}} = (A_{\mathscr{F}}, R_{\mathscr{F}})$ by letting $A_{\mathscr{F}} = cl(A)$ and $R_{\mathscr{F}} = \{(T, c) \mid c \in cl(A), T \in HS_{min}(\{cl(x_R^-) \mid x \in A, cl(x) = c\})\}$. For a SETAF $SF = (A, R)$, we define the corresponding CAF $\mathscr{F}_{SF} = (A_{SF}, R_{SF}, cl_{SF})$ with $A_{SF} = \{x_{c,h} \mid c \in A,\ h \in HS_{min}(c_R^-)\}$, $cl_{SF}(x_{c,h}) = c$, and $R_{SF} = \{(x_{c,h_x}, y_{d,h_y}) \mid c \in h_y\}$.

Restricting the translation to *redundancy-free* CAFs, i.e., frameworks s.t. there are no $x, y \in A$ with $cl(x) = cl(y)$, $x^+ = y^+$, and $x^- \subseteq y^-$, we obtain the following result.

Proposition 3.21. *Given an ABA $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, {}^-)$, its corresponding SETAF SF_D^c (cf. Definition 3.17), let $\mathscr{F}_D$ be the corresponding CAF (cf. Definition 3.6), and let $SF_{\mathscr{F}_D}^c$ be the SETAF corresponding to the CAF $\mathscr{F}_D$ (cf. Definition 3.20). It holds that $SF_D^c = SF_{\mathscr{F}_D}^c$.*

4. Strong Intertranslatability of LPs, CAFs, and SETAFs

In this section we strengthen the results regarding CAFs, LPs, and SETAFs by providing structure-preserving translations for suitable normal forms of the formalisms. This highlights their equivalent expressiveness. While there is an immediate correspondence between CAFs and LPs, the connection to SETAFs is via a detour making use of hitting sets, as we will explain in more detail in Section 4.1 (cf. [20]). The relations we will discuss are depicted in Figure 3. Our way to extract arguments from an LP is similar to the AF-instantiation reported in [15] where a semantics correspondence between LPs and AFs has been established. Due to space restrictions, we will focus our attention on stable semantics since this is the most commonly used semantics for LPs, but we want to emphasize that analogous results hold for the other cases, i.e. *com*, *grd*, and *pref* as well. Moreover, most results reported in this section are concerned with syntactical properties.

Logic Programs. We consider logic programs with default negation *not*. Such programs consist of rules of the form "$c \leftarrow a_1, \ldots, a_n, \text{not } b_1, \ldots, \text{not } b_m$." where $0 \leq n, m$ and the a_i, b_i and c are ordinary atoms. We let $head(r) = c$, $pos(r) = \{a_1, \ldots, a_n\}$ and $neg(r) = \{b_1, \ldots, b_m\}$. Let $\mathscr{L}(P)$ be the set of all atoms occurring in P. For $B = \{b_1, \ldots, b_m\}$, we use not B as a shorthand for the conjunction not $b_1, \ldots,$ not b_m. A rule r is *atomic* [6] if $pos(r) = \emptyset$; a program P is *atomic* if each rule in P is.

For LPs without default negation ($neg(r) = \emptyset$) the unique stable model is the smallest set of atoms closed under all rules, where a set E is closed under a rule r with $neg(r) = \emptyset$

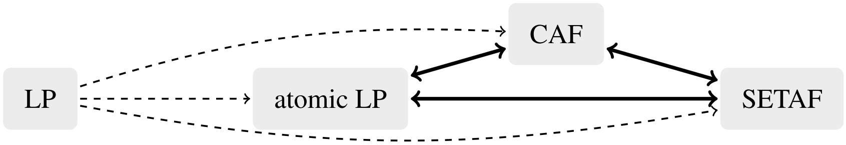


Figure 3. Transformations between formalisms discussed in Section 4

iff $pos(r) \subseteq E$ implies $head(r) \in E$. For any LP P, a set E of atoms is a *stable model* ($E \in stb(P)$) iff E is the stable model of $P^E = \{head(r) \leftarrow pos(r) \mid neg(r) \cap E = \emptyset\}$.

Example 4.1. If $P = \{(d \leftarrow \text{not}\, a,\ \text{not}\, b.), (d \leftarrow \text{not}\, c.), (a \leftarrow \text{not}\, c.), (c \leftarrow \text{not}\, a.), (b.)\}$, then P is atomic. For $E = \{b, c\}$ we have $P^E = \{(c.), (b.)\}$ and thus $E \in stb(P)$.

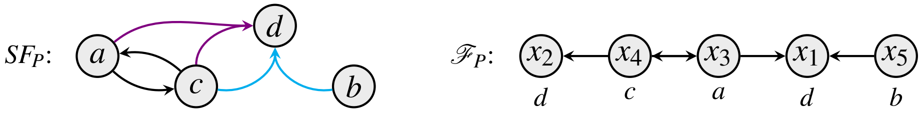
Redundancies. Throughout this section we will require redundancy notions for our formalisms. An argument $x \in A$ in a CAF $\mathscr{F} = (A, R, cl)$ is *redundant* if there is $y \in A$ with $cl(x) = cl(y)$ and $y^- \subseteq x^-$. An attack $(T, h) \in R$ in a SETAF $SF = (A, R)$ is *redundant* if there is $(T', h) \in R$ with $T' \subsetneq T$. A rule $r \in P$ of an atomic LP is *redundant* if there is $r' \in P$ with $head(r) = head(r')$ and $neg(r') \subseteq neg(r)$; an atom $a \in \mathscr{L}(P)$ is *redundant* if it does not occur as a rule head in P. A CAF resp. SETAF resp. LP without redundant arguments resp. attacks resp. rules and atoms is *redundancy-free*.

4.1. High Level Point of View

In the following subsections we will require various translations between the formalisms, which may appear rather technical at first glance. However, by closely inspecting all cases we observe that the constructed instances of the respective formalisms are quite similar in their spirit and translations are obtained by using suitably applied simple steps.

More precisely, inter-translating CAFs and atomic LPs is done by identifying rule heads with claims and bodies with in-going attacks. Recall our program P from above. In a rather immediate way, the program induces a CAF $\mathscr{F}_P$ consisting of five arguments x_i (one for each rule) where $cl(x_1) = cl(x_2) = d$, $cl(x_3) = a$, $cl(x_4) = c$, and $cl(x_5) = b$ corresponding to the rule heads. Moreover, $cl(x_1^-) = \{a, b\}$, $cl(x_2^-) = cl(x_3^-) = \{c\}$, $cl(x_4^-) = \{a\}$, and $cl(x_5^-) = \emptyset$ defines the attack relation of the well-formed CAF $\mathscr{F}_P$.

When connecting either CAFs or atomic LPs to SETAFs, the notion of a hitting set is required. In SETAFs, we do not use multiple copies of the same claim resp. rule head, but encode the acceptability condition solely in the attack relation. The corresponding SETAF would therefore possess only the four arguments a, b, c, d. For example, d *cannot* be accepted if (i) either a or b is inferred (first rule not applicable) and (ii) c is inferred (second rule not applicable either). This yields the following SETAF SF_P. Below, we also depict the CAF $\mathscr{F}_P$ we calculated earlier:



The more challenging part is dropping the assumption that the given LP P is atomic (see Figure 3). For this, we will utilize an inductive procedure constructing arguments [15].

Definition 4.2. For an LP P, A is an argument in P ($A \in Args(P)$) with $\textsc{Conc}(A) = c$, $\textsc{Rules}(A) = \bigcup_{i \leq n} \textsc{Rules}(A_i) \cup \{r\}$, and $\textsc{Vul}(x) = \bigcup_{i \leq n} \textsc{Vul}(A_i) \cup \{b_1, \ldots, b_m\}$ iff there are $A_1, \ldots, A_n \in Args(P)$ and a rule $r \in P$ with $r = c \leftarrow \textsc{Conc}(A_1), \ldots, \textsc{Conc}(A_n)$, not $b_1, \ldots,$ not b_m, and $r \notin \textsc{Rules}(A_i)$ for all $i \leq n$.

We will show that this procedure can be mimicked by rewriting P. For example let $P' = \{(d \leftarrow c,\ \text{not}\, b.), (d \leftarrow \text{not}\, c.), (a \leftarrow \text{not}\, c.), (c \leftarrow \text{not}\, a.), (b.)\}$. The atomic program P from above is the result of inserting the rule $(c \leftarrow \text{not}\, a.)$ in $(d \leftarrow c,\ \text{not}\, b.)$.

4.2. Translations

CAFs and Logic Programs. We will now formally establish the correspondence between CAFs and LPs, by making use of $Args(P)$ in case P is not atomic.

Definition 4.3. For a CAF $\mathscr{F} = (A,R,cl)$, we define the corresponding atomic LP $P_{\mathscr{F}}$ by $P = \{c \leftarrow \text{not}\, B. \mid a \in A,\ cl(a) = c,\ cl(a^-) = B\}$. For an LP P, we set $\mathscr{F}_P = (A_P, R_P, cl_P)$ where $A_P = Args(P)$, $R_P = \{(a,b) \mid cl(a) \in \textsc{Vul}(b)\}$, and $cl_P(a) = \textsc{Conc}(a)$.

Example 4.4. The LP P' from above yields four arguments stemming from the atomic rules, e.g. there is some argument A with $\textsc{Conc}(A) = c$, $\textsc{Vul}(A) = \{a\}$ and $\textsc{Rules}(A) = \{(c \leftarrow \text{not}\, a.)\}$. From $(d \leftarrow c,\ \text{not}\, b.)$ and this argument A we construct another argument with conclusion d and vulnerabilities $\{a,b\}$ (inherited from A and the applied rule). The complete corresponding CAF $\mathscr{F}_{P'}$ is the same as the CAF $\mathscr{F}_P$ depicted in Section 4.1.

A rather convenient feature of this approach is that we can infer the semantics correspondence from [15] due to the way CAF semantics make use of the claims.

Proposition 4.5. *For $\mathscr{F}$ a CAF and P an LP, $stb(\mathscr{F}) = stb(P_{\mathscr{F}})$ and $stb(P) = stb(\mathscr{F}_P)$.*

By inspecting Definition 4.2 we observe that the challenging part is handling positive atoms in rule bodies. If P is atomic, we can extract the corresponding CAF $\mathscr{F}_P = (A_P, R_P, cl_P)$ immediately via $A_P = P$, $R_P = \{(a,b) \mid head(a) \in neg(b)\}$, and $cl_P(a) = head(a)$. The fact that atomic LPs and CAFs are so closely related motivates the question whether we can transform the LP before constructing the arguments as done in [15]. A technique of this kind could pre-process the LP instead of utilizing the instantiation procedure. In the following, we formalize this idea.

Definition 4.6. For an LP P the corresponding atomic LP P_{AT} is defined inductively:

- If $r \in P$ is atomic, then $r \in P_{AT}$.
- If there is a rule $r_0 \in P$ with $pos(r_0) = \{a_1, \dots, a_n\}$ and for each a_i, $1 \leq i \leq n$, there is some rule $r_i \in P_{AT}$ s.t. $head(r_i) = a_i$, then there is a rule $r \in P_{AT}$ with $head(r) = head(r_0)$, $pos(r) = \emptyset$, and $neg(r) = \bigcup_{i=1}^{n} neg(r_i)$.

Example 4.7. Applied to our LP $P' = \{(d \leftarrow c,\ \text{not}\, b.), (d \leftarrow \text{not}\, c.), (a \leftarrow \text{not}\, c.), (c \leftarrow \text{not}\, a.), (b.)\}$ this procedure yields $P'_{AT} = P$ with P as in Example 4.1.

The following theorem formalizes that this pre-processing step successfully mimics the inductive procedure from [15]. Informally speaking, instantiating the LP is done as in Definition 4.2 and yields the same result as turning the LP into an atomic one via iterative insertion of atomic rules and then extracting the corresponding CAF by identifying rule heads with claims and rule bodies with in-going attacks. Formally:

Theorem 4.8. *Let P be an LP. Then $\mathscr{F}_P = \mathscr{F}_{P_{AT}}$.*

SETAFs and LPs We also want to briefly mention that analogous results hold when turning an LP into a SETAF, which can be done as follows. For an LP P we define by $A_P = \bigcup_{A \in Args(P)} \textsc{Conc}(a)$ and $R_P = \{(T,c) \mid T \in HS_{min}(\{\textsc{Vul}(A) \mid A \in Args(P),\ \textsc{Conc}(A) = c\})\}$ the associated SETAF SF_P. For a SETAF $SF = (A,R)$, we define its associated LP $P_{SF} = \{c \leftarrow \text{not}\, B. \mid B \in HS_{min}(c_R^-)\}$. As observed before, the construction of $Args(P)$ can be omitted if P is atomic. With these constructions, we find:

Theorem 4.9. *For a SETAF SF and an LP P, $stb(SF) = stb(P_{SF})$ and $stb(P) = stb(SF_P)$. Moreover, $SF_P = SF_{P_{AT}}$.*

4.3. Summary & Compatibility

In this section, we presented translations from LPs to SETAFs and to CAFs, respectively. We observe that when instantiating an LP as CAF or SETAF, an exponential blow-up cannot be avoided due to the construction of arguments which is an inherent part of both procedures. For atomic LPs, on the other hand, the number of arguments is linear in the number of rules in both formalisms. For the other direction, i.e., when translating a CAF or SETAF into an LP, the resulting LP is atomic. It can be shown that for atomic LPs, these constructions are bijective and each others inverse, establishing a close relation.

Lemma 4.10. *For all redundancy-free atomic LPs P, CAFs $\mathscr{F}$, and SETAFs SF, respectively, it holds that i) $SF_{P_{SF}} = SF$; ii) $P_{SF_P} = P = P_{\mathscr{F}_P}$; and iii) $\mathscr{F}_{P_{\mathscr{F}}} = \mathscr{F}$.*

We end this section with a strong intertranslatability result in the spirit of Theorem 3.21, stating that all (atomic, well-formed, and redundancy-free) instances of the considered formalisms can be equivalently represented as CAFs, LPs, or SETAFs without any loss of information via the presented translations and the method in [9] (cf. Definition 3.20). This shows that all of our constructions are compatible with each other and similar in their behavior. In particular, the order in which they are applied is arbitrary.

Theorem 4.11. *For all redundancy-free atomic LPs P, SETAFs SF, CAFs $\mathscr{F}$, we have $\mathscr{F}_{SF} \cong \mathscr{F}_{P_{SF}}$; $P_{SF} = P_{\mathscr{F}_{SF}}$; $\mathscr{F}_P \cong \mathscr{F}_{SF_P}$; $SF_P = SF_{\mathscr{F}_P}$; $SF_{\mathscr{F}} = SF_{P_{\mathscr{F}}}$; and $P_{\mathscr{F}} = P_{SF_{\mathscr{F}}}$.*

5. Discussion

In this paper we investigated translations between the argumentation formalisms ABA, CAF, SETAF as well as their connections to LP. We strengthened the implicitly existing intertranslatability result by providing additional translations, filling some of the existing gaps. For selected translations we showed structure-preserving properties and argued why others (such as those involving ABA) might not feature this preservation. Finally, our overview yields implications regarding expressiveness: the formalisms under our consideration admitting strong intertranslatability are equally expressive—i.e. , they can describe the same sets of models (extensions). These results illustrate the usefulness of the versatility in argumentation formalisms: while certain applications might suggest the usage of a specific formalism, it might be useful to later translate this framework and utilize features that are native to another formalism. Strong intertranslatability even guarantees the preservation of the structure, which opens interesting topics for future work: as some of the discussed translations are modular in some sense, one might even be able to instantiate the same knowledge base as part formalism A and part formalism B, while connecting both parts in later steps during the workflow. Another useful consequence of our findings is that it is now easier to transfer concepts and ideas between formalisms, serving as a starting point for various investigations that highlight the similarities of the considered approaches even further.

Acknowledgments

This research has been supported by the Vienna Science and Technology Fund (WWTF) through project ICT19-065, the Austrian Science Fund (FWF) through projects W1255-N23 and P32830, and by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig.

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artif Intell. 1995;77(2):321-58.

[2] Brewka G, Woltran S. Abstract Dialectical Frameworks. In: Proceedings of KR 2010. AAAI Press; 2010. p. 780-5.

[3] Bondarenko A, Dung PM, Kowalski RA, Toni F. An Abstract, Argumentation-Theoretic Approach to Default Reasoning. Artif Intell. 1997;93:63-101.

[4] Dvořák W, Woltran S. Complexity of abstract argumentation under a claim-centric view. Artif Intell. 2020;285:103290.

[5] Nielsen SH, Parsons S. A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In: Proceedings of ArgMAS 2006. Springer; 2006. p. 54-73.

[6] Janhunen T. Representing Normal Programs with Clauses. In: Proceedings of ECAI 2004. IOS Press; 2004. p. 358-62.

[7] Caminada M, Schulz C. On the Equivalence between Assumption-Based Argumentation and Logic Programming. J Artif Intell Res. 2017;60:779-825.

[8] Dvořák W, Rapberger A, Woltran S. Argumentation Semantics under a Claim-centric View: Properties, Expressiveness and Relation to SETAFs. In: Proceedings of KR 2020; 2020. p. 341-50.

[9] Dvořák W, Rapberger A, Woltran S. On the Relation Between Claim-Augmented Argumentation Frameworks and Collective Attacks. In: Proceedings of ECAI 2020. vol. 325. IOS Press; 2020. p. 721-8.

[10] Dvořák W, Keshavarzi Zafarghandi A, Woltran S. Expressiveness of SETAFs and Support-Free ADFs Under 3-Valued Semantics. In: Proceedings of COMMA 2020. IOS Press; 2020. p. 191-202.

[11] Alcântara JFL, Sá S. Equivalence Results between SETAF and Attacking Abstract Dialectical Frameworks. In: Proceedings NMR 2021; 2021. p. 139-48.

[12] Polberg S. Developing the Abstract Dialectical Framework [PhD Thesis]. Vienna University of Technology, Institute of Information Systems; 2017.

[13] Strass H. Approximating operators and semantics for abstract dialectical frameworks. Artif Intell. 2013;205:39-70.

[14] Alcântara JFL, Sá S, Guadarrama JCA. On the Equivalence Between Abstract Dialectical Frameworks and Logic Programs. Theory Pract Log Program. 2019;19(5-6):941-56.

[15] Caminada M, Sá S, Alcântara J, Dvořák W. On the equivalence between logic programming semantics and argumentation semantics. Int J Approx Reasoning. 2015;58:87-111.

[16] Berge C. Hypergraphs - combinatorics of finite sets. vol. 45 of North-Holland mathematical library. North-Holland; 1989.

[17] Cyras K, Fan X, Schulz C, Toni F. Assumption-Based Argumentation: Disputes, Explanations, Preferences. In: Handbook of Formal Argumentation. College Publications; 2018. p. 365-408.

[18] Bikakis A, Cohen A, Dvořák W, Flouris G, Parsons S. Joint Attacks and Accrual in Argumentation Frameworks. FLAP. 2021;8(6):1437-501.

[19] Flouris G, Bikakis A. A comprehensive study of argumentation frameworks with sets of attacking arguments. Int J Approx Reason. 2019;109:55-86.

[20] Dvořák W, König M, Ulbricht M, Woltran S. A Reduct-Driven Study of Argumentation Frameworks With Collective Attacks. In: Proceedings of NMR 2021; 2021. p. 285-94.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220155

On the Impact of Data Selection when Applying Machine Learning in Abstract Argumentation

Isabelle KUHLMANN, Thorsten WUJEK and Matthias THIMM
Artificial Intelligence Group, University of Hagen, Germany

Abstract. We examine the impact of both training and test data selection in machine learning applications for abstract argumentation, in terms of prediction accuracy and generalizability. For that, we first review previous studies from a data-centric perspective and conduct some experiments to back up our analysis. We further present a novel algorithm to generate particularly challenging argumentation frameworks wrt. the task of deciding skeptical acceptability under preferred semantics. Moreover, we investigate graph-theoretical aspects of the existing datasets and perform some experiments which show that some simple properties (such as in-degree and out-degree of an argument) are already quite strong indicators of whether or not an argument is skeptically accepted under preferred semantics.

Keywords. Abstract Argumentation, Approximate Reasoning, Artificial Neural Networks, Graph Neural Networks, Machine Learning

1. Introduction

Formal argumentation is a modern approach for non-monotonic reasoning within the general area of *Artificial Intelligence*. In particular, an *(abstract) argumentation framework* [1] consists of a set of arguments and a relation describing attacks between such arguments. Semantics are expressed in form of *extensions*, i.e., sets of arguments that meet certain prerequisites and are thus considered mutually acceptable. Typical computational problems in abstract argumentation are deciding whether an argument is included in some extension of a given semantics (or in all such extensions), or enumerating one (or all) extensions of a given semantics [2,3,4]. Most algorithmic approaches for solving such problems are sound and complete methods; see [4] for a recent overview. However, the high complexity of these problems—for instance, the problem of deciding whether an argument is included in all preferred extensions is Π_2^P-complete [2]—prohibits a scalable behavior of such approaches in the worst case. Thus, with the rise of *Deep Learning* approaches for numerous fields of application in recent years, a few authors suggested to use artificial neural networks for this task [5,6,7]. The advantage of a neural network is that, once it is trained properly, it can solve problems in time linear to the input. Nevertheless, this comes at the cost of exactness: the result is not guaranteed to be correct.

In this work, we review the existing literature on the topic of deep learning approaches for abstract argumentation with emphasis on the data used in the individual works. Although there is some overlap, none of the works use the same dataset, neither for training nor for testing purposes. This is (at least partly) due to the reason that for abstract argumentation, there exists no dedicated "standard" data set suitable for Machine Learning (ML) purposes, as opposed to, e.g., image processing, where datasets such as MNIST[1] or CIFAR[2] are well-known and publicly available. This paper takes a data-centric perspective and examines which properties a dataset in our field should possess. We perform an experimental analysis in which we explore the impact of different training and test sets on two different neural network architectures known from the literature. We also propose an algorithm to generate particularly challenging argumentation frameworks for the task of deciding skeptical acceptability under preferred semantics.

Furthermore, we investigate graph-theoretical properties of the datasets at hand. We show that some simple properties can be quite strong indicators of an argument's acceptability status. To illustrate this, we conduct some experiments in which we use a selection of "classical" ML methods which are only given the arguments' in-degrees and out-degrees as features. While the accuracy of these approaches is still lower than the accuracy of the deep learning methods, it is surprisingly high and raises doubt about the requirement to use complex deep learning methods for this purpose.

The remainder of this paper is structured as follows. We begin by giving an overview on abstract argumentation as well as related neural network techniques in Section 2. In Section 3, we take a closer look at the data used in the context of existing deep learning solutions for abstract argumentation and propose a new dataset which is particularly challenging for the task of deciding skeptical acceptability under preferred semantics. We conduct some experimental analysis on this new dataset as well as existing ones in Section 4. Section 5 provides a discussion and deeper analysis, in particular regarding graph-theoretical properties, and Section 6 concludes this work.

2. Preliminaries

We provide the basics of abstract argumentation in Section 2.1 and give an overview on existing works using artificial neural networks for abstract argumentation in Section 2.2.

2.1. Abstract Argumentation

An abstract argumentation framework (AF) [1] is a tuple $F = (\mathsf{Args}, R)$, where Args is the set of arguments and $R \subseteq \mathsf{Args} \times \mathsf{Args}$ is the attack relation. An argument $a \in \mathsf{Args}$ is said to *attack* another argument $b \in \mathsf{Args}$ if $(a,b) \in R$. We abbreviate

$$a_F^- = \{b \in \mathsf{Args} \mid (b,a) \in R\} \qquad\qquad a_F^+ = \{b \in \mathsf{Args} \mid (a,b) \in R\}$$

and analogously E_F^- and E_F^+ for a set $E \subseteq \mathsf{Args}$. An argument $a \in \mathsf{Args}$ is *defended by* a set of arguments $E \subseteq \mathsf{Args}$ if $a_F^- \subseteq E^+$. A set $E \subseteq \mathsf{Args}$ is *conflict-free* if $E \cap E^+ = \emptyset$. A set $E \subseteq \mathsf{Args}$ is called *admissible* (we also say $E \in \mathsf{ad}(F)$) if E is conflict-free and each $a \in E$ is defended by E within a given AF F. E is a *preferred* extension (i.e., $E \in \mathsf{pr}(F)$) if $E \in \mathsf{ad}(F)$ and for every $E' \in \mathsf{ad}(F)$, $E \not\subset E'$. An argument $a \in \mathsf{Args}$ is *skeptically accepted wrt. preferred semantics* (abbreviated pa) iff a is contained in

[1] http://yann.lecun.com/exdb/mnist/

[2] https://www.cs.toronto.edu/~kriz/cifar.html

every preferred extension. Let $\mathsf{pa}(F)$ denote the set of all pa arguments in F and define $\mathsf{pna}(F) = \mathsf{Args} \setminus \mathsf{pa}(F)$. We denote the computational problem of deciding whether an argument is skeptically accepted wrt. preferred semantics as $\mathsf{DS}_{\mathsf{pr}}$. An argument $a \in \mathsf{Args}$ is *credulously accepted wrt. preferred semantics* iff a is contained in some preferred extension.

We say that $E \in \mathsf{co}(F)$ (E is *complete*) if $E \in \mathsf{ad}(F)$ and for each $a \in \mathsf{Args}$ defended by E in F, it holds that $a \in E$. We define that E is a *grounded* extension if $E \in \mathsf{co}(F)$ and for every $E' \in \mathsf{co}(F)$, $E' \not\subset E$. Moreover, E is an *ideal* extension if $E \in \mathsf{ad}(F)$, for every $E' \in \mathsf{pr}(F)$, $E \subseteq E'$, and E is maximal (wrt. set inclusion) with these two properties. Note that every AF F has a uniquely defined grounded extension E_{gr} [1] and a uniquely defined ideal extension E_{id} [8] and that $E_{\mathsf{gr}} \subseteq E_{\mathsf{id}} \subseteq \mathsf{pa}(F)$. An argument $a \in \mathsf{Args}$ is *accepted wrt. grounded semantics* (abbreviated ga), if a is contained in the grounded extension E_{gr}. Let $\mathsf{ga}(F)$ denote the set of all ga arguments and let $\mathsf{ia}(F)$ be the corresponding notion for ideal semantics.

2.2. *Artificial Neural Networks for Abstract Argumentation*

The purpose of an ML approach, such as an artificial neural network, is to “learn” from given data (i.e., the *training set*), in order to subsequently apply the acquired “knowledge” to previously unknown data (e.g., the *test set* during evaluation). The term *validation set* refers to a dataset that is essentially used as a trial test set. It is used during the training process of a neural network to check how well it would perform on unknown data at certain training stages. A typical problem for the application of ML is that of classification. In abstract argumentation, our objective is to decide whether an argument is acceptable or not under a given semantics and wrt. a given reasoning mode (credulous or skeptical). Therefore, we aim to classify an argument as *acceptable* or *not acceptable*. More precisely, we can train, e.g., a neural network using a set of AFs with labels (*accepted/not accepted*) for each argument.

There are a few works about the application of neural networks to decide the acceptability status of arguments [5,6,7]. A first work [5] makes use of so-called Graph Convolutional Networks (GCNs) as proposed by Kipf and Welling [9]. The authors consider the problem of credulous acceptance under preferred semantics. Malmqvist et al. [7] improve the GCN approach used by [5] by proposing a randomized training regime as well as a scheme to dynamically balance the training data. They consider both credulous and skeptical acceptance under preferred semantics. Craandijk and Bex [6] introduce an Argumentation Graph Neural Network (AGNN) which learns a message-passing algorithm. The authors apply their approach on the tasks of deciding credulous and skeptical acceptance under all four classical semantics (i.e., *complete*, *grounded*, *preferred*, and *stable* semantics). The latter work is the most promising one so far—the authors report almost perfect predictions in their evaluation.

3. Selection of Data for Abstract Argumentation

As this paper takes a data-centric perspective, we now investigate how the problem of data selection was approached in the previously mentioned works (Section 2.2) and we discuss these design choices. In this paper we consider only the problem $\mathsf{DS}_{\mathsf{pr}}$, but note that some of the works mentioned above also consider other problems.

Kuhlmann and Thimm [5] generate training sets of different sizes using the *probo Benchmark Suite*[3] [10] as well as *AFBenchGen*[4] [11], which include a total of six different graph generators. The authors create datasets of different sizes which contain between 30 and 600 AFs, each consisting of 100–400 arguments. A test set of 120 AFs is generated in the same manner. In addition, they use some benchmark data from the *International Competition on Computational Models of Argumentation*[5] (ICCMA) 2017 for testing purposes. During training, a fraction of the training set is used as a validation set. Malmqvist et al. [7] used a dataset of 900 AFs from ICCMA 2017, divided into training set (90%) and test set (10%). Again, part of the training data is used as a validation set. Craandijk and Bex [6] use the same generators as [5], i.e., those included in the Probo Benchmark Suite and AFBenchGen, and generate a test set consisting of 1000 AFs with exactly 25 arguments each. A fixed validation set is generated analogously. As a training set they create a total of one million AFs with $5 \leq |\mathsf{Args}| \leq 25$.

All works above use different datasets, which makes their results difficult to compare. Further, none of the papers poses the question what a suitable dataset should look like. To begin with, a test set should contain instances with various properties (different sizes, complexity, etc.), and at least some of them should be considered "challenging"—after all, the purpose of using a deep learning approach for abstract argumentation is to solve problems faster than with an exact approach. While at least some instances of the ICCMA dataset can be considered "challenging" (as they are competition benchmarks), the data generated by the Probo Benchmark Suite and AFBenchGen is a different matter. Craandijk and Bex [6] use a test set consisting of AFs that are made up of 25 elements each by using the aforementioned generators. Deciding whether an argument in such an AF is pa merely takes 0.0013 s on average using, e.g., the μ-toksia solver[6] [12]. In comparison, an AGNN takes 0.00003 s on average[7] for the same task[8]. Consequently, there is a reduction in computation time when using the neural network approach, nevertheless this advantage must be weighed against the disadvantage that exactness can never be guaranteed with a deep learning method. In the case of such small AFs, it might be more practical to use an exact approach, since the *absolute* difference in runtimes (0,00127 s on average absolute difference between the correct system μ-toksia and the approximate system AGNN) is very likely to be negligible for real-world applications. However, in the case of more complex AFs, the fast solving time of a neural network may be the more suitable solution in practice. This highlights that researchers should select their (test) data in a way that it represents those cases, in which a deep learning approach would grant an actual advantage in practical applications.

Furthermore, the ICCMA instances exhibit a property that makes them somewhat less "challenging" as well, as most arguments that are pa are also ga (see the $|\mathsf{ga}|$ and $|\mathsf{pa}|$ values of iccma-test in Table 1). Recall that the computational complexity of deciding $\mathsf{DS}_{\mathsf{pr}}$ is Π_P^2-complete [13] in general. However, there are certain easy cases where $\mathsf{DS}_{\mathsf{pr}}$ can be decided with much less effort. For example, arguments in the grounded ex-

[3] https://sourceforge.net/projects/probo/

[4] https://sourceforge.net/projects/afbenchgen/

[5] http://argumentationcompetition.org/

[6] System properties: 32 GB RAM, AMD® Ryzen 7 pro 5850u

[7] Note that we measured the time the model took to process the entire test set and afterwards divided it by the number of arguments in the test set (i.e., 25,000).

[8] System and training setup as described in Section 4.1

Table 1. Overview of all datasets used in the experiments. Let F be an arbirtrary AF. All columns including mean values additionally include the standard deviation.

Dataset	# AFs	# arguments ($=n$)	Mean # arguments per AF	Mean # attacks per AF	Mean $\|\mathsf{pa}(F)\|$	Mean $\|\mathsf{ia}(F)\|$	Mean $\|\mathsf{ga}(F)\|$
pbbg-train	100000	1898327	18.98 ±6.07	55.34 ±48.62	5.96 ±4.50	5.94 ±4.51	5.33 ±4.84
pbbg-test	1000	25000	25.00 ±0.0	84.97 ±60.96	7.07 ±5.70	7.02 ±5.74	6.14 ±6.15
kwt-train	1000	151000	151.00 ±0.0	6523.54 ±1728.29	69.44 ±23.47	69.32 ±23.82	16.31 ±34.45
kwt-test	1000	151000	151.00 ±0.0	6567.07 ±1734.52	69.12 ±23.63	68.99 ±23.99	15.90 ±34.16
iccma-test	450	292221	649.38 ±1398.99	52734.64 ±175454.90	70.23 ±333.51	64.233 ±324.47	59.08 ±313.15

tension are always pa (and the grounded extension can be computed in polynomial time) as well as arguments in the ideal extension [14] (where related problems are "only" Θ_2^P-complete), and arguments attacked by some admissible set are never pa (and deciding this is a problem in NP). We developed a graph generator (to which we refer by KWT in the following) that is tailored towards generating abstract argumentation frameworks that are particularly hard for tasks related to $\mathsf{DS}_{\mathsf{pr}}$ by avoiding (as much as possible) these easy cases. The graph generator KWT takes as parameters (among others) the total number of arguments, the number of arguments to be pa, the number of arguments to be contained in at least one preferred extension, and the number of preferred extensions. Arguments are associated to the different preferred extensions (at random, using some further parameters for randomisation) and each argument of each extension is attacked by at least one other argument (so we will have a small grounded extension in most cases). We performed some simple experiments to verify that these graphs are indeed relatively hard for problems related to $\mathsf{DS}_{\mathsf{pr}}$, but a careful analysis of this is part of ongoing work (and also not relevant for the work reported here, due to our results below). In our experiments we generated argumentation frameworks with 151 arguments, 60–90 pa arguments, 15–60 further arguments that are in at least one extension, and 100–200 preferred extensions. The graph generator[9] and the script[10] we used can be found online.

4. Experimental Analysis

We conduct two experiments. First we examine different test sets for the same trained model (an AGNN as presented by Craandijk and Bex [6]), and, as follow-up work, we use an alternative training set in order to achieve more accurate results for one of the more challenging test sets. We repeat these experiments with a different neural network model which was proposed in [5].

4.1. Experimental Setup

We generate AFs as in [6]. We create a test set and a validation set containing 1000 AFs each, with each AF consisting of exactly 25 arguments (i.e., $|\mathsf{Args}| = 25$), as well as a training set consisting of AFs with $5 \leq |\mathsf{Args}| \leq 25$. We denote these sets pbbg-test, pbbg-val, and pbbg-train, respectively. However, we only create 100,000 AFs for the training set, as opposed to Craandijk and Bex, who used 1 million AFs, as the results of a training with 100,000 AFs are expressive enough (see Section 4.2). See Table 1 for

[9] http://tweetyproject.org/r/?r=kwt_gen

[10] http://tweetyproject.org/r/?r=kwt_gen_ex

a statistical overview of the training and test set we generated. To actually generate the AFs and corresponding solutions, we use the AGNN framework[11], which also offers the option to train and evaluate other neural network models.

As a second dataset, we generate a total of 2200 KWT instances[12] as described in Section 3. We split this data into a training set (kwt-train) consisting of 1000 AFs, a test set (kwt-test), also consisting of 1000 instances, and a validation set (kwt-val) of 200 instances. Moreover, we use part of the ICCMA 2017 data as an additional test set[13] (iccma-test[14]). In total, we use 450 AFs[15] from groups A, B, and C, spanning all difficulty levels. Details on all these sets are displayed in Table 1 The corresponding solutions were generated using the solvers *Pyglaf* [15] and *μ-toksia* [12].

In our first experiment, we train[16] an AGNN as in [6] (the only difference being the smaller training set). Hence, for training we use pbbg-train, for validation pbbg-val, and for testing pbbg-test. We then test the trained model on the other two test sets (kwt-test and iccma-test) to examine whether it performs similarly well. As a follow-up experiment, we train an AGNN on kwt-train, in order to inspect if it is able to perform better on kwt-test. Again, we use all three test sets. Afterwards, we conduct the same two experiments on an implementation of the FM2 model [5] which is also included in the AGNN framework. In each experiment, the network is trained for 300 epochs, i.e., the training set is passed through the model 300 times.

In order to quantify our results, we use the Matthews Correlation Coefficient (MCC), as well as accuracy, True Positive Rate (TPR) and True Negative Rate (TNR). Let TP/FP and TN/FN denote *true/false positives* and *true/false negatives*, respectively, where the positive class are the pa arguments. Then we can define $\mathrm{MCC} = \frac{\mathsf{TP}\cdot\mathsf{TN}-\mathsf{FP}\cdot\mathsf{FN}}{\sqrt{(\mathsf{TP}+\mathsf{FP})\cdot(\mathsf{TP}+\mathsf{FN})\cdot(\mathsf{TN}+\mathsf{FP})\cdot(\mathsf{TN}+\mathsf{FN})}}$. Moreover, $\mathrm{TPR} = \mathsf{TP}/(\mathsf{TP}+\mathsf{FN})$, and $\mathrm{TNR} = \mathsf{TN}/(\mathsf{TN}+\mathsf{FP})$. Accuracy is defined as $\frac{\mathsf{TP}+\mathsf{TN}}{\mathsf{TP}+\mathsf{TN}+\mathsf{FP}+\mathsf{FN}}$, and precision as $\mathsf{TP}/\mathsf{TP}+\mathsf{FP}$.

4.2. Results

Reproducing the training procedure described by Craandijk and Bex [6] yields an MCC of 0.962. Although this value is slightly lower than the one provided in the original paper (0.997), it is still quite high. Besides, the small discrepancy can be explained by the fact that we used a smaller training set, and could be compensated by the use of more message passing steps during the testing phase. However, the MCC values regarding kwt-test and iccma-test are significantly lower (0.631 and 0.507, respectively). As Table 2 reveals, this is mostly due to a low TPR, meaning that arguments that are pa are not recognized as such in many cases. Training an AGNN with kwt-train results in an increased MCC for kwt-test (0.927). This shows that AGNNs are actually able to learn KWT data quite well if they are exposed to such data during training. The MCCs regarding the other two

[11] https://github.com/DennisCraandijk/DL-abstract-argumentation

[12] https://fernuni-hagen.sciebo.de/s/ZEmipULEN05FxxC

[13] Due to hardware limitations (in particular concerning the GPU memory), we could not use the ICCMA data for training.

[14] https://fernuni-hagen.sciebo.de/s/qjbSqMETOjb2JSY

[15] https://fernuni-hagen.sciebo.de/s/ZvNyWJ4af4ehEaB

[16] All computations were conducted on a computer equipped with an NVIDIA GeForce GTX 980 Ti GPU (6144 MB internal memory), an AMD® Ryzen 5 2600 processor, and 16 GB RAM.

Table 2. Overview of the experimental results.

Training Set	Test set	MCC	Accuracy	TPR	TNR	Precision
Model: AGNN						
pbbg-train	pbbg-test	0.962	0.985	0.964	0.993	0.981
	kwt-test	0.631	0.801	0.588	0.980	0.962
	iccma-test	0.507	0.847	0.446	0.974	0.845
kwt-train	pbbg-test	0.693	0.880	0.666	0.965	0.882
	kwt-test	0.927	0.962	1.000	0.930	0.924
	iccma-test	0.250	0.780	0.312	0.928	0.577
Model: FM2						
pbbg-train	pbbg-test	0.558	0.822	0.670	0.882	0.692
	kwt-test	0.359	0.642	0.220	0.998	0.989
	iccma-test	0.211	0.763	0.315	0.905	0.512
kwt-train	pbbg-test	0.078	0.719	0.030	0.991	0.568
	kwt-test	0.364	0.643	0.220	1.000	1.000
	iccma-test	0.055	0.761	0.016	0.996	0.583

test sets are lower than in the previous experiment (see Table 2). Thus, a training on KWT data alone is not practical either, and a future standard training set should probably contain instances of both KWT data and other datasets, such as pbbg-train. But these results suggest that the AGNN does no actually learn the concept (or an approximate concept) of skeptical acceptance wrt. preferred semantics, but rather particular properties of the benchmarks that correlate with it.

The results of our experiments with the FM2 network point in the same direction as the previously described ones. Training the network with pbbg-train results in a lower MCC wrt. kwt-test and iccma-test than wrt. pbbg-test (see the bottom half of Table 2). However, the FM2 model is clearly less accurate (in particular wrt. accepted arguments) and does not generalize very well. This problem becomes even more apparent when the network is trained with kwt-train. The resulting model does not significantly recognize accepted arguments from pbbg-test or iccma-test. Nevertheless, one should note that the problem of FM2 with imbalanced data has been recognized in the literature [5,7] and that other parameters could potentially improve the model's accuracy. Also, the fact that FM2 does not learn KWT data very well supports the hypothesis that the KWT instances are rather challenging—which was our goal when we developed the generator.

5. Discussion and Further Analysis

In the following, we look a little bit deeper into the actual approach that ML methods take to solve $\mathsf{DS}_{\mathsf{pr}}$, and how the data may bias learning. In particular, we have a look at the different graph types and how the accuracy of the existing approaches varies over these. Furthermore, we show that classical ML approaches perform already quite well when just considering very simple graph-theoretic features, suggesting that the complex deep learning approaches tend to simply learn these features as well.

Table 3. AGNN results on different subsets of iccma-test. Rows containing sets of AFs generated using methods which are *not* included in the generation process of pbbg-train and pbbg-test are marked in gray.

	Training Set									
	pbbg-train					kwt-train				
Test Set	MCC	Accuracy	TPR	TNR	Precision	MCC	Accuracy	TPR	TNR	Precision
ABA2AF	0.804	0.989	0.692	0.999	0.947	0.680	0.978	0.764	0.985	0.626
admbuster	−0.110	0.486	0.003	0.968	0.086	−0.454	0.329	0.000	0.657	0.000
BA	0.946	0.973	0.977	0.970	0.963	0.645	0.812	0.917	0.729	0.725
ER	−0.001	0.999	0.000	0.999	0.000	0.009	0.998	0.012	0.999	0.028
grd	0.690	0.970	0.600	0.992	0.827	0.265	0.935	0.226	0.979	0.391
Planning2AF	0.660	0.914	0.759	0.938	0.655	0.227	0.693	0.552	0.715	0.231
scc	0.135	0.998	0.127	0.999	0.151	0.016	0.989	0.049	0.990	0.008
sembuster	–	0.687	–	0.687	–	–	0.804	–	0.804	–
stb	0.463	0.926	0.329	0.988	0.742	0.172	0.904	0.097	0.987	0.438
traffic	0.789	0.931	0.723	0.989	0.947	0.541	0.858	0.499	0.958	0.768
WS	0.262	0.992	0.233	0.997	0.302	0.067	0.986	0.091	0.991	0.062

5.1. Analysis of Graph Types

As iccma-test overall seems to be hard to predict, we divide the dataset into multiple subsets. Each subset contains data from one distinct generator or source[17]. The sets grd, scc, and stb contain those instances produced by the generators of the *Probo Benchmark Suite*, i.e., the *GroundedGenerator*, *SccGenerator*, and *StableGenerator*, respectively. The sets BA, ER, and WS contain the instances generated using *AFBenchGen2*, i.e., they correspond to the *Barabasi-Albert*, *Erdős-Rényi*, and *Watts-Strogatz* approach, respectively. Consequently, these six graph "types" are also included in pbbg-train (and pbbg-test)[18]. The other five subsets, namely ABA2AF, admbuster, Planning2AF, sembuster, and traffic, conform to the remaining five generators (for a detailed explanation, see the official competition benchmark report [16]).

We considered the AGNN models trained on pbbg-train and kwt-train which we described in Section 4.2 and tested them on all individual subsets of iccma-test. Table 3 shows that the results vary widely. For instance wrt. pbbg-train, the BA instances are predicted quite accurately, with an MCC of 0.946. On the other hand, wrt. the ER subset, the model essentially just learned to classify all arguments as pna, which results in a TNR of 0.999, but a TPR of 0.000 (and an MCC of −0.001). Further, we can observe that both models tend to perform similarly. Although the model trained on pbbg-train overall performs superior to the one trained on kwt-train, they both perform best (wrt. MCC) on ABA2AF and BA, and worst on admbuster and ER. Moreover, it should be highlighted once again that the model trained on pbbg-train performs very poorly on ER—although the training set itself contained AFs generated using the Erdős-Rényi algorithm. Another interesting observation is that the AGNN performs very poorly on the admbuster set [17]. This is surprising because the admbuster graphs are acyclic and $\mathsf{pa}(F)$ coincides with the grounded extension in all its instances F. So, although "being in the grounded extension" is a sufficient condition for pa and easy to compute for classical algorithms, it is not learned by the deep learning approach.

5.2. Impact of In-degree and Out-degree

An advantage of deep learning approaches to ML is that the tedious task of feature engineering is taken over by the learning approach. The approaches [5,6,7] all learn the

[17] Overview of the AFs belonging to each subset: https://fernuni-hagen.sciebo.de/s/pySMMAfE7zEzn6i

[18] Note, however, that different parameters were used.

features used for classification implicitly when presented with the raw input data. While this is beneficial from the point of view of the engineer, it also makes the model hard to explain as it is not clear, how exactly recommendations are drawn from the input. This can also lead to models that make predictions on correlations rather than on causal relationships. A famous example of such a misbehavior comes from image recognition [18]. There, an ML approach was given a set of images classified either as wolf or dog (husky) and a classifier was trained to predict these two classes. But rather than identifying some intrinsic feature to distinguish wolves from dogs, the ML approach learned the feature of "snow in the background". In both the training and test set, most images of wolves had snow in the background, and this feature actually could be used very well for distinguishing wolves from dogs. However, it is clear that "snow in the background" is not a good feature to describe what makes a wolf a wolf.

While the above example is an extreme case of an ML approach focussing on correlation, we would like to analyse whether something similar happens in the case of predicting $\mathsf{DS}_{\mathsf{pr}}$ in abstract argumentation frameworks. For that we analysed the distributions of several graph-theoretic features, such as degrees, diameter, clustering coefficient, etc., of the graphs in our datasets, with respect to the differences between pa and pna arguments. We found out that the two features of in- and out-degree (i.e., the number of incoming and outgoing attacks per argument) already distinguish these two classes very well. Table 4, which comprises the mean degrees of pas and pnas wrt. all datasets used in this work, shows that both mean in-degree and mean out-degree are in all cases lower wrt. pa compared to pna. Regarding the out-degree, this is only a tendency, and there exist exceptions—for instance, we examined a dataset consisting of 1000 AFs (25 arguments each) generated by Probo's *GroundedGenerator*, where the mean out-degree per pa (14.83) was higher than the mean out-degree per pna (10.75). Regarding the in-degree, it is also possible to construct cases where the mean pa in-degree is higher than the mean pna in-degree. However, the clear tendency of lower in-degrees of pa is clearly visible. This is no coincidence in the data, but an intrinsic property of $\mathsf{DS}_{\mathsf{pr}}$: the number of the incident attacks in $\mathsf{pna}(F)$ is necessarily at least as large as the number of incident attacks in $\mathsf{pa}(F)$ (this is actually true for all conflict-free semantics).

Proposition 1. *For any* AF F, $|\mathsf{pa}(F)^+_F| + |\mathsf{pa}(F)^-_F| \leq |\mathsf{pna}(F)^+_F| + |\mathsf{pna}(F)^-_F|$.

Proof. Let $(a,b) \in R$. Since $\mathsf{pa}(F)$ is conflict-free, it follows that either

1. $a \in \mathsf{pa}(F), b \in \mathsf{pna}(F)$,
2. $a \in \mathsf{pna}(F), b \in \mathsf{pa}(F)$, or
3. $a \in \mathsf{pna}(F), b \in \mathsf{pna}(F)$.

In the first two cases, both sets $\mathsf{pa}(F)$ and $\mathsf{pna}(F)$ are incident to the attack (a,b), while in case three, only $\mathsf{pna}(F)$ is incident to the attack (with both end points). Summing over all attacks, the claim follows. □

The above observation becomes important, when we recall what "exact" problem the classifiers are learned upon. The actual problem is that, given an argumentation framework, predict pa/pna *simultaneously* for all arguments of the framework. As already discussed in Section 3, for many arguments the problem of deciding whether they are pa/pna is actually quite easy. With the additional information that low in-degrees are a

Table 4. Comparison of the mean number (and standard deviation) of outgoing and incoming attacks (i.e., in-degree and out-degree) wrt. the set of skeptically accepted arguments and the set of unaccepted arguments under preferred semantics.

Dataset	Mean out-degree per pa		Mean out-degree per pna		Mean in-degree per pa		Mean in-degree per pna	
pbbg-train	2.15	±2.02	2.71	±1.93	0.82	±0.98	3.29	±1.86
pbbg-test	2.06	±2.09	3.41	±2.38	0.84	±1.04	3.90	±2.25
kwt-train	15.92	±12.83	66.45	±12.40	13.64	±10.77	69.17	±12.01
kwt-test	15.97	±12.78	66.62	±12.33	13.88	±10.75	69.27	±11.92
iccma-test	20.69	±69.57	56.71	±133.83	12.11	±48.15	57.07	±134.16

Table 5. Results of a naive classifier which uses the constraint "$a_{\mathsf{in}} < \mathsf{mean}_{\mathsf{in}}^{\mathsf{pa}} \vee a_{\mathsf{out}} < \mathsf{mean}_{\mathsf{out}}^{\mathsf{pa}}$?" to classify arguments as pa or pna.

Training Set	Test Set	MCC	Accuracy	TPR	TNR	Precision
pbbg-train	pbbg-test	0.396	0.674	0.825	0.615	0.458
pbbg-train	iccma-test	0.364	0.737	0.827	0.726	0.268
kwt-train	kwt-test	0.739	0.868	0.768	0.951	0.930

very good indicator for having a pa argument, we can already classify many arguments correctly.

In the following, we will describe the results of some additional experiments to explore the impact of in-degree and out-degree in classification tasks. For that, we simplified the instances of our datasets to a tabular format, in which each row only contains an argument ID, the argument's in-degree, out-degree, and label (pa or pna). So, the actual argumentation frameworks are not given to the learning algorithm as input! We then train several ML algorithms on this data and measure the resulting MCC, accuracy, TPR and TNR. To begin with, we define a naive classifier whose "training" consists of calculating the mean pa in-degree ($\mathsf{mean}_{\mathsf{in}}^{\mathsf{pa}}$) and out-degree ($\mathsf{mean}_{\mathsf{out}}^{\mathsf{pa}}$) of the training set. The classification process then simply consists of checking whether the in-degree or the out-degree of a given argument a (a_{in} and a_{out}) is smaller than $\mathsf{mean}_{\mathsf{in}}^{\mathsf{pa}}$or $\mathsf{mean}_{\mathsf{out}}^{\mathsf{pa}}$, respectively: if one of them is indeed smaller, the argument is classified as pa, if this is not the case, it is classified as pna. Although the results are, as one would expect, far from perfect, Table 5 shows that in-degree and out-degree alone are quite good indicators of the acceptability status of an argument. In particular, accepted arguments are predicted surprisingly accurately, with a TPR between 0.768 and 0.825.

Furthermore, we train a number of "classical" ML models. Again, we use the formatted datasets which only contain the in-degree and out-degree of each node, but no further information about the graph structure. To be precise, we consider the following methods [19]: k-Nearest Neighbors (KNN) (with $k = 5$), Naive Bayes (NB), Decision Tree (DT), and Random Forest (RF)[19]. The results, displayed in Table 6, show that the previously described results yielded by the naive classifier can generally be improved by using a more sophisticated approach. In particular, the results on kwt-test are quite accurate, with an MCC of 0.924 when using an RF. While DT and RF exhibit the best performance wrt. MCC, one should notice that NB offers the most "balanced" results wrt. TPR and TNR, in particular regarding pbbg-train. Besides, it is noticeable that DT and RF feature very similar, or even identical results in all cases. This is most likely due

[19] We used the implementations and default parameters provided by https://scikit-learn.org/stable/.

Table 6. Overview of test results wrt. multiple training and test sets as well as four different ML methods.

Training Set	Test Set	Approach	MCC	Accuracy	TPR	TNR	Precision
pbbg-train	pbbg-test	KNN	0.567	0.834	0.576	0.936	0.779
		NB	0.514	0.761	0.831	0.733	0.551
		DT	**0.630**	0.857	0.615	0.952	0.836
		RF	**0.630**	0.857	0.615	0.952	0.836
pbbg-train	iccma-test	KNN	0.191	0.832	0.309	0.895	0.264
		NB	0.389	0.775	0.793	0.773	0.298
		DT	**0.399**	0.911	0.200	0.997	0.895
		RF	**0.399**	0.911	0.200	0.997	0.895
kwt-train	kwt-test	KNN	0.884	0.942	0.952	0.934	0.924
		NB	0.868	0.934	0.935	0.934	0.923
		DT	0.923	0.960	0.998	0.929	0.922
		RF	**0.924**	0.961	0.998	0.929	0.922

to the fact that the datasets only possess two features, which might lead to the decision trees in the random forest to be similar (or identical) to a single decision tree.

Overall, our results suggest that classical ML techniques trained only on the features *in-degree* and *out-degree* might not be accurate enough for practical applications, yet they also show that these two simple features are still very strong indicators for the acceptance status of an argument. Therefore, the question arises whether (and if so, in which manner) such a dominant feature influences a neural network's training process.

6. Conclusion

The objective of this work was to shed some light on the importance of data selection—a common problem in ML research, but less common in knowledge representation and reasoning. To successfully bridge the gap between both fields, we need to take different perspectives. Our experiments showed that a training set consisting of only rather small AFs (5–25 arguments) offers little room for variability. Thus, a neural network cannot solve more complex cases as accurately. However, we also saw that a neural network (in particular the AGNN architecture [6]) is in fact able to learn more complex features if it is exposed to them during training. We also discussed the question whether the application of ML techniques is even useful for small and simple AFs. Such cases could be solved by an exact approach in a reasonable amount of time without any losses in terms of accuracy. Further, there are other "simple" cases which should be taken into consideration. For example, when regarding DS$_{\text{pr}}$, a dataset should contain a significant number of arguments that are pa but not ga, since the grounded extension can be computed in polynomial time.

Furthermore, as a subject of future work, we need to take a closer look at graph-theoretical aspects. For instance, a dataset could benefit from AFs which include accepted arguments that have similar properties (such as in-/out-degree) as arguments that are not accepted. Therefore, a neural network would need to learn less superficial features to learn the acceptability status of arguments. Another topic of future work is to examine other semantics and tasks more closely, as we only focused on DS$_{\text{pr}}$. Yet another aspect that could be explored is to use ML techniques to guide sound and complete solvers, in order to accelerate their execution times. Moreover, the compilation of a standard dataset that includes all of the above perspectives and aspects could facilitate future research in this area.

Acknowledgements

We would like to thank Lars Malmqvist for providing a large sum of solutions for the ICCMA 2017 benchmark instances. We would also like to thank Mark Mints for his active support concerning various GPU issues. The research reported here was supported by the Deutsche Forschungsgemeinschaft under grant 375588274.

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995;77(2):321-58.

[2] Dvorák W, Dunne PE. Computational problems in formal argumentation and their complexity. Handbook of formal argumentation. 2018:631-87.

[3] Cerutti F, Gaggl SA, Thimm M, Wallner JP. Foundations of Implementations for Formal Argumentation. In: Handbook of Formal Argumentation. College Publications; 2018. p. 2623-705.

[4] Lagniez JM, Lonca E, Mailly JG, Rossit J. Design and Results of ICCMA 2021. arXiv preprint arXiv:210908884. 2021.

[5] Kuhlmann I, Thimm M. Using graph convolutional networks for approximate reasoning with abstract argumentation frameworks: A feasibility study. In: International Conference on Scalable Uncertainty Management. Springer; 2019. p. 24-37.

[6] Craandijk D, Bex F. Deep learning for abstract argumentation semantics. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence; 2020. p. 1667-73.

[7] Malmqvist L, Yuan T, Nightingale P, Manandhar S. Determining the Acceptability of Abstract Arguments with Graph Convolutional Networks. In: SAFA@ COMMA; 2020. p. 47-56.

[8] Dung PM, Mancarella P, Toni F. Computing ideal sceptical argumentation. Artificial Intelligence. 2007;171(10-15):642-74.

[9] Kipf TN, Welling M. Semi-Supervised Classification with Graph Convolutional Networks. In: International Conference on Learning Representations (ICLR); 2017. .

[10] Cerutti F, Oren N, Strass H, Thimm M, Vallati M. A Benchmark Framework for a Computational Argumentation Competition. In: COMMA; 2014. p. 459-60.

[11] Cerutti F, Giacomin M, Vallati M. Generating Challenging Benchmark AFs. COMMA. 2014;14:457-8.

[12] Niskanen A, Järvisalo M. μ-toksia: An efficient abstract argumentation reasoner. In: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning; 2020. p. 800-4.

[13] Dunne PE, Bench-Capon TJM. Coherence in finite argument systems. Artificial Intelligence. 2002;141(1/2):187-203.

[14] Dunne PE. The computational complexity of ideal semantics. Artificial Intelligence. 2009 December;173(18):1559-91.

[15] Alviano M. The pyglaf argumentation reasoner. In: Technical Communications of the 33rd International Conference on Logic Programming; 2018. p. 2:1-2:3.

[16] Gaggl SA, Linsbichler T, Maratea M, Woltran S. Design and results of the Second International Competition on Computational Models of Argumentation. Artificial Intelligence. 2020;279:103193.

[17] Caminada M, Podlaszewski M. AdmBuster: a benchmark example for (strong) admissibility; 2017. The Second International Competition on Computational Models of Argumentation (ICCMA'17).

[18] Ribeiro MT, Singh S, Guestrin C. " Why should i trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1135-44.

[19] Alpaydin E. Introduction to machine learning. MIT press; 2020.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220156

Algorithms for Reasoning in a Default Logic Instantiation of Assumption-Based Argumentation[1]

Tuomo LEHTONEN [a], Johannes P. WALLNER [b] and Matti JÄRVISALO [a]
[a] *University of Helsinki, Finland*
[b] *Graz University of Technology, Austria*
ORCiD ID: Tuomo Lehtonen https://orcid.org/0000-0001-6117-4854, Johannes P. Wallner https://orcid.org/0000-0002-3051-1966, Matti Järvisalo https://orcid.org/0000-0003-2572-063X

Abstract. Assumption-based argumentation (ABA) is one of the most-studied formalisms for structured argumentation. While ABA is a general formalism that can be instantiated with various different logics, most attention from the computational perspective has been focused on the logic programming (LP) instantiation of ABA. Going beyond the LP-instantiation, we develop an algorithmic approach to reasoning in the propositional default logic (DL) instantiation of ABA. Our approach is based on iterative applications of Boolean satisfiability (SAT) solvers as a natural choice for implementing derivations as entailment checks in DL. We instantiate the approach for deciding acceptance and for assumption-set enumeration in the DL-instantiation of ABA under several central argumentation semantics, and empirically evaluate an implementation of the approach.

Keywords. structured argumentation, assumption-based argumentation, default logic, decision procedures, SAT, counterexample-guided abstraction refinement

1. Introduction

Assumption-based argumentation (ABA) [1,2] is a central approach to structured argumentation [3,4,5]. ABA captures and generalizes different approaches default reasoning, constituting a general-purpose framework which can be instantiated for any formal logic to support various different application settings. Derivations of conclusions from assumptions via inference rules in the logic of choice give rise to arguments.

Arguably the most-studied ABA instantiation is the logic programming (LP) fragment of ABA [1] in which arguments are derived with logic programming rules. From the computational perspective, development of algorithmic approaches to central reasoning problems, such as acceptance, in ABA has focused on the LP-instantiation [6,7,8,9,10]. In the LP-instantiation, derivations are computable in polynomial time, which implies

[1] This work has been financially supported in part by Academy of Finland grant 322869 and the University of Helsinki Doctoral Programme in Computer Science DoCS. The authors wish to thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources.

that deciding acceptance in ABA is a computational problem contained in NP for various argumentation semantics [11]. However, algorithmic approaches to reasoning in ABA beyond the LP-instantiation, i.e., ABA instantiated with more expressive logics, and in particular ones for which derivations may be hard to compute, are scarce. Addressing this challenge, in this work we develop algorithms for reasoning in ABA instantiated for propositional default logic (DL) [12,13], which we refer to as DL-ABA.

In DL-ABA, derivations of arguments require checking entailment (coNP). This implies that reasoning about acceptance is beyond the first level of the polynomial hierarchy [11]. The need for entailment checking suggests using Boolean satisfiability (SAT) solvers [14] as a basis for developing approaches to reasoning in DL-ABA. Indeed, we employ incremental SAT solving for developing an approach that allows for deciding acceptance and the enumeration of assumption-sets under several central argumentation semantics. We provide an implementation of the approach and a first empirical evaluation of its scalability. Although there have been earlier developments for computing extensions in DL [15,16,17,18] which in particular corresponds to computing extensions under stable semantics in DL-ABA, to our best understanding our approach presented is the first in its generality for DL-ABA.

2. Preliminaries

We review background on assumption-based argumentation (ABA) [1,2] with propositional default logic (DL) [12] as the underlying deductive system.

The first ingredient for ABA is a deductive system $(\mathscr{L},\mathscr{R})$. For our purposes $\mathscr{L}$ is a set of propositional formulas and $\mathscr{R}$ a set of rules of the form $r = a_0 \leftarrow a_1,\ldots,a_n$ with $a_i \in \mathscr{L}$. We say that a_0 is the head of the rule ($head(r) = a_0$) and the set $\{a_1,\ldots,a_n\}$ is the body ($body(r) = \{a_1,\ldots,a_n\}$). A sentence $a \in \mathscr{L}$ is derivable from $A \subseteq \mathscr{L}$ (in symbols $A \vdash_{\mathscr{R}} a$) if either $a \in A$, or there is a sequence of rules $(r_1,\ldots,r_n)$ from $\mathscr{R}$ s.t. $head(r_n) = a$ and $body(r_i) \subseteq \bigcup_{j<i} head(r_j) \cup A$ for $1 \leq i \leq n$; that is, each body element of rules must be present either in A or as heads of previous rules in the sequence. We omit subscript $\mathscr{R}$ when clear from context. We assume that $(\mathscr{L}_p,\mathscr{R}_p)$ is a deductive system for propositional logic, i.e., $\mathscr{L}_p$ is the set of all propositional formulas and $A \vdash_{\mathscr{R}_p} a$ iff $A \models a$ (i.e., there is a derivation via $\mathscr{R}_p$ iff classical semantic entailment holds; one may select any sound and complete inference system for classical propositional logic as $\mathscr{R}_p$).

A propositional default theory is a pair $T = (W,D)$, where $W \subseteq \mathscr{L}_p$ and D is a set of default rules of the form $r = c \leftarrow a, Mb_1,\ldots,Mb_n$ with $c,a,b_1,\ldots,b_n \in \mathscr{L}_p$ and $Mb_i \notin \mathscr{L}_p$ (Mb_i are not propositional formulas). We refer to c as the conclusion, to a as the prerequisite, and to $\{Mb_1,\ldots,Mb_m\}$ as the justifications of the default rule r. We use the shorthands $M(r) = \{Mb_1,\ldots,Mb_n\}$, $prereq(r) = a$ and $conc(r) = c$. Intuitively, Mb is interpreted as $\neg b$ can not be proven and thus it is consistent to assume b.

We directly state ABA instantiated with propositional default logic.

Definition 1. *Let (W,D) be a propositional default theory. The assumption-based argumentation framework (ABF) corresponding to (W,D) is $F = (\mathscr{L},\mathscr{R},W,\mathscr{A},\bar{\ })$ with*

- $\mathscr{L} = \mathscr{L}_p \cup \{M\alpha \mid \alpha \in \mathscr{L}_p\}$,
- $\mathscr{R} = \mathscr{R}_p \cup D$,
- $\mathscr{A} = \{Mb \mid Mb$ *occurs in some default rule in D*, $b \in \mathscr{L}_p\}$, *and*

- *$\overline{}$ a function mapping $\mathscr{A}$ to $\mathscr{L}$ defined by $\overline{Mb} = \neg b$ for all $Mb \in \mathscr{A}$.*

An $Mb \in \mathscr{A}$ is an assumption (in a given ABF). For brevity, as (W,D) uniquely determines the corresponding ABF and as we focus on ABFs corresponding to propositional default theories, we identify (W,D) with the corresponding ABF F and write $F = (W,D)$ referring to the ABF corresponding to (W,D).

Given an ABF, derivability from a set of assumptions A is defined by $W \cup A \vdash_{\mathscr{R}} a$, i.e., a is derivable from W and the assumption set (note that $\mathscr{R}$ includes default rules from the default theory). A set of assumptions $A \subseteq \mathscr{A}$ attacks a set of assumptions $B \subseteq \mathscr{A}$ iff $W \cup A \vdash_{\mathscr{R}} \overline{Mb}$ for some $b \in B$, or equivalently, $W \cup A \vdash_{\mathscr{R}} \neg b$; that is, a set of assumptions is a set of justifications (which are "assumed"), and A attacks B, if one can derive the negation of some justification in B from the propositional theory W with all default rules whose justifications are met by A. For a singleton $B = \{b\}$ we say that A attacks b. Note that an atom b may be entailed although Mb is not, as assumptions are not part of the propositional vocabulary. In ABA terminology, an ABF is flat if assumptions do not occur as heads of rules, as is the case for DL-ABA.

A set of assumptions A is conflict-free if A does not attack A. A defends the assumption set B if A attacks each assumption set C that attacks B.

Definition 2. *For a given ABF $F = (W,D)$ and a conflict-free set of assumptions $A \subseteq \mathscr{A}$, we say A is*

- *admissible (in F) if A defends itself,*
- *complete (in F) if A is admissible and contains every assumption set it defends,*
- *grounded (in F) if A is $\subseteq$-minimally complete, and*
- *stable (in F) if A is conflict-free and attacks every assumption $a \in \mathscr{A} \setminus A$.*

A useful equivalent characterization for grounded semantics is by utilizing $\mathscr{F}_F$ for an ABF $F = (W,D)$, defined by $\mathscr{F}_F(S) = \{Mb \in \mathscr{A} \mid \{Mb\}$ defended by $S\}$ for an $S \subseteq \mathscr{A}$. The grounded assumption set is then the (unique) least fixed point of $\mathscr{F}_F$ [1]. There is a direct correspondence between Reiter's default extensions [12] and stable assumption sets: E is a default extension of (W,D) iff E' is a stable assumption set of $F = (W,D)$ with $E = \{a \mid W \cup E' \vdash a\} \cap \mathscr{L}_p$ [1]. An atom $x \in \mathscr{L}$ is credulously (resp., skeptically) accepted under semantics $\sigma \in \{adm, com, grd, stb\}$ (for admissible, complete, grounded, stable) if x is derivable from some (resp., all) σ-assumption set(s).

Example 1. *Let $W = (\neg b \vee \neg a) \wedge (\neg b \vee \neg c)$ and D contain four default rules: $(r_1 = a \leftarrow Ma)$, $(r_2 = b \leftarrow Mb)$, $(r_3 = c \leftarrow Mc)$, and $(r_4 = d \leftarrow a \wedge b, Md)$. For the corresponding ABF, $\mathscr{L}$ and $\mathscr{R}$ are together an extension of a propositional deductive system with defaults and the assumptions $\mathscr{A} = \{Ma, Mb, Mc, Md\}$. Contrariness is given by $\overline{Mx} = \neg x$ for $x \in \{a,b,c,d\}$. The four singleton assumption sets $\{Mx\}$ are all conflict-free. For instance, $\neg a$ is not derivable from W and Ma, i.e., $W \cup \{Ma\} \nvdash \neg a$. Using only rules from $\mathscr{R}_p$ (representing classical propositional entailment), $\neg a$ cannot be derived from $W \wedge Ma$. However, Ma can be derived and thereby via r_1 the atom a. Thus $W \cup \{Ma\} \vdash a$. Moreover, $\neg b$ is derived via the first clause in W. Thus $\{Ma\}$ attacks $\{Mb\}$. Symmetrically, $\{Mb\}$ attacks $\{Ma\}$. Only assumption sets including Mb derive $\neg a$. Thus $\{Ma\}$ attacks all assumption sets that attack $\{Ma\}$, indicating that $\{Ma\}$ is admissible. $\{Md\}$ does not attack any other set and is not attacked, as d does not occur anywhere outside r_4. As all other sets are attacked, $\{Md\}$ is the (unique) grounded assumption set. A com-*

plete and stable assumption set is $\{Ma, Md\}$. It holds that a is credulously accepted under complete semantics, since the complete assumption set $\{Ma, Md\}$ derives a. It holds that $\{Mb, Md\}$ (another complete assumption set) does not derive a. Thus a is not skeptically accepted. Finally, rule r_4 does not trigger: its prerequisite is never entailed by a conflict-free assumption set ($a \wedge b$ is unsatisfiable with W).

The complexity of ABA instantiated with propositional default logic was investigated in [13,11,19]. For flat ABA instantiated with a deductive system whose derivation complexity is in coNP, we have Σ_2^P-completeness for credulous reasoning under admissible and stable semantics (and since credulous acceptance under admissible and complete semantics coincides [2], also for complete semantics). Skeptical acceptance under admissibility is coNP-complete and Π_P^2-complete under stable semantics. The complexity of reasoning under grounded semantics (and skeptical acceptance for complete semantics) is in Δ_P^2 (i.e., decidable via a deterministic polynomial time algorithm with access to an NP oracle). Furthermore, the complexity of credulous and skeptical reasoning in propositional default logic (i.e., deciding whether a formula is entailed by one or all extensions according to Reiter [12]) coincides with the complexity of stable semantics [13].

3. A SAT-Based Approach to Deciding DL-ABA

Our SAT-based approach to deciding DL-ABA under different argumentation semantics is based on counterexample-guided abstraction refinement [20] which has earlier proven successful in the realm of abstract argumentation (see, e.g., [21,22]). For high-level intuition, our algorithms work by iteratively first "guessing" with a SAT solver a candidate assumption set A, and then employing a SAT solver to determine the set C of conclusions of default rules in D that are applicable by A. After this, we employ a SAT solver to check if A conforms to the semantics of interest (together with a query, in case of acceptance problems) based on C. If this is the case, the search terminates. If not, a counterexample witnessing this fact is obtained, and we proceed to the next iteration to guess another candidate assumption set A. The counterexamples obtained are used for further restricting (or "refining") the set of candidate assumption sets that will be considered in the forthcoming iterations, depending on the semantics at hand.

We will continue by describing the approach in more detail, including how the candidate assumption sets are guessed, conclusions determined, counterexamples obtained, and refinements made. As convenient, we will treat a set of propositional formulas interchangeably as the conjunction of the formulas the set contains.

The following observation is central to our approach. For a given ABF $F = (W, D)$, we define a function that, given a subset $S \subseteq \mathscr{A}$ of assumptions, iteratively constructs the set of conclusions of defaults in D that are applicable by S. Let $derived_S(X) = \{c \mid (c \leftarrow a, Mb_1, \ldots, Mb_n) \in D \text{ where } Mb_1, \ldots, Mb_n \in S \text{ and } W \cup \{\bigwedge_{c' \in X} c'\} \models a\}$. For a given S, it holds that $derived_S$ is $\subseteq$-monotone; this follows by monotonicity of classical propositional entailment $\models$. Further, a unique least fixed-point exists and can be computed by iteratively applying $derived_S$ starting with $X = \emptyset$, i.e., $derived_S^i(\emptyset)$ for some $i \geq 0$ reaches the least fixed point. This fact can be directly inferred since D is assumed to be finite: $derived_S^i(\emptyset)$ reaches some fixed-point after some number of iterations, and the result of each iteration must be a subset of each fixed-point of $derived_S$ (e.g., it holds that $derived_S(\emptyset)$ is part of each fixed-point, and then also $derived_S^1(\emptyset)$ must be, and so on).

Proposition 1. *Suppose an ABA framework $F = (\mathscr{L}, \mathscr{R}, W, \mathscr{A}, ^{-})$, a set of assumptions $A \subseteq \mathscr{A}$ and a sentence $x \in \mathscr{L}$. Let X be the least fixed point of $derived_A$ and $C = \bigwedge_{c \in X} c$. It holds that x is derivable from A in F if and only if either $x \in A$ or the propositional formula $W \wedge C \wedge \neg x$ is not satisfiable.*

Proof. We first show that $y \in X$ implies that y is derivable from A in F via induction on $derived_A^i(\emptyset)$ with $i \geq 1$. For the base case $i = 1$, since $y \in derived_A(\emptyset)$, there is a rule $y \leftarrow a, Mb_1, \ldots, Mb_n$ with $\{Mb_1, \ldots, Mb_n\} \subseteq A$ and $W \models a$. Thus there is a derivation of a from W in F. For the induction step, assume that $y \in derived_A^i(\emptyset)$ implies y is derivable from A in F. We need to show that $z \in derived_A^{i+1}(\emptyset)$ implies z is derivable from A in F. Then there is, again, a rule $z \leftarrow a, Mb_1, \ldots, Mb_n$ with $\{Mb_1, \ldots, Mb_n\} \subseteq A$. It must hold that a is entailed by $W \wedge \bigwedge_{c \in derived_A^i(\emptyset)} c$, implying that there is a derivation in F for a from A and, in turn, also for z. Thus, $y \in X$ implies y is derivable from A in F. Now assume that $W \wedge C \wedge \neg x$ is not satisfiable, and thus $W \wedge C \models x$. Each conjunct c of C is a subset of X and thus, as shown, derivable from A. Therefore x is derivable from A in F.

Suppose x is derivable from A in F. Then there is a sequence of rules in $\mathscr{R}$ s.t. each body element of each rule is either in A or the head of a rule previously in the sequence. We can assume that the derivation is of finite length, since there are only finitely many default rules, and a derivation in a sound and complete inference system in propositional logic can be assumed to be only requiring finitely many steps. Consider again an inductive line of reasoning on the length of the derivation. The base case is straightforward: the body of the first rule is in A or W, and, in turn, the head is entailed by $W \wedge C$. Assume that up to i each head of preceding rules is entailed by $W \wedge C$. Independently of the rule being a default, the body of the rule is either entailed by $W \wedge C$ (from previous rules and induction hypothesis) or in A, implying the claim. □

We continue by detailing our algorithmic approach. Apart from W, central to the approach is a propositional formula ϕ over all of the assumptions Ma of the framework in question, used to guess candidate assumption sets. Initially ϕ is the empty formula, and ϕ is expanded at each refinement step conjunctively with further propositional clauses. Following Proposition 1, W is used to decide derivations and thus attacks from any given assumption set. In the following, for a truth assignment I we let $I_{\mathscr{A}} = \{Ma \mid Ma \in \mathscr{A} \cap I\}$ to denote the set of assumptions that are assigned to *true* by I.

Algorithm 1 gives the skeleton for credulous reasoning under stable, admissible and complete semantics. Recall that ϕ is a conjunction of *refinement clauses* ruling out provably incorrect candidate solutions (initially the empty formula). A candidate assumption

Algorithm 1 Credulous reasoning: skeleton for stable, admissible and complete

```
Require: ABA framework $(W, D)$ and a query $q \in \mathscr{L}_p$
Ensure: return YES if $q$ is credulously justified under given semantics, NO otherwise
Let $\phi$ be an empty propositional formula over $\mathscr{A}$
while $I \leftarrow$ Sat($\phi$) do
    $C \leftarrow$ Concluded_via_Defaults($I_{\mathscr{A}}$)
    if CF_Derive_Query($I_{\mathscr{A}}, C, q$) then
        if $\neg$Counterexample($I, C$) then return YES
        $\phi \leftarrow$ Refine($I$)
return NO
```

Algorithm 2 Concluded_via_Defaults(A)

```
Require: $A \subseteq \mathscr{A}$
Ensure: return the conclusions of the default rules that are applicable by $A$.
 1: $X \leftarrow \{r \in X \mid M(r) \subseteq A\}$
 2: $C \leftarrow \top$
 3: $changes \leftarrow$ true
 4: while $changes$ do
 5:    $changes \leftarrow$ false
 6:    $X' \leftarrow X$
 7:    while $X' \neq \emptyset$ do
 8:       $testrule \leftarrow pop(X')$
 9:       if $I \leftarrow \text{Sat}(W \wedge C \wedge \neg prereq(testrule))$ then $X' \leftarrow X' \setminus \{r \in X \mid \neg prereq(r) \in I\}$
10:       else
11:          $C \leftarrow C \wedge conc(testrule)$
12:          Remove $testrule$ from $X$
13:          $changes \leftarrow$ true
14:          break
15: return C
```

set $I_{\mathscr{A}}$ is guessed by constraining the solution space with the refinement clauses. As explained further in what follows, the following subroutines are used in the algorithms. Here C is the least fixed point of $derived_A$ for assumption set A (Algorithm 2).

- notDerivable(C,q): invokes a SAT solver on $W \wedge C \wedge (\neg q)$, true iff q is not derivable from A.
- ExistUndefeated(B,C): invokes a SAT solver on $W \wedge C \wedge \bigvee_{Ma \in B} a$, true iff there is an assumption in B not attacked by A.
- Attacked(B,C): returns the set of assumptions in B attacked by A (Algorithm 4).

On Line 3 of Algorithm 1, the rules applicable by $I_{\mathscr{A}}$ are determined, and the conclusions of those rules are conjoined as C. Line 4 checks if $I_{\mathscr{A}}$ is conflict-free and the query is derived from $I_{\mathscr{A}}$; both checks use W with every element of C enforced to hold. If those checks succeed, the existence of a counterexample is checked on Line 5. The form of the counterexample depends on the semantics as detailed later on. If a counterexample is not found, $I_{\mathscr{A}}$ is a σ-assumption set that derives q. Otherwise a refinement is added, again depending on semantics; see below.

The details of the subroutines Concluded_via_Defaults, CF_Derive_Query, and Attacked are provided as Algorithms 2, 3 and 4, respectively. Given an assumption set A, Algorithm 2 computes the least fixed-point of $derived_A$ (recall Proposition 1). The rules whose justifications is a subset of A are collected to X (Line 1). C is a conjunction over

Algorithm 3 CF_Derive_Query(I,C,q)

```
Ensure: return YES if $I_{\mathscr{A}}$ is conflict-free and derives $q$
if Attacked$(I_{\mathscr{A}}, C) \neq \emptyset$ or notDerivable$(C,q)$ then
   $\phi \leftarrow$ Refine$(I)$
   return NO
return YES
```

Algorithm 4 Attacked(B,C)

```
Ensure: return the elements of $B$ that are attacked by $A$
1: $X \leftarrow \{x \mid Mx \in B\}$
2: while $X$ not empty do
3:     if $I \leftarrow \mathrm{Sat}(W \wedge C \wedge \bigvee_{x \in X} x)$ then $X \leftarrow X \setminus I$ else return $\{Mx \mid x \in X\}$
4: return $\{Mx \mid x \in X\}$
```

the set of conclusions of rules in X that are in the deductive closure; initially C is empty (Line 2). In the loop starting on Line 4 it is iteratively checked for rules in X if their prerequisite is entailed by $W \wedge C$. If not, all rules whose prerequisite occurs negatively in the found assignment I are removed from consideration for this loop, because none of those prerequisites are entailed (Line 9). If a rule is entailed, the conclusion of this rule is added to C (Line 11). In this case the rule is removed from further consideration as it is already determined to be applicable (Line 12). This rule being applicable changes what $W \wedge C$ entails, and thus the algorithm will not terminate (Line 13). Instead, each rule not already found to be applicable is considered in the next iteration. This is continued until no more prerequisites of rules in X are entailed.

Algorithm 3 checks if the candidate is conflict-free and derives the query. Concretely it returns NO and applies the appropriate refinements if there is an attack from the candidate to itself, checked with Algorithm 4, or the query is not entailed by $W \wedge C$. Given assumption sets A and B, and the conclusions of default rules applicable by A, Algorithm 4 considers the contents of assumptions in B (Line 1), and iteratively removes from consideration those contents of B whose negation $W \wedge C$ does not entail, as this implies that A does not derive them. The rest of the assumptions are attacked by A. If the call on Line 3 is unsatisfiable, then the set of attacked assumptions returned on Line 3 is not empty, otherwise the empty set is returned on Line 4.

3.1. Counterexamples

Counterexample(I,C) is defined as follows for the individual semantics. These subroutines determine if a conflict-free assumption set A is a σ-assumption set under stable, admissible, or complete semantics, respectively. They employ W and C (the least fixed point of $derived_A$) as computed in Algorithm 2.

- Stable: return ExistUndefeated$(\mathscr{A} \setminus I_{\mathscr{A}}, C)$. That is, it is checked whether there is an assumption that is not in the current candidate A and not defeated by it.
- Admissibility: return true if Attacked$(I_{\mathscr{A}}, C') \neq \emptyset$, where $U = \mathscr{A} \setminus$ Attacked$(\mathscr{A} \setminus I_{\mathscr{A}}, C)$, and $C' =$ Concluded_via_Defaults(U). That is, collect all assumptions U not defeated by A, and then check that U does not defeat A.
- Complete: with U and C' as above, return true if either Attacked$(I_{\mathscr{A}}, C') \neq \emptyset$ or ExistUndefeated$(\mathscr{A} \setminus I_{\mathscr{A}}, C')$ is true. That is, A is not complete if it is not admissible or there is an assumption outside A that is not attacked by the assumptions that A does not attack.

3.2. Refinements

The details of the refinement step made when a counterexample is found depend on the semantics at hand. In particular, we make use of the observations detailed in Proposition 2 to obtain in cases stronger refinements—ruling out more than one candidate from further consideration at forthcoming iterations of Algorithm 1 than what is be achieved by the so-called trivial refinement, consisting of ruling out only the particular candidate for which a counterexample is obtained from further consideration.

Proposition 2. *Given an ABA framework $F = (\mathscr{L}, \mathscr{R}, W, \mathscr{A}, \bar{\ })$ and an assumption set $A \subseteq \mathscr{A}$, it holds that*

1. *if A is not conflict-free, then no $A' \supseteq A$ is conflict-free,*
2. *if A is conflict-free but not stable, then no $A' \subseteq A$ is stable,*
3. *if A does not derive x, then no $A' \subseteq A$ derives x, and an A' that derives x must cover the justifications of at least one default rule not covered by A,*
4. *if A derives x, then all $A' \supseteq A$ derive x, and*
5. *if A is conflict-free but not admissible, then no superset of the assumptions not defeated by A is admissible.*

Proof. Item 1: the same self-attack is present in $A' \supseteq A$ as in A. For Item 2, suppose that A is conflict-free but not stable, implying that there is an $a \in \mathscr{A} \setminus A$ that A does not attack. Derivability is $\subseteq$-monotone, implying that a is not attacked by any subset of A either. The first part of Item 3 follows from the same observation for an A that does not derive x. The second part follows from the fact that to increase what A derives, one needs to apply additional default rules. Item 4: the same derivation for x exists from any $A' \supseteq A$ than from A. Lastly, for Item 5 assume that A is conflict-free but not admissible, implying that there is an attack from the set U of assumptions not defeated by A to A. Since A is conflict-free, $A \subseteq U$. Thus U is self-defeating and the claim follows from Item 1. □

Refine(I) is defined as follows for the different cases of a candidate being discarded, based on Proposition 2. Here U is the set of assumptions not defeated by $I_{\mathscr{A}}$.

Refinement	Required	Optional
A is not conflict-free	$\bigvee_{Ma \in I_{\mathscr{A}}} \neg Ma$	
Query not deriv. from A (cred.)	$\bigvee_{Ma \in \mathscr{A} \setminus I_{\mathscr{A}}} Ma$	$\bigvee_{r \in D,\ M(r) \not\subseteq I} \bigwedge_{Mb_i \in M(r)} Mb_i$
Query deriv. from A (skept.)	$\bigvee_{Ma \in I_{\mathscr{A}}} \neg Ma$	
A is not stable	$\bigvee_{Ma \in \mathscr{A} \setminus I_{\mathscr{A}}} Ma$	
A is not admissible	$\bigvee_{Ma \in \mathscr{A} \setminus I_{\mathscr{A}}} Ma \vee \bigvee_{Ma \in I_{\mathscr{A}}} \neg Ma$	$\bigvee_{Ma \in U} \neg Ma$
A is not complete	$\bigvee_{Ma \in \mathscr{A} \setminus I_{\mathscr{A}}} Ma \vee \bigvee_{Ma \in I_{\mathscr{A}}} \neg Ma$	

The required clauses for admissible and complete simply rule out the current candidate. We also list optional clauses for a query not being derived from the candidate for credulous reasoning and the candidate not being admissible. The optional clauses are not neceessary for correctness, but are potentially useful for restricting the remaining candidate space further towards earlier termination.

Algorithm 5 Acceptance under grounded

```
Require: ABA framework $F = (W, D)$ and query $q \in \mathscr{L}_p$
Ensure: return YES if $q$ is credulously justified in $F$ under grounded, NO otherwise
1: $S \leftarrow \emptyset$
2: while true do
3:     $C \leftarrow$ Concluded_via_Defaults$(S)$
4:     $U \leftarrow \mathscr{A} \setminus$ Attacked$(\mathscr{A}, C)$
5:     $C' \leftarrow$ Concluded_via_Defaults$(U)$
6:     $defended \leftarrow \mathscr{A} \setminus$ Attacked$(\mathscr{A} \setminus S, C')$
7:     if $defended$ is empty then break else $S \leftarrow S \cup defended$
8: return $\neg$notDerivable$(C, q)$
```

3.3. The Special Case of Grounded Semantics

Algorithm 5 for grounded semantics differs from the other semantics. Here we rely on the fact that DL-ABA frameworks are flat and that the grounded assumption set is the least fixed point of $\mathscr{F}_F(S) = \{Mb \in \mathscr{A} \mid \{Mb\}$ defended by $S\}$ for an $S \subseteq \mathscr{A}$. In other words, the grounded assumption set can be iteratively build in the set S, initially empty, by iteratively adding to it all assumptions defended by it. More concretely, on Line 3 the conclusions of rules applicable by A are identified, and on Line 4 the set of assumptions U not defeated by S is identified. Similarly, on Lines 5 and 6 the conclusions of rules applicable by U and assumptions defended by S are identified; any assumption that is not attacked by U is defended by S, because S counters all attacks not originating from U. If S defends no assumptions outside of itself, S is the least fixed point of the defense-operator, and thus the grounded assumption set (Line 7). Finally, q is accepted under grounded semantics if and only if it is derivable from S (Line 8).

3.4. Skeptical Reasoning and Enumeration

Algorithm 1 requires only minor modifications for deciding skeptical acceptance by checking for the existence of a counterexample instead of set deriving the query. In particular, negate the entailment check in Algorithm 3 and invert the answer given by Algorithm 1. Then the subroutine returns NO if the query is entailed by $W \wedge C$, since then the candidate derives the query and thus can not constitute a counterexample. To enumerate all σ-assumption-sets, remove the entailment check of Algorithm 3 and when a solution is found, instead of terminating collect the solution and continue by calling Refine(I).

To query for an assumption Mb instead of member of $\mathscr{L}_p$, it suffices to initialize ϕ with the assertion that Mb holds and omit the entailment check in Algorithm 3.

As there is guaranteed to be a unique grounded assumption set in a flat ABA framework (including any DL-ABA framework), Algorithm 5 answers skeptical acceptance as such. To enumerate (more precisely, return the single) grounded assumption set, S should be returned on Line 8 instead of performing the entailment check.

4. Experiments

We provide first empirical results on the runtime performance of our SAT-based approach to DL-ABA, focusing on credulous and skeptical reasoning. We implemented the algorithms (available at https://bitbucket.org/coreo-group/satfordl-aba) using the PySAT Python interface [23] and Glucose 3 [24] as the SAT solver. We were unable to run earlier systems [15,16,17,18] for finding extensions in DL (specifically for finding stable extensions in DL-ABA) for a comparison due to unavailability and compilation issues. The experiments were run on 2.60-GHz Intel Xeon E5-2670 8-core 64-GB machines with CentOS 7 under a per-instance 600-s time and 16-GB memory limit.

For benchmarks, we randomly sampled a set of 400 CNF formulas from a large set of real-world SAT instances originally used for benchmarking iterative SAT-based algorithms for the beyond-NP problem of backbone computation [25]. These CNF formulas can be considered suitably challenging for the purpose of our evaluation. For each CNF formula, we chose one literal uniformly at random from the set of positive and negative instances of all variables occuring in the formula, and added default rules to obtain ABFs. Specifically, for each CNF formula, we generated 12 ABFs for a total of 4800 instances, with $a \in \{20, 50, 100, 200\}$ assumptions, each of whose content was randomly selected from the literals in the CNF, and $d \in \{100, 200, 400\}$ default rules per framework. Each rule had one literal selected uniformly at random as its prerequisite and conclusion, respectively, and from one to five assumptions as its justifications, with both the amount and identities of the assumptions selected uniformly at random.

Table 1 gives the number of timeouts and mean runtimes over solved instances wrt. the number of assumptions and rules for the different reasoning tasks for credulous reasoning under stable, admissible, complete and grounded semantics, and for skeptical reasoning under stable semantics.[2] Note that these results are obtained without using the optional refinements for credulous reasoning and admissible semantics (recall Section 3.2). Acceptance under grounded semantics appears the easiest to solve, likely due to the fact that in this algorithm, unlike for the other semantics, we do not need to "guess" assumption sets, but instead build the grounded set iteratively. Acceptance under stable semantics appears harder to solve compared to the other second-level problems, which fits the intuition that there are fewer stable sets, making it harder to find a suitable one. Performance on skeptical acceptance under stable semantics is very similar to credulous acceptance, and likewise for performance under admissible and complete semantics for credulous reasoning. For the semantics other than grounded, there is a clear increase in difficulty of the instances as the number of assumptions and defeasible rules, respectively, increases. For example, for all parameters combinations with at least 100 assumptions, our current implementation timed out on over half of the instances, suggesting that there would be room for further runtime improvements as well as for developing further understanding on what makes individual instances hard to solve.

Figure 1 shows the impact of the number of assumptions on runtime performance and (in the left side plot) the effect of the optional credulous reasoning refinement under stable semantics. Interestingly, the optional refinement clauses appear to degrade runtime performance, especially for instances with 50 assumptions. For a possible explanation, note that while the number of candidates that do not derive the given query is much lower when using the optional refinement, there are in turn more candidates that fail the other criteria (conflict-freeness and stability). Hence the number of iterations may even increase when using the optional refinements. On the other hand, the optional refinement

[2] Skeptical reasoning under admissible semantics is relatively easy as it reduces to checking if the empty assumption set derives the query. Skeptical reasoning under complete semantics coincides with acceptance under grounded semantics. Credulous reasoning under admissible and complete semantics also coincide.

Table 1. Detailed runtime results. There are a total of 400 instances per each combination of $|\mathscr{A}|$ and $|D|$.

		#timeouts (mean runtime over solved instances (s))									
$\lvert\mathscr{A}\rvert$	$\lvert D\rvert$	*stb* cred		*adm* cred		*com* cred		*grd* accept		*stb* skept	
20	100	20	(55.5)	29	(50.8)	28	(52.6)	9	(11.0)	22	(54.6)
	200	65	(67.9)	66	(55.1)	65	(59.4)	12	(15.7)	66	(62.2)
	400	97	(77.9)	88	(66.3)	92	(60.2)	10	(15.9)	96	(71.3)
50	100	115	(113.0)	108	(83.5)	108	(84.1)	13	(14.7)	118	(117.3)
	200	200	(107.8)	180	(57.2)	180	(58.6)	8	(17.0)	197	(111.4)
	400	264	(48.8)	223	(26.9)	225	(21.6)	13	(19.5)	263	(56.5)
100	100	240	(101.6)	201	(51.7)	201	(55.4)	12	(16.4)	238	(108.8)
	200	285	(61.2)	226	(27.5)	228	(24.4)	14	(18.8)	281	(61.9)
	400	299	(32.3)	235	(15.3)	235	(16.9)	12	(24.1)	301	(32.2)
200	100	293	(45.3)	229	(25.4)	231	(24.2)	13	(21.5)	297	(37.6)
	200	312	(8.4)	239	(7.0)	239	(7.9)	16	(20.7)	310	(24.8)
	400	312	(13.5)	239	(11.0)	240	(12.9)	16	(20.9)	316	(16.7)

clauses for admissibility seem to slightly improve performance on admissible and complete semantics, allowing for solving 1 and 3 more instances, respectively. Figure 1(right) shows that instances with more default rules are generally harder to solve under complete and stable semantics (we omit here admissible due to similar performance to complete and grounded due to the relative easiness of the task).

5. Conclusions

We developed an algorithmic approach, based on iterative applications of Boolean satisfiability (SAT) solvers, to reasoning in ABA instantiated with propositional default

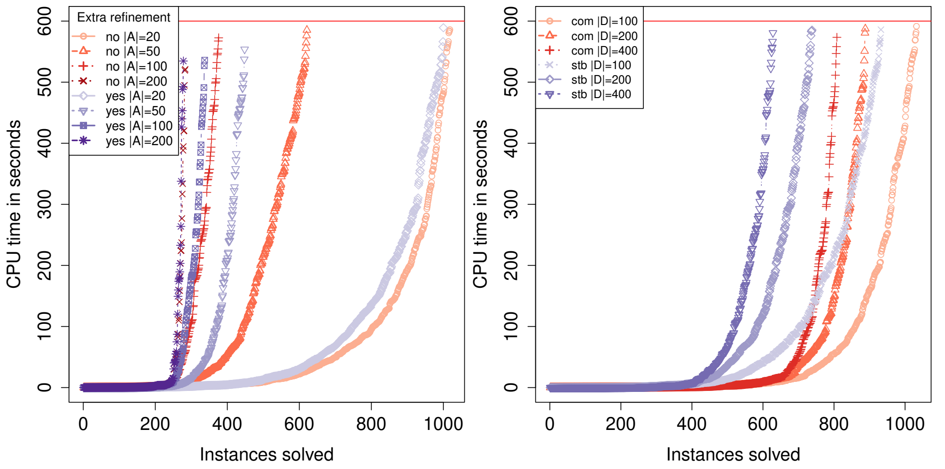


Figure 1. Left: credulous reasoning under stable semantics with and without the optional refinement clauses. Right: credulous reasoning under complete and stable semantics. 1200 instances for each variant.

logic (DL). We detailed instantiations of the approach for deciding acceptance and for assumption-set enumeration in the DL-instantiation of ABA under several central argumentation semantics. We also provided an implementation of the approach which to the best of our understanding is the first system in its generality targeted at the DL instantiation of ABA, and empirically evaluated its potential.

References

[1] Bondarenko A, Dung PM, Kowalski RA, Toni F. An Abstract, Argumentation-Theoretic Approach to Default Reasoning. Artif Intell. 1997;93:63-101.

[2] Čyras K, Fan X, Schulz C, Toni F. Assumption-Based Argumentation: Disputes, Explanations, Preferences. In: Handbook of Formal Argumentation. College Publications; 2018. p. 365-408.

[3] Modgil S, Prakken H. A general account of argumentation with preferences. Artif Intell. 2013;195:361-97.

[4] Besnard P, Hunter A. Elements of Argumentation. MIT Press; 2008.

[5] García AJ, Simari GR. Defeasible Logic Programming: An Argumentative Approach. Theory Pract Log Program. 2004;4(1-2):95-138.

[6] Gaertner D, Toni F. CaSAPI: A system for credulous and sceptical argumentation. In: Proc. NMR; 2007. p. 80-95.

[7] Thimm M. Tweety: A Comprehensive Collection of Java Libraries for Logical Aspects of Artificial Intelligence and Knowledge Representation. In: Proc. KR. AAAI Press; 2014. p. 528-37.

[8] Craven R, Toni F. Argument graphs and assumption-based argumentation. Artif Intell. 2016;233:1-59.

[9] Lehtonen T, Wallner JP, Järvisalo M. Declarative Algorithms and Complexity Results for Assumption-Based Argumentation. J Artif Intell Res. 2021;71:265-318.

[10] Lehtonen T, Wallner JP, Järvisalo M. Harnessing Incremental Answer Set Solving for Reasoning in Assumption-Based Argumentation. Theory Pract Log Program. 2021;21(6):717-34.

[11] Dimopoulos Y, Nebel B, Toni F. On the computational complexity of assumption-based argumentation for default reasoning. Artif Intell. 2002;141(1/2):57-78.

[12] Reiter R. A Logic for Default Reasoning. Artif Intell. 1980;13(1-2):81-132.

[13] Gottlob G. Complexity Results for Nonmonotonic Logics. J Log Comput. 1992;2(3):397-425.

[14] Biere A, Heule M, van Maaren H, Walsh T, editors. Handbook of Satisfiability, 2nd edition. vol. 336 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2021.

[15] Schaub T, Brüning S. Prolog Technology for Default Reasoning: Proof Theory and Compilation Techniques. Artif Intell. 1998;106(1):1-75.

[16] Cholewinski P, Marek VW, Truszczynski M. Default Reasoning System DeReS. In: Proc. KR. Morgan Kaufmann; 1996. p. 518-28.

[17] Nicolas P, Saubion F, Stéphan I. GADEL: a Genetic Algorithm to Compute Default Logic Extensions. In: Proc. ECAI. IOS Press; 2000. p. 484-90.

[18] Chen Y, Wan H, Zhang Y, Zhou Y. dl2asp: Implementing Default Logic via Answer Set Programming. In: Proc. JELIA. vol. 6341 of LNCS. Springer; 2010. p. 104-16.

[19] Cyras K, Heinrich Q, Toni F. Computational complexity of flat and generic Assumption-Based Argumentation, with and without probabilities. Artif Intell. 2021;293:103449.

[20] Clarke EM, Gupta A, Strichman O. SAT-based counterexample-guided abstraction refinement. IEEE Trans Comput Aided Des Integr Circuits Syst. 2004;23(7):1113-23.

[21] Dvořák W, Järvisalo M, Wallner JP, Woltran S. Complexity-sensitive decision procedures for abstract argumentation. Artif Intell. 2014;206:53-78.

[22] Wallner JP, Niskanen A, Järvisalo M. Complexity Results and Algorithms for Extension Enforcement in Abstract Argumentation. J Artif Intell Res. 2017;60:1-40.

[23] Ignatiev A, Morgado A, Marques-Silva J. PySAT: A Python Toolkit for Prototyping with SAT Oracles. In: Proc. SAT. vol. 10929 of LNCS. Springer; 2018. p. 428-37.

[24] Audemard G, Simon L. On the Glucose SAT Solver. Int J Artif Intell Tools. 2018;27(1):1840001:1-1840001:25.

[25] Janota M, Lynce I, Marques-Silva J. Algorithms for computing backbones of propositional formulae. AI Commun. 2015;28(2):161-77.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220157

Value-Based Practical Reasoning: Modal Logic + Argumentation

Jieting Luo [a,1] Beishui Liao [b] and Dov Gabbay [c]

[a] *University of Bern, Swtzerland*
[b] *Zhejiang University, China*
[c] *King's College London, UK*

Abstract. Autonomous agents are supposed to be able to finish tasks or achieve goals that are assigned by their users through performing a sequence of actions. Since there might exist multiple plans that an agent can follow and each plan might promote or demote different values along each action, the agent should be able to resolve the conflicts between them and evaluate which plan he should follow. In this paper, we develop a logic-based framework that combines modal logic and argumentation for value-based practical reasoning with plans. Modal logic is used as a technique to represent and verify whether a plan with its local properties of value promotion or demotion can be followed to achieve an agent's goal. We then propose an argumentation-based approach that allows an agent to reason about his plans in the form of supporting or objecting to a plan using the verification results.

Keywords. Value, Practical Reasoning, Modal Logic, Argumentation

1. Introduction

Autonomous agents are supposed to be able to perform value-based ethical reasoning based on their value systems in order to distinguish moral from immoral behavior. Existing work on value-based practical reasoning such as [1][2] [3] demonstrates how an agent can reason about what he should do among alternative action options that are associated with value promotion or demotion. However, agents are supposed to be able to finish tasks or achieve goals that are assigned by their users through performing a sequence of actions. Since there might exist multiple plans that an agent can follow and each plan might promote or demote different values along each action, the agent should be able to resolve the conflicts between them and evaluate which plan he should follow. If the decision-making problem concerns choosing a plan instead of an action, then we first need to know how an agent can see whether he can follow a plan to achieve his goal. Verification approaches that are developed based on modal logic only allow us to verify whether a goal can be achieved under specific conditions such as norm compliance assumptions [4][5][6], namely telling us whether a plan works or not, but cannot tell us what we should do. For sure, we can collect the verification results regarding whether a plan promotes or demotes a specific set of values and then compare different plans using

[1] Corresponding Author.

lifting approaches as what has been done in [7]. However, the order lifting problem is a major challenge in many areas of AI and no approach is ultimately "correct". Moreover, the agent in our setting needs to lift the preference over values to the preference over plans with respect to value promotion and demotion, which even complicates the problem. Therefore, we need a more natural and intuitive approach. It has been shown that argumentation provides a useful mechanism to model and resolve conflicts [8], and particularly can be used for the decision-making of artificial intelligence and provides explanation for that [9][10]. In this paper, we develop a logic-based framework that combines modal logic and argumentation for value-based practical reasoning with plans. Modal logic is used as a technique to represent and verify whether a plan with its local properties of value promotion or demotion can be followed to achieve an agent's goal. Using the verification results to construct arguments, we then propose an argumentation-based approach that allows an agent to reason about his plans in the form of support and objection without using lifting approaches. We prove several formal properties to characterize our approach, indicating it is consistent with our rationality of decision-making.

2. Logical Framework

The semantic structure of this paper is a transition system that represents the computational behavior of a system caused by an agent's actions in the agent's subjective view. It is basically a directed graph where a set of vertexes S corresponds to possible states of the system, and the relation $\rightarrow \subseteq S \times Act \times S$ represents the possible transitions of the system. When a certain action $\alpha \in Act$ is performed, the system might progress from a state s to a different state s' in which different propositions hold. Formally,

Definition 1 (Transition Systems). *Let $\Phi = \{p, q, ...\}$ be a finite set of atomic propositional variables, a transition system is a tuple $T = (S, Act, \rightarrow, V)$ over Φ, where*

- *S is a finite, non-empty set of states;*
- *Act is a finite, non-empty set of actions;*
- *$\rightarrow \subseteq S \times Act \times S$ is a transition relation between states with actions, which we refer to as the transition relation labeled with an action; we require that for all $s \in S$ there exists an action $a \in Act$ and a state $s' \in S$ such that $(s, a, s') \in \rightarrow$; we restrict actions to be deterministic, that is, if $(s, a, s') \in \rightarrow$ and $(s, a, s'') \in \rightarrow$, then $s' = s''$; since the relation is partially functional, we write $s[\alpha]$ to denote the state s' for which it holds that $(s, \alpha, s') \in \rightarrow$; we also use $s[\alpha_1, \ldots, \alpha_n]$ to denote the resulting state for which a sequence of actions $\alpha_1, \ldots, \alpha_n$ succinctly execute from state s;*
- *V is a propositional valuation $V : S \rightarrow 2^{\Phi}$ that assigns each state with a subset of propositions which are true at state s; thus for each $s \in S$ we have $V(s) \subseteq \Phi$.*

Note that the model is deterministic: the same action performed in the same state will always result in the same resulting state. A pointed transition system is a pair (T, s) such that T is a transition system, and $s \in S$ is a state from T. Adopted from [11][12], the language $\mathcal{L}$ is propositional logic extended with action modality. Formally, its grammar is defined below:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid Do(\alpha)\varphi \quad (p \in \Phi, \alpha \in Act)$$

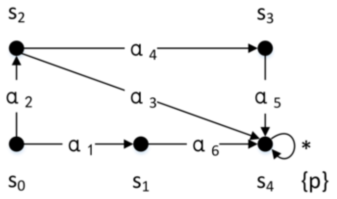


Figure 1. Transition system T.

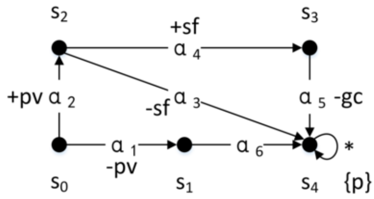


Figure 2. A Value-based transition system VT.

Given a pointed transition system (T,s), we define the semantics with respect to the satisfaction relation $\vDash$ inductively as follows:

- $T,s \vDash p$ iff $p \in V(s)$;
- $T,s \vDash \neg\varphi$ iff $T,s \nvDash \varphi$;
- $T,s \vDash \varphi \vee \psi$ iff $T,s \vDash \varphi$ or $T,s \vDash \psi$;
- $T,s \vDash Do(\alpha)\varphi$ iff $s[\alpha] \vDash \varphi$.

The remaining classical logic connectives are assumed to be defined as abbreviations in terms of $\neg$ and $\vee$ in the conventional manner. Given a pointed transition system (T,s), we say that a sequence of actions $\alpha_1 \dots \alpha_n$ brings about a φ-state if and only if $T,s \vDash Do(\alpha_1)\dots Do(\alpha_n)\varphi$. As standard, we write $T \vDash \varphi$ if $T,s \vDash \varphi$ for all $s \in S$, and $\vDash \varphi$ if $T \vDash \varphi$ for all T.

A transition system represents how a system progresses by an agent's actions. Besides, an agent in the system is assumed to have his own goal, which is a formula expressed in propositional logic $\mathcal{L}_{prop}$. It is indeed possible for an agent to have multiple goals and his preference over different goals. For example, a goal hierarchy is defined in [4] to represent increasingly desired properties that the agent wishes to hold. However, we find that the setting about whether the agent has a goal or multiple goals is in fact not essential for our analysis, so we simply assume that the agent only has a goal for simplifying our presentation.

Example 1. *Consider the transition system T in Figure 1, which represents how an agent can get to a pharmacy to buy medicine for his user. State s_0 is the initial state, representing staying at home, and proposition p, representing arriving at a pharmacy, holds in state s_4. The agent can perform actions α_1 to α_6 in order to get to state s_4. From this transition system, the following formulas hold:*

$$T,s_0 \vDash Do(\alpha_1)Do(\alpha_6)p,$$

$$T,s_0 \vDash Do(\alpha_2)Do(\alpha_3)p,$$

$$T,s_0 \vDash Do(\alpha_2)Do(\alpha_4)Do(\alpha_5)p,$$

which means that the agent can first perform action α_1 and then action α_6, or action α_2 followed by action α_4, or action α_2 followed by actions α_4 and α_5, to get to the pharmacy.

It is important for an agent not only to achieve his goal, but also to think about how to achieve his goal. As we can see from the running example, there are multiple ways for the agent to get to the pharmacy, and the agent needs to evaluate which one is the best to choose. In this paper, agents are able to perform value-based practical reasoning in terms of planning their actions to achieve their goals. We first assume that an agent has a set of values. A value can be seen as an abstract standard according to which agents have their preferences over options. For instance, if we have a value denoting *equality*, we prefer the options where equal sharing or equal rewarding hold. Unlike [7] where a value is interpreted as a state formula, we simply assume a value as a primitive structure without considering how it is defined. We assume that agents can always compare any two values, so we define an agent's value system as a total pre-order (instead of a strict total order) over a set of values, representing the degree of importance of something.

Definition 2 (Value System)**.** *A value system* $V = (\mathrm{Val}, \precsim)$ *is a tuple consisting of a finite set of values* $\mathrm{Val} = \{v, ..., v'\}$ *together with a total pre-ordering* $\precsim$ *over* Val. *When* $v \precsim v'$*, we say that value* v' *is at least as important as value* v*. As is standard, we define* $v \sim v'$ *to mean* $v \precsim v'$ *and* $v' \precsim v$*, and* $v \prec v'$ *to mean* $v \precsim v'$ *and* $v \not\sim v'$*.*

We label some of the transitions with the values promoted and demoted by moving from a starting state to a ending state. Notice that not every transition can be labeled, as some transitions may not be relevant to any value in an agent's value system. Formally, function $\delta : \{+,-\} \times \mathrm{Val} \to 2^{\to}$ is a valuation function which defines the status (promoted (+) or demoted (-)) of a value $v \in \mathrm{Val}$ ascribed to a set of transitions. We then define a value-based transition system VT as a transition system together with a value system V and a function δ.

Definition 3 (Value-based Transition Systems)**.** *A value-based transition system is defined by a triple* $VT = (T, V, \delta)$*, where* T *is a transition system,* V *is a value system and* δ *is a valuation function that assigns value promotion or demotion to a set of transitions.*

Given a sequence of actions with respect to a value-based transition system, we then express whether the performance of the sequence in a state promotes or demotes a specific value, which can be done by extending our language. Given a pointed value-based transition system (VT, s) and a value $v \in \mathrm{Val}$, the satisfaction relation $VT, s \vDash \psi$ is extended with the following new semantics:

- $VT, s \vDash_{+v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ iff $s[\alpha_1, \ldots, \alpha_n] \vDash \varphi$ and there exists $1 \leq m \leq n$ such that $(s[\alpha_1, \ldots, \alpha_{m-1}], \alpha_m, s[\alpha_1, \ldots, \alpha_m]) \in \delta(+, v)$;
- $VT, s \vDash_{-v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ iff $s[\alpha_1, \ldots, \alpha_n] \vDash \varphi$ and there exists $1 \leq m \leq n$ such that $(s[\alpha_1, \ldots, \alpha_{m-1}], \alpha_m, s[\alpha_1, \ldots, \alpha_m]) \in \delta(-, v)$.

The formula $VT, s \vDash_{+v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ (resp. $VT, s \vDash_{-v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$) should be intuitively read as φ is achieved after the performance of a sequence of actions $\alpha_1, \ldots, \alpha_n$ in state s and there exists an action that promotes (resp. demotes) value v in the sequence. Notice that the formula only expresses the local property of a sequence of actions in terms of value promotion or demotion by an action within the sequence. Thus, it is possible that an action within the sequence promotes value v but it gets demoted by another action within the sequence, meaning that both $VT, s \vDash_{+v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ and $VT, s \vDash_{-v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ hold at the same time. Through checking the above formulas, the agent is then aware of whether he can perform the sequence of actions to achieve his goal and which value gets promoted or demoted along the sequence. We continue our running example to illustrate how to use our logical language to express and verify properties of sequences of actions.

Example 2. *Suppose the ethical agent has privacy (pv), safety (sf) and good conditions (gc) as his values and a value system as $pv < gc < sf$. As in Figure 2, some of the transitions have been labeled with value promotion or demotion with respect to the agent's values. Taking action α_1 in state s_0 is interpreted as going through the neighbor's garden for taking shortcut, which demotes the value of privacy of the neighbor, conversely taking action α_2 in state s_0 is interpreted as stepping on a normal way, which promotes the value of privacy of the neighbor. Taking action α_3 means crossing the road without using the crosswalk, which demotes the value of safety of the agent, and conversely taking action α_4 in state s_2 promotes the value of safety of the agent. Finally, performing action α_5 in state s_3 means stepping into water. As the agent is a robot, which should avoid getting wet, this choice will demote the value of maintaining good conditions of the agent. The agent can verify whether he can achieve his goal while promoting or demoting a specific value by performing a sequence of actions. The verification results are listed below:*

$$VT, s_0 \vDash_{-pv} Do(\alpha_1)Do(\alpha_6)p$$

$$VT, s_0 \vDash_{+pv} Do(\alpha_2)Do(\alpha_3)p$$

$$VT, s_0 \vDash_{-sf} Do(\alpha_2)Do(\alpha_3)p$$

$$VT, s_0 \vDash_{+pv} Do(\alpha_2)Do(\alpha_4)Do(\alpha_5)p$$

$$VT, s_0 \vDash_{+sf} Do(\alpha_2)Do(\alpha_4)Do(\alpha_5)p$$

$$VT, s_0 \vDash_{-gc} Do(\alpha_2)Do(\alpha_4)Do(\alpha_5)p$$

3. Planning: an Argumentation-based Approach

Given a transition system and an agent's goal, model checking and verification techniques allow us to verify whether an agent can achieve his goal while promoting or demoting a specific value by performing a sequence of actions. Since following different plans might promote or demote different sets of values, next question is how the agent decides what to do given the verification results. In this paper, we propose to use argumentation as a technique for an agent's decision-making. Formal argumentation is a nonmonotonic formalism for representing and reasoning about conflicts based on the construction and the evaluation of interacting arguments [8]. In particular, it has been used in practical reasoning, which is concerned by reasoning about what agents should do, given different alternatives and outcomes they bring about [2][10]. We first define the notion of plans. A plan is defined as a finite sequence of actions that are enabled by our underlying transition system. Formally,

Definition 4 (Plans)**.** *Given a pointed value-based transition system (VT, s) and a formula $p \in \mathcal{L}_{prop}$ as an agent's goal, a plan is defined as a finite sequence of actions over Act, denoted as $\lambda = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, such that $VT, s \vDash Do(\alpha_1)Do(\alpha_2)\ldots Do(\alpha_n)p$.*

The definition is equivalent to saying that a plan is a sequence of actions, each of which can be performed succinctly with respect to the pointed value-based transition system, to achieve the agent's goal. The agent has to reason about the available plans with respect to their goal achievement and value promotion or demotion. In order to do that, it is intuitive to define an argument as a plan together with its local property of value promotion or demotion. Based on the verification results, we define two types of arguments.

Definition 5 (Ordinary Arguments and Blocking Arguments)**.** *Given a pointed value-based transition system (VT,s), a formula $p \in \mathcal{L}_{prop}$ as an agent's goal, a plan $\lambda = (\alpha_1,\alpha_2,\ldots,\alpha_n)$ and $v \in \mathrm{Val}$, an ordinary argument is a pair $\langle +v,\lambda\rangle$ such that $VT,s \vDash_{+v} Do(\alpha_1)Do(\alpha_2)\ldots Do(\alpha_n)p$; a blocking argument is a pair $\langle -v,\neg\lambda\rangle$ such that $VT,s \vDash_{-v} Do(\alpha_1)Do(\alpha_2)\ldots Do(\alpha_n)p$; we use $\mathcal{A}_o$ (resp. $\mathcal{A}_b$) to denote the set of ordinary arguments (resp. blocking arguments), and $\mathcal{A} = \mathcal{A}_o \cup \mathcal{A}_b$ to denote the set of two types of arguments.*

Both an ordinary argument and a blocking argument correspond to a verification result. An ordinary argument $\langle +v,\lambda\rangle$ is interpreted as "the agent should follow plan λ to achieve his goal because it promotes a value v", which supports the performance of plan λ, and a blocking argument $\langle -v,\neg\lambda\rangle$ is interpreted as "the agent should not follow plan λ to achieve his goal because it demotes a value v", which objects to the performance of plan λ. Conventionally, we might represent an argument using an alphabet $(a,b,\ldots)$ if we do not care about the internal structure of the argument.

Example 3. *From the verification results listed in Example 2, the agent can construct the following arguments: $\langle -pv,\neg(\alpha_1,\alpha_6)\rangle$, $\langle +pv,(\alpha_2,\alpha_3)\rangle$, $\langle -sf,\neg(\alpha_2,\alpha_3)\rangle$, $\langle +pv,(\alpha_2,\alpha_4,\alpha_5)\rangle$, $\langle +sf,(\alpha_2,\alpha_4,\alpha_5)\rangle$ and $\langle -gc,\neg(\alpha_2,\alpha_4,\alpha_5)\rangle$.*

When we get to choose a plan to follow, there are conflicts between the alternatives as they cannot be followed all at the same time. The conflicts are interpreted as attacks between two ordinary arguments supporting different plans and one ordinary argument and one blocking argument supporting and objecting to the same plan respectively in this paper.

Definition 6 (Attacks)**.** *Given a set of ordinary arguments $\mathcal{A}_o$ and a set of blocking arguments $\mathcal{A}_b$,*

- *for any two ordinary arguments $\langle +v_a,\lambda_a\rangle,\langle +v_b,\lambda_b\rangle \in \mathcal{A}_o$, $\langle +v_a,\lambda_a\rangle$ attacks $\langle +v_b,\lambda_b\rangle$ iff $\lambda_a \neq \lambda_b$;*
- *for any ordinary argument $\langle +v_a,\lambda_a\rangle \in \mathcal{A}_o$ and any blocking argument $\langle -v_b,\neg\lambda_b\rangle \in \mathcal{A}_b$,*
    - *$\langle +v_a,\lambda_a\rangle$ attacks $\langle -v_b,\neg\lambda_b\rangle$ iff $\lambda_a = \lambda_b$;*
    - *$\langle -v_b,\neg\lambda_b\rangle$ attacks $\langle +v_a,\lambda_a\rangle$ iff $\lambda_a = \lambda_b$.*

The set of attacks over $\mathcal{A}$ are denoted as $\mathcal{R}$.

It is obvious that our attack relation is mutual. It should be noticed that there is no attack between two blocking arguments, as a blocking argument only functions as blocking the conclusion of an ordinary argument but does not make a conclusion by itself.

The attack relation represents conflicts between plans. However, the notion of attack may not be sufficient for modeling conflicts between arguments, as an agent has his preference over the values that are promoted or demoted by different plans. In structured argumentation frameworks such as *ASPIC*$^+$ [13], an argument a can be used as a counter-argument to another argument b, if a successfully attacks, i.e. defeats, b. Whether an attack from a to b (on its sub-argument b') succeeds as a defeat, may depend on the relative strengths of a and b, i.e. whether a is strictly stronger than, or strictly preferred over b'. For this paper, recall that an agent has a value system, which was defined as a total pre-order over a set of values. We can then determine the preference over two arguments with respect to value promotion and demotion based on the value system. The notion of defeats combines the notions of attack and preference.

Definition 7 (Defeats)**.** *Given a set of ordinary arguments $\mathcal{A}_o$ and a set of blocking arguments $\mathcal{A}_b$, a set of attacks $\mathcal{R}$ over $\mathcal{A}$ and a value system V,*

- *for any two ordinary arguments $\langle +v_a, \lambda_a\rangle, \langle +v_b, \lambda_b\rangle \in \mathcal{A}_o$, $\langle +v_a, \lambda_a\rangle$ defeats $\langle +v_b, \lambda_b\rangle$ iff $\langle +v_a, \lambda_a\rangle$ attacks $\langle +v_b, \lambda_b\rangle$ and $v_a \nprec v_b$;*
- *for any ordinary argument $\langle +v_a, \lambda_a\rangle \in \mathcal{A}_o$ and any blocking argument $\langle -v_b, \neg\lambda_b\rangle \in \mathcal{A}_b$,*
  - *$\langle +v_a, \lambda_a\rangle$ defeats $\langle -v_b, \neg\lambda_b\rangle$ iff $\langle +v_a, \lambda_a\rangle$ attacks $\langle -v_b, \neg\lambda_b\rangle$ and $v_a \nprec v_b$;*
  - *$\langle -v_b, \neg\lambda_b\rangle$ defeats $\langle +v_a, \lambda_a\rangle$ iff $\langle -v_b, \neg\lambda_b\rangle$ attacks $\langle +v_a, \lambda_a\rangle$ and $v_b \nprec v_a$.*

The set of defeats over $\mathcal{A}$ based on an attack relation and a value system are denoted as $\mathcal{D}(\mathcal{R}, V)$. We write $\mathcal{D}$ for short if it is clear from the context.

In words, given mutual attacks between two arguments, the attack from the argument with less preferred value to the attack from the argument with a more preferred value does not succeed as a defeat. One might ask whether it is more convenient to combine the notions of attack relation and defeat relation. We argue that two notions represent the relation between two arguments from different perspectives, one for the conflicts between plans and the other for the preferences over values. Because of that, defining these two notions separately can make our framework more clear, even though technically it is possible to combine them. It is obvious to see that our defeat relation can form a *two-length cycle* in which two arguments have equivalent or the same values.

Proposition 1. *Given two ordinary arguments $\langle +v_a, \lambda_a\rangle, \langle +v_b, \lambda_b\rangle \in \mathcal{A}_o$, $\langle +v_a, \lambda_a\rangle$ and $\langle +v_b, \lambda_b\rangle$ form a two-length cycle iff $\lambda_a \neq \lambda_b$ and ($v_a = v_b$ or $v_a \sim v_b$). Given an ordinary argument $\langle +v_a, \lambda_a\rangle \in \mathcal{A}_o$ and a blocking argument $\langle -v_b, \neg\lambda_b\rangle \in \mathcal{A}_b$, $\langle +v_a, \lambda_a\rangle$ and $\langle -v_b, \neg\lambda_b\rangle$ form a two-length cycle iff $\lambda_a = \lambda_b$ and ($v_a = v_b$ or $v_a \sim v_b$).*

Proof. Proof follows from Definition 7 directly. □

However, we have the result that our defeat relation obeys the property of irreflexivity and never forms any odd cycle.

Proposition 2. *Given a set of arguments $\mathcal{A}$, a defeat relation $\mathcal{D}$ on $\mathcal{A}$ never forms any odd cycle.*

Proof. According to Definition 7, in order for an argument a to defeat another argument b, the value v_a that belongs to a must be not less preferred than v_b that belongs to b. Since an agent's value system is a total pre-order over a set of values, arguments can only form a cycle in which any two arguments are mutually defeated with the values involved are equivalent or the same. So $\mathcal{D}$ on $\mathcal{A}$ never forms any odd cycle. □

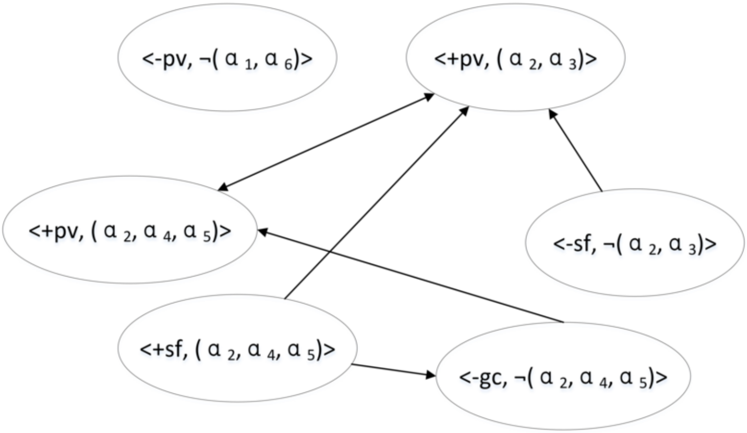


Figure 3. An argumentation framework.

Proposition 3. *Given a set of arguments $\mathcal{A}$, a defeat relation $\mathcal{D}$ on $\mathcal{A}$ is irreflexive.*

Proof. It is a special case of Proposition 2 for the number of arguments in the odd cycle being one. □

We are now ready to construct a Dung-style abstract argumentation framework with ordinary arguments, blocking arguments and the defeat relation on them.

Definition 8 (Plan-based Argumentation Frameworks)**.** *Given a pointed value-based transition system (VT, s) and a formula $p \in \mathcal{L}_{prop}$ as an agent's goal, a plan-based argumentation framework over (VT, s) and p is a pair $PAF = (\mathcal{A}, \mathcal{D})$, where $\mathcal{A}$ is a set of arguments and $\mathcal{D}$ is a defeat relation on $\mathcal{A}$.*

Example 4. *The running example has three ordinary arguments and three blocking arguments, and any two ordinary arguments with different plans are mutually attacked, and any ordinary argument and blocking argument with the same plan are mutually attacked. Suppose the agent has a value system as $pv \prec gc \prec sf$, which means that safety is more important than keeping good condition, and keeping good condition is more important than privacy. We then can see some of the attacks do not succeed as defeats. For example, argument $\langle +pv, (\alpha_2, \alpha_4, \alpha_5)\rangle$ and argument $\langle -gc, \neg(\alpha_2, \alpha_4, \alpha_5)\rangle$ are mutually attacked, but since pv is less preferred than gc, only the attack from argument $\langle -gc, \neg(\alpha_2, \alpha_4, \alpha_5)\rangle$ to argument $\langle +pv, (\alpha_2, \alpha_4, \alpha_5)\rangle$ becomes a defeat. For arguments $\langle +pv, (\alpha_2, \alpha_4, \alpha_5)\rangle$ and $\langle +pv, (\alpha_2, \alpha_3)\rangle$, since $pv = pv$, the mutual attacks between them succeed as mutual defeats. For the space limitation, we do not analyze all the defeats, which is depicted in Figure 3. Interestingly, argument $\langle -pv, \neg(\alpha_1, \alpha_6)$ does not receive any defeats or defeat any arguments not because pv is the most preferred value, but because there is no ordinary argument with plan (α_1, α_6).*

Given a plan-based argumentation framework *PAF*, status of arguments is evaluated, producing sets of arguments that are acceptable together, which are based on the notions of conflict-freeness, acceptability and admissibility. The well-known argumentation semantics are listed as follows, each of which provides a pre-defined criterion for determining the acceptability of arguments in a *PAF* [8].

Definition 9 (Conflict-freeness, Acceptability, Admissibility and Extensions)**.** *Given $PAF = (\mathcal{A}, \mathcal{D})$ and $E \subseteq \mathcal{A}$,*

- *A set E of arguments is conflict-free iff there does not exist $a, b \in E$ such that $(a, b) \in \mathcal{D}$.*
- *An argument $a \in \mathcal{A}$ is acceptable w.r.t. a set E (a is defended by E), iff $\forall (b, a) \in \mathcal{D}$, $\exists c \in E$ such that $(c, b) \in \mathcal{D}$.*
- *A conflict-free set of arguments E is admissible iff each argument in E is acceptable w.r.t. E.*
- *E is a complete extension of PAF iff E is admissible and each argument in $\mathcal{A}$ that is acceptable w.r.t. E is in E.*
- *E is the grounded extension of PAF iff E is the minimal (w.r.t. set inclusion) complete extension.*
- *E is the preferred extension of PAF iff E is a maximal (w.r.t. set inclusion) complete extension.*
- *E is a stable extension of PAF iff E is conflict-free and $\forall b \in \mathcal{A} \backslash E$, $\exists a \in E$ such that $(a, b) \in \mathcal{D}$.*

We use $sem \in \{cmp, prf, grd, stb\}$ to denote the complete, preferred, grounded and stable semantics, respectively, and $\mathcal{E}_{sem}(PAF)$ to denote the set of extensions of *PAF* under a semantics in *sem*. The following propositions characterize our argumentation framework in terms of Dung's semantics.

Proposition 4. *Given $PAF = (\mathcal{A}, \mathcal{D})$, $\mathcal{E}_{prf}(PAF) = \mathcal{E}_{stb}(PAF)$.*

Proof. Since our defeat relation $\mathcal{D}$ never forms an odd cycle by Proposition 2, which means that *PAF* is limited controversial, each preferred extension of *PAF* is stable. Detailed proof can be found in [8]. □

Proposition 5. *Given $PAF = (\mathcal{A}, \mathcal{D})$, if there exists an ordinary argument $\langle +v_a, \lambda_a \rangle$ such that for all $\langle +v_b, \lambda_b \rangle \in \mathcal{A}_o$ and $\langle -v_b, \neg\lambda_b \rangle \in \mathcal{A}_b$ it is the case that $v_b \precsim v_a$ and $\langle +v_a, \lambda_a \rangle$ is not in any cycle, then $\mathcal{E}_{prf}(PAF) = \mathcal{E}_{grd}(PAF)$.*

Proof. Because $v_b \precsim v_a$ and $\langle +v_a, \lambda_a \rangle$ is not in any cycle, argument $\langle +v_a, \lambda_a \rangle$ does not receive any defeats. So the grounded extension is not an empty set. Suppose $\mathcal{E}_{prf}(PAF) \neq \mathcal{E}_{grd}(PAF)$, which means that there are more than one preferred extensions. Since $\langle +v_a, \lambda_a \rangle$ is contained in the grounded extension, it should also be contained in each preferred extension. However, each preferred extension indicates a distinct plan, which will be later proved by Proposition 6 and its implication. Contradiction! □

The justification of optimal plans is then defined under various semantics in Definition 9. Similarly to [14], we write $\text{concl}(\langle +v, \lambda \rangle)$ for the conclusion λ of an ordinary argument, and $\text{Oplans}(PAF, sem)$ for the set of conclusions of ordinary arguments from the extensions under a specific semantics.

Definition 10 (Optimal Plans)**.** *Given $PAF = (\mathcal{A}, \mathcal{D})$, a set of optimal plans, written as* $\text{Oplans}(PAF, sem)$*, are the conclusions of the ordinary arguments within extensions.*

$$\text{Oplans}(PAF, sem) = \{\text{concl}(\langle +v, \lambda \rangle) \mid \langle +v, \lambda \rangle \in E \text{ and } E \in \mathcal{E}_{sem}(PAF)\}$$

We show that the results of our approach are consistent with the rationality of decision-making through the following propositions. Firstly, all the accepted arguments within an extension indicate the same plan.

Proposition 6. *Given a plan-based argumentation framework PAF* $= (\mathcal{A}, \mathcal{D})$ *and an extension E of PAF under a specific semantics as defined in Definition 9,*

1. *for any two ordinary arguments* $\langle +v_a, \lambda_a \rangle, \langle +v_b, \lambda_b \rangle \in E$*, it is the case that* $\lambda_a = \lambda_b$*;*
2. *for any ordinary argument* $\langle +v_a, \lambda_a \rangle \in E$ *and any blocking argument* $\langle -v_b, \neg\lambda_b \rangle \in E$*,* $\lambda_a \neq \lambda_b$*.*

Proof. For any extension E under a specific semantics, it is required that all the arguments in E should be conflict-free. 1. By Definition 7, we can derive two cases: either there is no attack between these two arguments, or one argument attacks the other but does not succeed as a defeat due to the preference between two values from the arguments. For the former case, two arguments contain the same plan. For the latter case, since any attack between two arguments is mutual, if an attack from argument $\langle +v_a, \lambda_a \rangle$ to argument $\langle +v_b, \lambda_b \rangle$ fails to be a defeat due to the preference between two values from the arguments, the attack from argument $\langle +v_b, \lambda_b \rangle$ to argument $\langle +v_a, \lambda_a \rangle$ will succeed to be a defeat. That means that the second case is impossible and only the first case holds. Hence, the two arguments have the same plan. 2. We can prove in a similar way that for any ordinary argument $\langle +v_a, \lambda_a \rangle \in E$ and any blocking argument $\langle -v_b, \neg\lambda_b \rangle \in E$, $\lambda_a \neq \lambda_b$, □

From that we can see, if there are multiple preferred extensions, then each of them indicates a distinct plan. Secondly, our argumentation-based approach always selects the plan through which the most preferred value gets promoted and does not select the plan through which the most preferred value gets demoted.

Proposition 7. *Given a plan-based argumentation framework PAF* $= (\mathcal{A}, \mathcal{D})$*, let* $v_a \in \mathrm{Val}$ *be a value such that for all* $\langle +v_b, \lambda_b \rangle \in \mathcal{A}_o$ *and* $\langle -v_b, \neg\lambda_b \rangle \in \mathcal{A}_b$ *it is the case that* $v_b \precsim v_a$*, then an argument with value* v_a *is in a preferred extension. Typically, if it is not in a cycle, then it is in the grounded extension.*

Proof. Because $v_b \precsim v_a$, according to Definition 7, an argument with v_a only gets defeated by an argument with $v_a \sim v_b$ or $v_a = v_b$. In such a case, the defeats are mutual so $\langle +v_a, \lambda_a \rangle$ is self-defended. Thus, it is contained in a preferred extension. If it is not in a cycle, which means that it is not self-defended, then it is in the grounded extension. □

Because of the above two propositions, the agent can conclude to follow an optimal plan to achieve his goal. Besides, the notion of optimal plans is defined as the set of conclusions of ordinary arguments from the extensions, so the set of optimal plans becomes empty if an extension does not contain any ordinary arguments. The following proposition indicates the conditions for which the set of optimal plans is not empty.

Proposition 8. *Given a PAF* $= (\mathcal{A}, \mathcal{D})$, $\mathrm{Oplans}(PAF, sem) \neq \emptyset$ *iff there exists an ordinary argument* $\langle +v_a, \lambda_a \rangle$ *such that it is not defeated by a blocking argument* $\langle -v_b, \neg\lambda_b \rangle$ *with* $v_a \prec v_b$*.*

Proof. Having $\mathrm{Oplans}(PAF, sem) \neq \emptyset$ means that there is at least one extension which contains at least one ordinary argument. ⇒: Suppose there does not exists an ordinary argument $\langle +v_a, \lambda_a \rangle$ such that it is not defeated by a blocking argument $\langle -v_b, \neg\lambda_b \rangle$ with $v_a \prec v_b$, which means that all the ordinary arguments (if exist) are defeated by a blocking argument and not self-defended against a blocking argument. In such a case, there exists a blocking argument that does not receive any defeats, which makes all the ordinary arguments rejected. Contradiction! ⇐: If there exists an ordinary argument such that it is not defeated by a blocking argument or self-defended against a blocking argument, then (1) the ordinary argument does not receive any defeats and thus it should be contained in the grounded extension, or (2) the ordinary argument is in a cycle with a blocking argument and thus it should be contained in a preferred extension, or (3) the ordinary argument receives defeats from other ordinary arguments and thus there is always an ordinary argument accepted. Hence, $\mathrm{Oplans}(PAF, sem)$ is not an empty set. □

Example 5. *The plan-based argumentation framework PAF can be represented as Figure. 3. Because*

$$\begin{aligned}\mathcal{E}_{prf}(PAF) = &\mathcal{E}_{grd}(PAF) = \mathcal{E}_{stb}(PAF) = \\ &\{\{\langle +pv, (\alpha_2, \alpha_4, \alpha_5)\rangle, \langle +sf, (\alpha_2, \alpha_4, \alpha_5)\rangle, \\ &\langle -pv, \neg(\alpha_1, \alpha_6)\rangle, \langle -sf, \neg(\alpha_2, \alpha_3)\rangle\}\}\end{aligned}$$

and thus $\mathrm{Oplans}(PAF, sem) = \{(\alpha_2, \alpha_4, \alpha_5)\}$*, the agent can follow plan* $(\alpha_2, \alpha_4, \alpha_5)$ *to get to a pharmacy.*

4. Related Work

Our approach is closely related to the value-based argumentation framework (VAF) [2][1]. The differences are as follows: firstly, their framework allows us to reason about what we should do given all the available actions, while our framework allows us to reason about what we should do given all the available plans each of which is a sequence of actions; secondly, in their framework informal arguments are constructed through asking critical questions associated with an argument scheme and a transition system, while in our framework agents construct formal arguments through checking formulas with respect to the underlying transition system; thirdly, the aim of VAF (having an audience as an element) is to explain different agents' choices based on their social values, while the aim of our paper is to design an approach for agents' planning. Existing work combines modal logic and argumentation in different ways. Proietti and Yuste-Ginel combines both techniques to reason about the knowledge of arguments in a debate and its dynamics [15], and Bulling etc. combine both techniques to reason about the abilities of coalitions of agents and the formation of coalitions [16]. Both use modal logic as meta-language to argumentation, while we use argumentation as meta-language to modal logic.

5. Conclusions

In this paper, we developed a logic-based framework that combines modal logic and argumentation for value-based practical reasoning. Modal logic is used as a technique to represent and verify whether a plan with its local properties of value promotion or demotion can be followed to achieve an agent's goal. Seeing a verification result as an argument and defining a defeat relation based on an attack relation and preference over values, we then proposed an argumentation-based approach that allows an agent to reason about his plans using the verification results. Thus, our framework not only offers an approach for value-based practical reasoning with plans, but also makes a bridge between modal logic and argumentation in terms of argument construction. In the future, we would like to extend our framework by allowing an agent to have multiple goals instead of one goal as we assumed, or taking the actions of other agents into account in the context of multi-agent systems. More interestingly, we can study how autonomous agents are properly aligned with human values through adding constraints to the decision-making mechanism presented in this paper.

References

[1] Atkinson K, Bench-Capon T. Taking account of the actions of others in value-based reasoning. Artificial Intelligence. 2018;254:1-20.

[2] Bench-Capon T, Atkinson K, McBurney P. Using argumentation to model agent decision making in economic experiments. Autonomous Agents and Multi-Agent Systems. 2012;25(1):183-208.

[3] Liao B, Anderson M, Anderson SL. Representation, justification, and explanation in a value-driven agent: an argumentation-based approach. AI and Ethics. 2021;1(1):5-19.

[4] Ågotnes T, van der Hoek W, Wooldridge M. Normative system games. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems; 2007. p. 1-8.

[5] Knobbout M, Dastani M. Reasoning under compliance assumptions in normative multiagent systems. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems; 2012. p. 331-40.

[6] Alechina N, Dastani M, Logan B. Reasoning about Normative Update. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. IJCAI '13. AAAI Press; 2013. p. 20–26.

[7] Luo J, Meyer JJ, Knobbout M. A formal framework for reasoning about opportunistic propensity in multi-agent systems. Autonomous Agents and Multi-Agent Systems. 2019;33(4):457-79.

[8] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence. 1995;77(2):321-57.

[9] Prakken H. Combining sceptical epistemic reasoning with credulous practical reasoning. COMMA. 2006;144:311-22.

[10] Amgoud L, Prade H. Formalizing practical reasoning under uncertainty: An argumentation-based approach. In: 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07). IEEE; 2007. p. 189-95.

[11] Knobbout M, Dastani M, Meyer JJ. A dynamic logic of norm change. In: Proceedings of the Twenty-second European Conference on Artificial Intelligence; 2016. p. 886-94.

[12] Knobbout M, Dastani M, Meyer JJC. Reasoning about dynamic normative systems. In: European Workshop on Logics in Artificial Intelligence. Springer; 2014. p. 628-36.

[13] Modgil S, Prakken H. A general account of argumentation with preferences. Artificial Intelligence. 2013;195:361-97.

[14] Liao B, Oren N, van der Torre L, Villata S. Prioritized norms in formal argumentation. Journal of Logic and Computation. 2019;29(2):215-40.

[15] Proietti C, Yuste-Ginel A. Dynamic epistemic logics for abstract argumentation. Synthese. 2021;199(3):8641-700.

[16] Bulling N, Dix J, Chesnevar CI. Modelling coalitions: ATL+ argumentation. In: AAMAS (2); 2008. p. 681-8.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220158

On the Complexity of Determining Defeat Relations Consistent with Abstract Argumentation Semantics

Jack MUMFORD [a,b], Isabel SASSOON [c], Elizabeth BLACK [b] and Simon PARSONS [d]

[a] *Department of Computer Science, University of Liverpool, UK*
[b] *Department of Informatics, King's College London, UK*
[c] *Department of Computer Science, Brunel University London, UK*
[d] *School of Computer Science, University of Lincoln, UK*

Abstract. Typically in abstract argumentation, one starts with arguments and a defeat relation, and applies some semantics in order to determine the acceptability status of the arguments. We consider the converse case where we have knowledge of the acceptability status of arguments and want to identify a defeat relation that is consistent with the known acceptability data – the σ-consistency problem. Focusing on complete semantics as underpinning the majority of the major semantic types, we show that the complexity of determining a defeat relation that is consistent with some set of acceptability data is highly dependent on how the data is labelled. The extension-based 2-valued σ-consistency problem for complete semantics is revealed as NP-complete, whereas the labelling-based 3-valued σ-consistency problem is solvable within polynomial time. We then present an informal discussion on application to grounded, stable, and preferred semantics.

Keywords. Abstract argumentation, Complexity analysis, σ-consistency

1. Introduction

The typical argumentation problem takes arguments, a defeat relation, and a labelling semantics as input and produces argument acceptability labellings as output. We instead examine a converse argumentation problem, which takes arguments, labelling semantics, and argument labellings as input and produces a defeat relation as output.

Argumentation is firmly established as a subject of importance for researchers interested in symbolic representations of knowledge and defeasible reasoning [5]. Argumentation frameworks (AFs) [10] offer a graph-based approach that determines logically consistent positions through the interaction of arguments solely through a defeat relation. Determining the presence, or absence, of defeats between arguments is fundamental to the construction of any AF and is the problem that concerns this paper. Given a set of arguments, and data on which arguments are acceptable, we examine the complexity of establishing a set of defeats that is consistent with the data.

In certain contexts a defeat relation between arguments can be reliably and efficiently inferred from the structure and content of the arguments themselves. However, if the structure and/or the content of the arguments cannot be relied upon, then we become more dependent upon the remaining two components of an AF: the labelling semantics, and the argument labellings. For example, enthymemes – arguments with missing structure – pose problems for inferring a defeat relation. Suppose if some argument a defeats some argument b via an undermining attack on an unstated implicit premise of b, then how is one to recognise the defeat exists? Perhaps knowledge of the contextual content of the arguments fills in the gap of the missing premise, but perhaps not. However, if we possess an argument labelling where a is accepted and b is rejected, then the defeat may be inferred from this information alone.

In this paper we shall examine the σ-consistency problem, whereby we seek a solution defeat relation that is wholly consistent with input acceptability data for a given semantics. It is not assumed that the input data are exhaustive; other labellings that are not present in the data set may also be consistent with the solution defeat relation. Moreover, the problem is general in that it accepts partial labellings where not every argument in the overall argument set need be represented in a given labelling. That is, an arbitrary labelling in the data set may pertain to a subgraph of arguments S where $S \subset A$, or it may pertain to all of A.

Example 1. *Legal domains are commonly modelled using stereotypical patterns of facts known as factors, following the CATO approach for legal case-based reasoning [1]. A case is decided by weighing up the balance of the factors from case precedents. This process could be modelled by the creation of a representative AF model in accordance with the σ-consistency problem. For example, in CATO's domain, US Trade Secret's Law, the two factors related to the issue of confidentiality were F1 (DisclosureInNegotiations) and F21 (KnewInfoConfidential). There is nothing explicit within the structure or content of the factors that definitively reveals how the factors are appropriately balanced. However, now suppose that we have a case history such that whenever F1 and F21 are argued (indeed F1 and/or F21 may not be argued in a case, but recall that the σ-consistency problem accepts partial labellings), we find F1 is rejected (labelled* OUT*) and F21 is accepted (labelled* IN*), we have a way of establishing a defeat relation where $(F21, F1) \in R$, that would be consistent with the precedents, thus presenting a method of balancing the factors. Conversely, if we then were presented with a new case in which both factors were accepted (labelled* IN*) then no defeat relation exists that would be consistent with the amended case history. But even a failure to produce a solution serves a purpose; indicating that the existing factor-based model of the domain may be insufficient and require broadening or narrowing. Note that in the actual CATO analysis such a converse case never arose and $(F21, F1) \in R$ was the appropriate conclusion.*

Whilst many semantics are available for investigation, our primary attention is placed on complete semantics due to the superset relation it forms with respect to the traditional major semantics as well as the intuition of it representing the range of reasonable positions that are deterministic consequences of the defeat relation (which admissibility semantics does not guarantee). We will show that the complexity of a σ-consistency problem with complete semantics (note from this point onward we shall refer to a 'σ-consistency with complete semantics' simply as a 'σ-consistency problem') depends significantly on the type of input acceptability data, where the extension-based 2-valued problem (arguments labelled IN or NOTIN) is **NP**-complete and the 3-valued problem (arguments labelled IN, UNDEC or OUT) can be solved within polynomial, specifically multi-variate quadratic, time complexity (although both are solvable within quadratic space complexity). We believe this paper offers the following contributions:

- Introduces and defines the σ-consistency problem, which asks if a solution defeat relation exists that is consistent with a set of argument acceptability labellings.
- Proves that the 2-valued σ-consistency problem is **NP**-complete and show its space complexity to be within $O(n^2)$, where n is the number of arguments.
- Proves the 3-valued σ-consistency problem is solvable by algorithms with a time complexity within $O(n^2|T|)$, where $|T|$ is the number of labellings in the input, and space complexity within $O(n^2)$.
- Provides informal complexity results for stable, preferred, and grounded semantics.
- Provides insight into the scaleability of algorithms designed to determine a defeat relation from argument acceptability data, suggesting a preference for sourcing 3-valued data to reduce the risk of solving high complexity problems.

2. Background

We draw upon the concept of argumentation underpinned by Dung's seminal paper [10] and the reinstatement principles advocated by Caminada [6] in developing our complexity proofs for the 2-valued and 3-valued σ-consistency problems.

An *argumentation framework* is a pair $\mathrm{AF} = \langle A, R\rangle$ where A is a set of arguments, and R is a binary relation on A, i.e. $R \subseteq A \times A$. We apply the notation for defeat from [6]: Let $a, b \in A$, we define a^+ as $\{b|a \text{ defeats } b\}$ and a^- as $\{b|b \text{ defeats } a\}$. A set S of arguments is said to be *conflict-free* if there are no arguments a and b in S such that $a \in b^-$. A set of arguments S defeats an argument a *iff* $\exists b \in S : b \in a^-$. An argument $a \in A$ is *acceptable* with respect to a set S *iff* for each argument $b \in A$: if $b \in a^-$ then $\exists c \in S : b \in c^+$. A conflict-free set of arguments S is *admissible iff* each argument in S is acceptable with respect to S. An admissible set S of arguments is a *complete extension iff* each argument that is acceptable with respect to S belongs to S. In a 2-valued complete labelling, all arguments in S are labelled IN, and all arguments in $A \setminus S$ are labelled NOTIN.

Def. 1 prescribes the implementation of the reinstatement labelling process to produce a 3-valued complete labelling, which coincides with complete extension semantics [7]. The formulation has been adopted from [24] with some notation changed in order to maintain consistency throughout this paper.

Definition 1. (3-valued complete labelling) *Let $L : A \to \{\text{IN}, \text{UNDEC}, \text{OUT}\}$ be a labelling of argumentation framework (A, R). We say that L is a complete labelling iff for each argument $b \in A$ it holds that:*

1. $(L(b) = \text{IN}) \equiv (\forall a \in A : (a \in b^-) \Rightarrow (L(a) = \text{OUT}))$*;*
2. $(L(b) = \text{OUT}) \equiv (\exists a \in A : (a \in b^-) \wedge (L(a) = \text{IN}))$*; and*
3. $(L(b) = \text{UNDEC}) \equiv ((\exists a \in A : (a \in b^-) \wedge (L(a) = \text{UNDEC})) \wedge (\forall c \in A : (c \in b^-) \Rightarrow (L(c) \neq \text{IN}))$.

A signature $\Sigma_\sigma((A,R))$ of a semantics σ, is the collection of all possible sets of labellings from all subgraphs of $\mathrm{AF} = \langle A, R\rangle$ under semantics σ [11]. We use the concept of a signature for both the 2-valued and 3-valued σ-consistency problems, since any solution defeat relation R would require all input acceptability labellings to be in $\Sigma_\sigma((A,R))$.

Definition 2. (Signatures of argumentation semantics, adapted – with notation changes – from Dunne et al. [11]) *Let $\sigma((S,R))$ be the set of possible labellings given arguments S, defeat relation R, and argumentation semantics σ. We can express the set of all possible labellings over a set of input AFs as $\Sigma_\sigma((A,R))$ such that:*

$$\Sigma_\sigma((A,R)) = \{\sigma((S,R))|(S,R) \text{ is an AF and } S \subseteq A\}$$

3. Complexity of 2-valued σ-consistency

In order to formalise the 2-valued σ-consistency problem we must adapt the properties governing complete extension semantics, rendering explicit the 2-valued labelling form that the semantics prescribes: for each extension, an argument is labelled IN *iff* it is within the extension, and NOTIN *iff* it is outside the extension. Def. 3 concisely presents complete extension semantics according to the 2-valued labelling form. We then further adapt this form to produce our complexity results. We add the OFF label to represent those arguments that are unlabelled in a particular reinstatement labelling. The OFF label is essential for when we formalise the 2-valued σ-consistency problem, since it allows us to include partial labellings representing subgraphs of the wider AF.

Definition 3. (2-valued complete labelling) *Let A be a set of arguments; let $L : A \to \{\text{IN}, \text{NOTIN}, \text{OFF}\}$ be a labelling; let $R \subseteq A \times A$ be a defeat relation. $L \in \sigma((S,R))$, where $S = A \setminus X$, and where X is the set of arguments labelled* OFF, *iff each of the following conditions hold $\forall b \in S$:*

1. *$L(b) = \text{IN} \iff \forall a \in S : ((L(a) = \text{IN} \implies (a,b) \notin R) \wedge (L(a) = \text{NOTIN} \implies (a,b) \notin R \vee \exists c \in S : (L(c) = \text{IN} \wedge (c,a) \in R)))$; and*
2. *$L(b) = \text{NOTIN} \iff \exists a \in S : ((L(a) = \text{IN} \wedge (a,b) \in R) \vee (L(a) = \text{NOTIN} \wedge (a,b) \in R \wedge \forall c \in S : (L(c) = \text{IN} \implies (c,a) \notin R)))$.*

Def. 4 formally presents the 2-valued σ-consistency problem of finding a solution defeat relation set consistent with 2-valued input argument acceptability data. From this definition we produce the principle complexity result in Theorem 2, that the σ-consistency problem is **NP**-complete.

Definition 4. (2-valued σ-consistency problem) *Given a set of arguments A, a semantics σ, and a set of labellings T such that for each $L \in T$, $L : A \to \{\text{IN}, \text{NOTIN}, \text{OFF}\}$, is there a defeat relation $R \subseteq A \times A$ consistent with T such that $\forall L : (L \in \sigma((S,R)) \in \Sigma_\sigma((A,R)))$, where $S = A \setminus X$, and where X is the set of arguments labelled* OFF*?*

It should be obvious that in general one should not assume there exists just one solution defeat relation for an arbitrary 2-valued σ-consistency problem (e.g. an input labelling of two arguments where one argument is labelled IN, and the other NOTIN has more than one defeat relation that could produce this labelling). Before we move on to the full complexity proof, we make the intermediate observation in Theorem 1 of the symmetry of defeat inconsistency – that for complete semantics if some defeat $(a,b) \in R$ is inconsistent with some labelling then (b,a) would also be inconsistent with the labelling.

Theorem 1. (Bidirectional defeat inconsistency for 2-valued labellings) *Assuming complete argumentation semantics, for any set of labellings* $T : \forall L \in T,\ L : A \to \{\text{IN}, \text{NOTIN}, \text{OFF}\}$*, we have* $\exists a,b \in A : ((a,b) \in R \implies L \notin \sigma((S,R)) \iff (b,a) \in R \implies L \notin \sigma((S,R)))$*, where* $S = A \setminus X$*, and* X *is the set of arguments labelled* OFF.

Proof. Suppose we have σ as complete semantics, a and b are arbitrary arguments in A, and $S = A \setminus X$, and where X is the set of arguments labelled OFF.
(*Forward implication*) Suppose $(a,b) \in R \implies L \notin \sigma((S,R))$. From Def. 3 we draw expressions of the form $L(a) \wedge L(b) \implies (a,b) \notin R$. We have three cases:
Case 1. $\forall a,b \in S : (L(b) = \text{IN} \wedge L(a) = \text{IN} \implies (a,b) \notin R)$. Suppose $(b,a) \in R$, then by the first condition of Def. 3 $L(a) = \text{IN} \implies (b,a) \notin R$. Contradiction!
Case 2. $\forall a,b \in S : ((L(b) = \text{IN} \wedge L(a) = \text{NOTIN} \implies (a,b) \notin R) \implies \nexists c \in S : (L(c) = \text{IN} \wedge (c,a) \in R))$. Suppose $(b,a) \in R$, then $\exists c \in S : (L(c) = \text{IN} \wedge (c,a) \in R)$. Contradiction!
Case 3. $\forall a,b \in S : ((L(b) = \text{NOTIN} \wedge L(a) = \text{IN} \implies (a,b) \notin R) \implies \exists c \in S : (L(c) = \text{NOTIN} \wedge (b,c)) \wedge \forall d \in (S \setminus b) : (\forall e \in S : (L(e) = \text{IN} \implies (e,d) \notin R) \wedge (d,c) \notin R))$. Suppose $(b,a) \in R$, then by the first condition of Def. 3 $\exists f \in S : (L(f) = \text{IN} \wedge (f,b) \in R)$. Contradiction!
Therefore if $(a,b) \in R \implies L \notin \sigma((S,R))$ then $(b,a) \in R \implies L \notin \sigma((S,R))$.
(*Backward implication*) Since a and b are arbitrary, the backward implication immediately follows from the permutation $a \leftrightarrow b$ in the forward implication.
Therefore if $(b,a) \in R \implies L \notin \sigma((S,R))$ then $(a,b) \in R \implies L \notin \sigma((S,R))$. □

We shall see later in the paper that the symmetry of inconsistency also holds for the 3-valued σ-consistency problem. Hence the symmetry of defeat inconsistency with acceptability data underpins any σ-consistency problem based on complete semantics. It is essential to understand that Theorem 1 applies to asymmetric as well as symmetric frameworks, but simply outlines that if an arbitrary defeat (a,b) is incompatible with a labelling set, then (b,a) will also be incompatible.

Theorem 2. (2-valued σ-consistency problem as **NP**-complete) *The 2-valued* σ*-consistency problem as defined in Def. 4 is* **NP***-complete.*

Proof. Suppose there is a set of arguments A, labellings T, and complete semantics σ.
It is easy to show that the 2-valued σ-consistency problem is in **NP**. Given a solution defeat relation R, from [8], the verification problem of confirming a given labelling L is consistent with R and σ, is solvable in polynomial-time. Since T is a fixed set of labellings, then the verification problem of confirming each labelling $L \in T$ is consistent with R and σ, is clearly also solvable within polynomial-time. Therefore the 2-valued σ-consistency problem is within **NP**.
We must show that the **NP**-complete Monotone 3SAT problem [14] can be reduced by function r to the 2-valued σ-consistency problem in polynomial-time. An arbitrary instance of a Monotone 3SAT problem consists of a set of clauses C, where we require that C is satisfied if and only if $\forall L_j \in r(C) : (L_j \in \sigma((S,R)))$, where $S = A \setminus X$, and where X is the set of arguments labelled OFF.
Suppose an arbitrary Monotone 3SAT problem and some arguments $\omega, \beta \in A$, where $\omega \neq \beta$. Suppose an arbitrary clause $c_i \in C$, then $a_j \in c_i \iff (a_j, \omega) \in R$ and $\neg a_j \in c_i \iff (a_j, \omega) \notin R$, where a_j is arbitrary. There exist two cases for an arbitrary clause c_i in an Monotone 3SAT problem; suppose we have r such that:

Case 1. $c_i = (a_i \vee b_i \vee c_i)$ and $r(c_i) = (S, L_{1i}, L_{2i})$ where $S = \{a_i, b_i, c_i, \omega\}$, $L_{1i} = \{L(a_i) = \text{NOTIN}, L(b_i) = \text{NOTIN}, L(c_i) = \text{NOTIN}, L(\omega) = \text{NOTIN}\}$, and $L_{2i} = \{L(\omega) = \text{IN}\}$, and a_i, b_i, c_i are arbitrary.
Case 2. $c_i = (\neg a_i \vee \neg b_i \vee \neg c_i)$ and $r(c_i) = (S, L_{3i}, L_{4i})$, where $S = \{a_i, b_i, c_i, \omega, \beta\}$ $L_{3i} = \{L(a_i) = \text{NOTIN}, L(b_i) = \text{NOTIN}, L(c_i) = \text{NOTIN}, L(\omega) = \text{IN}, L(\beta) = \text{NOTIN}\}$, and $L_{4i} = \{L(\omega) = \text{IN}, L(\beta) = \text{IN}\}$, and a_i, b_i, c_i are arbitrary.
We show the forward and backward implications hold for the two possible cases for arbitrary c_i via equivalence. In the reduction we use the shorthand (a,b) to indicate $(a,b) \in R$, and $\neg(a,b)$ to indicate $(a,b) \notin R$.
(*Case 1*) Suppose arbitrary clause $c_k = (a_i \vee b_i \vee c_i)$, from Def. 3 we have $r(c_k) \implies \bigwedge_{y \in S} (\bigvee_{x \in S} (x,y)) \wedge \neg(\omega,\omega)$. Since a_i, b_i and c_i are always labelled NOTIN in every labelling $L_j \in r(C)$, they are all free to defeat themselves and one another. Hence we are left to satisfy $r(c_k) \implies (a_i,\omega) \vee (b_i,\omega) \vee (c_i,\omega) \equiv c_k$.
(*Case 2*) Suppose arbitrary clause $c_m = (\neg a_i \vee \neg b_i \vee \neg c_i)$, from Def. 3 we have $r(c_m) \implies \bigwedge_{y \in S \setminus \omega} ((\omega,y) \vee \bigvee_{x \in S \setminus \omega} ((x,y) \wedge \neg(\omega,x))) \wedge \neg(\omega,\omega) \wedge \bigwedge_{z \in S \setminus \omega} ((z,\omega) \implies (\omega,z)) \wedge \neg(\omega,\omega) \wedge \neg(\omega,\beta) \wedge \neg(\beta,\omega) \wedge \neg(\beta,\beta)$. As outlined in the previous case, a_i, b_i and c_i are free to defeat themselves and one another, and this is also extended to β. Hence we are left to satisfy $r(c_m) \implies \bigvee_{x \in \{a_i,b_i,c_i\}} (\neg(\omega,x)) \wedge \bigwedge_{y \in \{a_i,b_i,c_i\}} ((y,\omega) \implies (\omega,y))$, which by modus tollens further reduces to $r(c_m) \implies \neg(a_i,\omega) \vee \neg(b_i,\omega) \vee \neg(c_i,\omega) \equiv c_m$.
It is clear that the reduction produces a permutation where $((a_i,\omega) \in R) \equiv (a_i = \top)$, $((a_i,\omega) \notin R) \equiv (\neg a_i = \top)$, $((b_i,\omega) \in R) \equiv (b_i = \top)$, $((b_i,\omega) \notin R) \equiv (\neg b_i = \top)$, $((c_i,\omega) \in R) \equiv (c_i = \top)$ and $((c_i,\omega) \notin R) \equiv (\neg c_i = \top)$. Therefore for both cases of arbitrary clause $c_i \in C$, the reduction simply produces a permutation of the monotone 3SAT problem and consequently C is satisfied if and only if $\forall L_j \in r(C) : (L_j \in \sigma((S,R)))$.
It is clear that $r : c_i \mapsto \{S, T\}$, where $S \subseteq \{a_i, b_i, c_i, \omega, \beta\}$ and $T \subseteq \{L_{1i}, L_{2i}, L_{3i}, L_{4i}\}$, requires constant number of operations to produce the elements in S and T for any arbitrary clause $c_i \in C$. Therefore for k clauses the reduction requires $O(k)$ operations, which is within polynomial-time.
Therefore the 2-valued σ-consistency problem is **NP**-complete. □

Deriving the space complexity result is a much simpler affair. For an arbitrary 2-valued σ-consistency problem a systematic search through a sorted argument power set would only need to store the current node in order to know which node to examine next. In which case, each node in the search would be of length n^2. Hence $O(n^2)$ is an upper bound for the space complexity for solving any 2-valued σ-consistency problem.

4. Complexity of 3-valued labellings

In order to formalise the 3-valued σ-consistency problem, we first directly translate the argument-set-based form of Def. 1 to accommodate partial data sets, adding the OFF label to represent those arguments in the wider data set that are unlabelled in a particular reinstatement labelling. The OFF label is essential since it allows a defeat relation R to be learned that is consistent with partial labellings. Def. 5 combines the reinstatement approach adopted in [6] alongside the concept of a signature $\Sigma_\sigma((A,R))$ from Def. 2.

Definition 5. (3-valued form) *Let A be a set of arguments; let $L : A \to \{\text{IN}, \text{UNDEC}, \text{OUT}, \text{OFF}\}$ be a labelling; let $R \subseteq A \times A$ be a defeat relation. $L \in \sigma((S,R))$, where $S = A \setminus X$, and where X is the set of arguments labelled* OFF, *iff each of the following conditions hold $\forall b \in S$:*

1. *$L(b) = \text{IN} \iff \forall a \in S : (L(a) \neq \text{OUT} \implies (a,b) \notin R)$;*
2. *$L(b) = \text{UNDEC} \iff \forall a \in S : (L(a) = \text{IN} \implies (a,b) \notin R) \wedge \exists c \in A : (L(c) = \text{UNDEC} \wedge (c,b) \in R)$; and*
3. *$L(b) = \text{OUT} \iff \exists a \in S : (L(a) = \text{IN} \wedge (a,b) \in R)$.*

Def. 6 formally presents the 3-valued σ-consistency problem of finding a solution defeat relation set consistent with input 3-valued argument acceptability data.

Definition 6. (3-valued σ-consistency problem) *Given a set of arguments A, a semantics σ, and a set of labellings T such that for each $L \in T$, $L : A \to \{\text{IN}, \text{UNDEC}, \text{OUT}, \text{OFF}\}$, is there a defeat relation $R \subseteq A \times A$ consistent with T such that $\forall L : (L \in \sigma((S,R)) \in \Sigma_\sigma((S,R)))$, where $S = A \setminus X$, and where X is the set of arguments labelled* OFF*?*

We again note that no assumption of a single solution defeat relation can be made in general (e.g. a labelling with two arguments labelled IN and one argument labelled OUT could be satisfied by multiple distinct defeat relations) and we gain insight into the complexity of the 3-valued problem by first observing the symmetry of inconsistent defeats in Theorem 3. It is important to note that, as was the case when we considered the ramifications of Theorem 1, Theorem 3 is not limited to symmetric AFs but also applies to asymmetric AFs. Theorem 3 outlines that for complete semantics if an arbitrary defeat $(a,b) \in R$ is incompatible with a labelling set, then $(b,a) \in R$ will also be incompatible.

Theorem 3. (Bidirectional defeat inconsistency for 3-valued labellings) *Assuming complete argumentation semantics, for any set of labellings $T : \forall L \in T$, $L : A \to \{\text{IN}, \text{OUT}, \text{UNDEC}, \text{OFF}\}$, we have $\exists a, b \in A : ((a,b) \in R \implies L \notin \sigma((S,R)) \iff (b,a) \in R \implies L \notin \sigma((S,R)))$, where $S = A \setminus X$, and X is the set of arguments labelled* OFF.

Proof. Suppose we have σ as complete semantics, a and b are arbitrary arguments in A, and $S = A \setminus X$, and where X is the set of arguments labelled OFF.
(*Forward implication*) Suppose $(a,b) \in R \implies L \notin \sigma((S,R))$. From Def. 5 we draw expressions of the form $L(a) \wedge L(b) \implies (a,b) \notin R$. We have three cases:
Case 1. $L(b) = \text{IN} \wedge L(a) = \text{IN} \implies (a,b) \notin R$. Suppose $(b,a) \in R$, then by the first condition of Def. 5 $L(a) = \text{IN} \implies (b,a) \notin R$. Contradiction!
Case 2. $L(b) = \text{IN} \wedge L(a) = \text{UNDEC} \implies (a,b) \notin R$. Suppose $(b,a) \in R$, then by the second condition of Def. 5 $L(a) = \text{UNDEC} \implies (b,a) \notin R$. Contradiction!
Case 3. $L(b) = \text{UNDEC} \wedge L(a) = \text{IN} \implies (a,b) \notin R$. Suppose $(b,a) \in R$, then by the first condition of Def. 5 $L(a) = \text{IN} \implies (b,a) \notin R$. Contradiction!
Therefore if $(a,b) \in R \implies L \notin \sigma((S,R))$ then $(b,a) \in R \implies L \notin \sigma((S,R))$.
(*Backward implication*) Since a and b are arbitrary, the backward implication immediately follows from the permutation $a \leftrightarrow b$ in the forward implication.
Therefore if $(b,a) \in R \implies L \notin \sigma((S,R))$ then $(a,b) \in R \implies L \notin \sigma((S,R))$. □

We use Algorithm 1 to indicate an upper bound for complexity in solving the 3-valued σ-consistency problem. The algorithm begins with a full initial defeat relation such that all arguments defeat all other arguments and then prunes the defeats that are incompatible with the acceptability data. The algorithm involves two passes through the data: one to remove defeats inconsistent with the labellings, and a second to check the resulting defeat relation R is consistent with the labellings.

Algorithm 1 A pruning algorithm that returns a defeat relation R that is consistent with a set of labellings T and argument set A or indicates that no such R is possible.

```
procedure DEFEAT PRUNING($A, T$)
    $R \leftarrow \{(a,b), \forall a,b \in A\}$
    for $\forall L \in T$ do
        for $\forall a,b \in A$ do
            if $(L(b) = \text{IN} \wedge L(a) \in \{\text{IN}, \text{UNDEC}\}) \vee$
                    $(L(b) = \text{UNDEC} \wedge L(a) = \text{IN})$ then
                $R \leftarrow R \setminus (a,b)$
    for $\forall L \in T$ do
        for $\forall b \in A$ do
            if $L(b) = \text{UNDEC}$ then
                if $\nexists a \in A : ((a,b) \in R) \wedge (L(a) = \text{UNDEC})$ then
                    return failure
            if $L(b) = \text{OUT}$ then
                if $\nexists a \in A : ((a,b) \in R) \wedge (L(a) = \text{IN})$ then
                    return failure
    return $R$
```

We stress that Algorithm 1 is not the only method, and we certainly do not suggest it is optimal in terms of any performance metric, of addressing the 3-valued σ-consistency problem but, as Theorem 4 indicates, it shows the problem is significantly less complex to solve than its 2-valued peer. This result may appear unintuitive given the bijective mapping of 2-valued and 3-valued labellings in forward argumentation. The simple reason for the divergence in complexity between the two σ-consistency problems is due to the ambiguity of the NOTIN label which can be either OUT or UNDEC in the 3-valued approach. Theorem 3 shows that those defeats that are inconsistent with 3-valued data are direct consequences of the labels themselves, whereas in Theorem 1 we see that the labels of 2-valued data are insufficient to determine inconsistency and the existing defeat relation R must be examined also. This self-referential search process in finding a solution defeat relation R for the 2-valued σ-consistency problem is the cause of the additional complexity.

Theorem 4. (Defeat pruning for 3-valued σ-consistency problem) *The 3-valued σ-consistency problem can be solved by a defeat pruning algorithm with a time complexity of $O(n^2|T|)$ and a space complexity of $O(n^2)$ where n is the number of arguments and $|T|$ is the number of labellings.*

Proof. Let us define some R' that is the defeat relation output by Algorithm 1 upon receiving input T. There exist two cases for solving the 3-valued σ-consistency problem: (*Case 1 Forward implication*) Suppose there $\exists R$ that is consistent with T. Suppose R' is not consistent with T, then $\exists L \in T : (L \notin \sigma((S,R)))$. By the definition of R' it cannot be

that some inconsistent defeat is in R' and so there must be some essential defeat missing from R'. From Def. 5 it must be that $(\exists b_1 \in S : (L(b_1) = \text{UNDEC}) \wedge \forall c \in S : (L(c) = \text{UNDEC} \implies (c,b_1) \notin R')) \vee (\exists b_2 \in S : (L(b_2) = \text{OUT}) \wedge \forall a \in S : (L(a) = \text{IN} \implies (a,b_2) \notin R'))$. But since R exists then for any such $L(b_1) = \text{UNDEC}$ there is some appropriate $(c,b_1) \in R \setminus R'$, or for any $L(b_2) = \text{OUT}$ there is some appropriate $(a,b_2) \in R \setminus R'$. But by the definition of R' it must be that $R' \supseteq R$, contradiction! Therefore if there exists some R that is consistent with T then R' is also consistent with T.
(*Case 2 Backward implication*) Suppose there $\nexists R$ consistent with T. Then clearly R' is not consistent with T. Hence if $\nexists R$ consistent with T then R' is also not consistent with T. Therefore there $\exists R$ that is consistent with T iff R' is consistent with T.
To find R' it is necessary in the worst case to check all n^2 possible defeats for each $L \in T$. From Theorem 3 it is clear that for any defeat (a,b) to be evaluated for consistency with some $L \in T$ it is sufficient to simply check $L(a)$ and $L(b)$. Hence there are required at most $2n^2|T|$ operations required to produce R'. Similarly, once R' has been derived, there will be at most $2n^2|T|$ operations required to check that R' is consistent with T. Therefore the defeat pruning algorithm will find a solution defeat relation R' or prove that none exist in time complexity of $O(2n^2|T| + 2n^2|T|) = O(n^2|T|)$.
Finally, for each step in the process we need to store the current defeat set R, of size n^2. It follows that the space complexity is within $O(n^2)$. □

5. σ-consistency For Other Semantics

Throughout this paper the focus has been on complete argumentation semantics. However, turning our attention to the alternative semantics as originally presented in [10] allows us to informally outline some relevant results.

For σ-consistency under stable semantics, we can quickly identify that both the 2-valued and 3-valued problems reduce to a special case of the 3-valued problem where no labelling contains UNDEC labelled arguments. Explicitly for the 2-valued problem this means that all NOTIN labelled arguments are interpreted as labelled OUT. Therefore the problem is solvable in $O(n^2|T|)$ time and $O(n^2)$ space complexity.

For σ-consistency under preferred semantics, we conjecture that the problem is not in **NP** unless **coNP** = **P**. The verification of any solution defeat relation is achieved by verifying each labelling in the data set. As outlined in [9,12], the verification of any labelling under preferred semantics is **coNP**-complete. We observe the **coNP**-complete result is derived from the special instance of verifying the empty set is a preferred extension, hence the result pertains to both the 2-valued and 3-valued variants. Further, as discussed in [13], verification results for all the major semantics hold for both 2-valued and 3-valued data. This means that solving σ-consistency is likely to be a hard problem.

For σ-consistency under grounded semantics, it is easy to see that for both 2-valued and 3-valued acceptability data, the problem is within **NP**, since the verification of each labelling in $|T|$ is within **P** [12]. However, we strongly conjecture that both 2-valued and 3-valued σ-consistency problems under grounded semantics are in fact **NP**-complete. There is not room to demonstrate a full proof in the confines of this paper. However, a proof similar to that used for Theorem 2 can be constructed by reducing from the Monotone 3SAT problem such that the two types of clauses are reduced by r thus:
Case 1: $c_i = (a_i \vee b_i \vee c_i)$ and $r(c_i) = (S, L_{1i}, L_{2i})$, where $S = \{a_i, b_i, c_i, \omega\}$, $L_{1i} = \{L(\omega) = \text{IN}\}$, and for 2-valued (resp. 3-valued) σ-consistency we have $L_{2i} = \{L(a_i) =$

$\text{NOTIN}, L(b_i) = \text{NOTIN}, L(c_i) = \text{NOTIN}, L(\omega) = \text{NOTIN}\}$ (resp. $L_{2i} = \{L(a_i) = \text{UNDEC}, L(b_i) = \text{UNDEC}, L(c_i) = \text{UNDEC}, L(\omega) = \text{UNDEC}\}$);
Case 2: $c_i = (\neg a_i \vee \neg b_i \vee \neg c_i)$ and $r(c_i) = (S, L_{1i})$, where $S = \{a_i, b_i, c_i, \omega\}$, and for 2-valued (resp. 3-valued) σ-consistency we have $L_{1i} = \{L(a_i) = \text{NOTIN}, L(b_i) = \text{NOTIN}, L(c_i) = \text{NOTIN}, L(\omega) = \text{IN}\}$ (resp. $L_{2i} = \{L(a_i) = \text{OUT}, L(b_i) = \text{OUT}, L(c_i) = \text{OUT}, L(\omega) = \text{IN}\}$).

6. Related Work

Research on the topic of extension enforcement [2,3], is concerned with determining what additions could be made to a defeat relation in order to accommodate new extensions. However, there are notable departures from the direction pursued in this paper, such as requiring monotonic growth of the defeat relation, whereas we also allow reduction when solving for σ-consistency.

Argumentation realizability [4,11,17,20], extends beyond extension enforcement by removing the requirement of monotonic enlargement of the defeat relation R. Realizability requires that there exists a defeat relation that can express precisely the given set of interpretations (labellings or extensions), with no other interpretations expressible from the defeat relation. This assumption of completeness of the input extension/labelling set can be understood as a special case of σ-consistency where partial labellings pertaining to argument subgraphs are not permitted and the input labellings are exactly $\sigma((A,R))$. Interestingly, research into argumentation realizability has thus far encountered difficulty in determining a complexity class for complete semantics, remaining apparently unsolved despite its importance as a foundation for other semantics as previously discussed. Note that the complexity for realizability under complete semantics is conjectured to be **NP**-hard due to the association with MaxSat algorithms in deriving a solution.

Argumentation synthesis [18,19] develops the concept of realizability by relaxing the requirement for one-to-one mapping; the solution defeat set must satisfy a maximal number of argument labels. The approach further differentiates itself from realizability by accepting positive labels that are to be satisfied, but also negative labels that should not be satisfied. Argumentation synthesis is posed as an optimization problem that can accommodate noisy data sets, which will be in accordance with a wide set of real problems. Thus far in the literature, argumentation synthesis has only been applied to 2-valued problems (i.e. extension-based) and ignored partial labellings, unlike the general σ-consistency problems considered in this paper. As an optimization problem it requires its own complexity analysis appropriate for a Max-Sat search. Similar to realizability, complexity results for complete semantics have thus far been elusive albeit conjectured to belong to the **NP**-hard class of problems [19].

An alternative to argumentation synthesis for handling noisy data are the probability-based approaches [15,16,22] that do not overtly seek out minimising the number of misclassified errors as the singular goal. Whilst [15,16] use Bayesian inference and [22] uses the more elementary Kolmogorov's axioms, both focus on 2-valued argument acceptability data. It is notable that [15,16] suffer the problem of exponential complexity when determining their Bayesian calculations, since power sets of extension argument acceptabilities must be considered with a resulting combinatorial explosion. In contrast, [22] does not suffer from this same problem but has an altogether different dilemma in identifying from where the prior probabilities that are assigned to the argument rules are obtained, before these are mapped to the corresponding graph.

The most closely related research [21,23] examines the σ-consistency problem from the 2-valued and 3-valued perspectives but under grounded semantics. The complexity results from [21,23] claim that processing the 2-valued (resp. 3-valued) σ-consistency problem under grounded semantics is solvable in $O(n^2|T|)$ (resp. $O(n^3|T|)$) time (by our notation). These findings clearly disagree with our conjecture from Section 5 that both problems are **NP**-complete. We believe that the complexity results from [21,23] are incorrect and the error stems from neglecting the subset minimality of grounded semantics and the potential for the empty set to be the grounded labelling. A formal proof is forthcoming.

7. Concluding Remarks

We examined the computational complexity of the σ-consistency problem (where 'σ-consistency' refers to 'σ-consistency under complete semantics' throughout the paper) that determines whether a solution defeat relation exists that is wholly consistent with a set of argument acceptability labellings under the given semantics. The paper offers the following contributions.

- Introduced and defined the σ-consistency problem, which asks if a solution defeat relation exists that is consistent with a set of argument acceptability labellings.
- The 2-valued σ-consistency problem is proved to be **NP**-complete, and shown to have a space complexity within $O(n^2)$.
- The 3-valued σ-consistency problem is proved to be solvable by algorithms with a time complexity within $O(n^2|T|)$ and space complexity within $O(n^2)$.
- Provided informal complexity results for stable, preferred, and grounded semantics.
- The complexity results provide insight into the scaleability of algorithms designed to determine a defeat relation from argument acceptability data, suggesting a preference for sourcing 3-valued data to reduce the risk of solving high complexity problems.

Future work expanding the formal attention to other semantics, such as preferred or semi-stable, as well as to more advanced forms of argumentation, such as weighted or bipolar semantics, would also require rigorous complexity analysis of these forms in order to locate the expectations for relevant algorithms. We would also identify, as a fertile ground for exploration, the pursuit of theory underpinning the enumeration and/or counting of solution defeat relations under σ-consistency, as well as related research into the "quality" of solution defeat relations compared with the notion of a ground truth.

Acknowledgements

This work was partially supported by a PhD studentship from King's College London.

References

[1] V. Aleven. *Teaching case-based argumentation through a model and examples*. PhD thesis, University of Pittsburgh, 1997.

[2] Ringo Baumann. What does it take to enforce an argument? minimal change in abstract argumentation. In *ECAI*, volume 12, pages 127–132, 2012.

[3] Ringo Baumann and Gerhard Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. *COMMA*, 10:75–86, 2010.

[4] Ringo Baumann, Wolfgang Dvorák, Thomas Linsbichler, Hannes Strass, and Stefan Woltran. Compact argumentation frameworks. In *ECAI*, pages 69–74, 2014.

[5] Trevor JM Bench-Capon and Paul E Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[6] Martin Caminada. On the issue of reinstatement in argumentation. In *European Workshop on Logics in Artificial Intelligence*, pages 111–123, 2006.

[7] Martin WA Caminada and Dov M Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109, 2009.

[8] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 317–328. Springer, 2005.

[9] Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science*, 170(1-2):209–244, 1996.

[10] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[11] Paul E Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artificial Intelligence*, 228:153–178, 2015.

[12] Paul E Dunne and Michael Wooldridge. Complexity of abstract argumentation. In *Argumentation in artificial intelligence*, pages 85–104. Springer, 2009.

[13] Wolfgang Dvorák and Paul E Dunne. Computational problems in formal argumentation and their complexity. *Handbook of formal argumentation*, 4, 2018.

[14] Michael R Garey and David S Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[15] Hiroyuki Kido and Beishui Liao. A Bayesian approach to direct and inverse abstract argumentation problems. *arXiv preprint arXiv:1909.04319*, 2019.

[16] Hiroyuki Kido and Frank Zenker. Argument-based Bayesian estimation of attack graphs: A preliminary empirical analysis. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 523–532. Springer, 2017.

[17] T Linsbichler, J Puehrer, and H Strass. Characterizing realizability in abstract argumentation. In *Proceedings of the 16th International Workshop on Non-monotonic reasoning*, 2016.

[18] Andreas Niskanen, Daniel Neugebauer, Matti Järvisalo, and Jörg Rothe. Deciding acceptance in incomplete argumentation frameworks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2942–2949, 2020.

[19] Andreas Niskanen, Johannes Wallner, and Matti Järvisalo. Synthesizing argumentation frameworks from examples. *Journal of Artificial Intelligence Research*, 66:503–554, 2019.

[20] Jörg Pührer. Realizability of three-valued semantics for abstract dialectical frameworks. *Artificial Intelligence*, 278:103198, 2020.

[21] Régis Riveret. On learning abstract argumentation graphs from bivalent statement labellings. In *28th International Conference on Tools with Artificial Intelligence*, pages 190–195. IEEE, 2016.

[22] Régis Riveret, Pietro Baroni, Yang Gao, Guido Governatori, Antonino Rotolo, and Giovanni Sartor. A labelling framework for probabilistic argumentation. *Annals of Mathematics and Artificial Intelligence*, pages 1–51, 2018.

[23] Régis Riveret and Guido Governatori. On learning attacks in probabilistic abstract argumentation. In *AAMAS*, pages 653–661, 2016.

[24] Yining Wu, Martin Caminada, and Mikotaj Podlaszewski. A labelling-based justification status of arguments. *Studies in Logic*, 3(4):12–29, 2010.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220159

Stability and Relevance in Incomplete Argumentation Frameworks

Daphne ODEKERKEN [a,b,1], AnneMarie BORG [a] and Floris BEX [a,c]
[a] *Department of Information and Computing Sciences, Utrecht University*
[b] *National Police Lab AI, Netherlands Police*
[c] *Tilburg Institute for Law, Technology and Society, Tilburg University*
ORCiD ID: Daphne Odekerken https://orcid.org/0000-0003-0285-0706, AnneMarie Borg https://orcid.org/0000-0002-7204-6046, Floris Bex https://orcid.org/0000-0002-5699-9656

Abstract. We explore the computational complexity of stability and relevance in incomplete argumentation frameworks (IAFs), abstract argumentation frameworks that encode qualitative uncertainty by distinguishing between certain and uncertain arguments and attacks. IAFs can be specified by, e.g., making uncertain arguments or attacks certain; the justification status of arguments in an IAF is determined on the basis of the certain arguments and attacks. An argument is *stable* if its justification status is the same in all specifications of the IAF. For arguments that are not stable in an IAF, the *relevance* problem is of interest: which uncertain arguments or attacks should be investigated for the argument to become stable? We redefine stability and define relevance for IAFs and study their complexity.

Keywords. Incomplete argumentation frameworks, stability, relevance, complexity

1. Introduction

Computational argumentation is an important research field in artificial intelligence, concerning reasoning with incomplete or inconsistent information [1]. A central concept are argumentation frameworks (AFs): a set of arguments and an attack relation between them [2]. Given an AF and so-called semantics, one can determine extensions: sets of arguments that can collectively be considered to be acceptable. Based on these extensions, each argument has at least one justification status (in terms of e.g. labels like IN, OUT and UNDEC). However, in practice, argumentation is a dynamic process in which not all arguments may be known in advance. Incomplete argumentation frameworks (IAFs) are designed to handle this dynamic process by extending AFs to allow for both certain and uncertain arguments and attacks [3,4,5,6]. In this paper, we study two problems in IAFs and their complexity for various semantics: stability and relevance.

Detecting stability was initially introduced for the ASPIC$^+$ framework in [7] and subsequently studied for structured and abstract argumentation settings in [8,6]. Informally, an argument is stable if more information cannot change its justification status.

[1] Corresponding Author; E-mail: d.odekerken@uu.nl

Stability detection has practical applications, for instance as a termination criterion for argumentative dialogue agents: in the agent architecture for inquiry proposed in [8], stability detection prevents the agent from asking unnecessary questions. In addition, [6] proposes an application of stability detection in negotiating agents, to recognise situations in which an agent should stop negotiating and accept its opponent's offer.

For situations in which the argument of interest is not stable in the given IAF, a natural question would be: which uncertainties should we resolve in order to reach a point where the argument is stable? In other words: which uncertain arguments or attacks are still relevant for the justification status? Adding relevance to an inquiry/negotiation process ensures that the questions that are asked contribute to reaching stability.

The contribution of this paper is the extensive study of both stability and relevance in the context of IAFs. Specifically, first we (re)define stability on IAFs, considering not only IN, but also OUT and UNDEC justification statuses. This results in a more fine-grained notion of stability than an earlier definition in [6]. Second, we present precise complexity results for stability of all these justification statuses in grounded, complete, stable and preferred semantics, refining preliminary results in [6]. Third, we define a notion of relevance in the context of reaching stability in IAFs. Finally, we present preliminary complexity results for the introduced relevance detection problem.

The paper is structured as follows. In Section 2, we provide the necessary preliminaries. In Section 3, we study the complexity of identifying the justification status of an argument and use these results in our complexity analysis of the stability problem. We then introduce the relevance problem for IAFs in Section 4 and provide complexity results. Related work is discussed in Section 5; we conclude in Section 6.

2. Preliminaries

In this section, we recall the most important notions from abstract argumentation and the considered semantics [2] as well as incomplete argumentation frameworks [3,4,5,6] and their specifications. Finally, we give a brief introduction to the polynomial hierarchy, which is required for our complexity study.

2.1. Argumentation frameworks and semantics

An argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$ (AF) consists of a finite set $\mathcal{A}$ of arguments and a binary attack relation $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ on them, where $(A,B) \in \mathcal{R}$ indicates that argument A attacks argument B. The evaluation of arguments is done using the semantics of [2].

Definition 1 (Extension-based semantics)**.** *Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and $S \subseteq \mathcal{A}$. Then:*

- *S is **conflict-free** iff for each $X, Y \in S : (X,Y) \notin \mathcal{R}$;*
- *$X \in \mathcal{A}$ is **acceptable with respect to** S iff for each $Y \in \mathcal{A}$ such that $(Y,X) \in \mathcal{R}$, there is a $Z \in S$ such that $(Z,Y) \in \mathcal{R}$;*
- *S is an **admissible set** iff S is conflict free and $X \in S$ implies that X is acceptable with respect to S;*
- *S is a **complete extension** (CP) iff S is admissible and for each X: if $X \in \mathcal{A}$ is acceptable with respect to S then $X \in S$;*
- *S is a **preferred extension** (PR) iff it is the set inclusion maximal admissible set;*

- *S is the **grounded extension** (GR) iff it is the set inclusion minimal complete extension; and*
- *S is a **stable extension** (ST) iff it is complete and attacks all the arguments in $\mathcal{A} \setminus S$.*

2.2. Incomplete argumentation frameworks

Incomplete argumentation frameworks (IAFs) are an extension to AFs, initially proposed as partial AFs in [3] and later studied as IAFs in e.g. [4,5,6]. In an IAF, the set of arguments and attacks is split into two disjoint parts: a certain part ($\mathcal{A}$ and $\mathcal{R}$) and an uncertain part ($\mathcal{A}^?$ and $\mathcal{R}^?$). For the uncertain elements, it is not known whether they are part of the argumentation framework or not. They may be added in the future, for example because more information is acquired in an inquiry dialogue, or removed, for example because after investigation, this element turned out not to be present in the given setting.

Definition 2 (Incomplete argumentation framework)**.** *An incomplete argumentation framework is a tuple $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, where $\mathcal{A} \cap \mathcal{A}^? = \emptyset$, $\mathcal{R} \cap \mathcal{R}^? = \emptyset$ and:*

- *$\mathcal{A}$ is the set of certain arguments;*
- *$\mathcal{A}^?$ is the set of uncertain arguments;*
- *$\mathcal{R} \subseteq (\mathcal{A} \cup \mathcal{A}^?) \times (\mathcal{A} \cup \mathcal{A}^?)$ is the certain attack relation; and*
- *$\mathcal{R}^? \subseteq (\mathcal{A} \cup \mathcal{A}^?) \times (\mathcal{A} \cup \mathcal{A}^?)$ is the uncertain attack relation.*

An IAF can be *specified* by obtaining more information about the uncertain part.

Definition 3 (Specification)**.** *Given an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, a specification is an IAF $\mathcal{I}' = \langle \mathcal{A}', \mathcal{A}^{?\prime}, \mathcal{R}', \mathcal{R}^{?\prime} \rangle$, where:*

- *$\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{A} \cup \mathcal{A}^?$;*
- *$\mathcal{R} \subseteq \mathcal{R}' \subseteq \mathcal{R} \cup \mathcal{R}^?$;*
- *$\mathcal{A}^{?\prime} \subseteq \mathcal{A}^?$;*
- *$\mathcal{R}^{?\prime} \subseteq \mathcal{R}^?$.*

We denote all possible specifications for $\mathcal{I}$ by $F(\mathcal{I})$. Note that $\mathcal{A}' \cap \mathcal{A}^{?\prime} = \emptyset$ and $\mathcal{R}' \cap \mathcal{R}^{?\prime} = \emptyset$ because $\mathcal{I}'$ is an IAF.

Since the semantics of [2] are defined on AFs, we define the certain projection of an IAF, which is an AF consisting of only the IAF's certain arguments and attacks.

Definition 4 (Certain projection)**.** *Given an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, the certain projection is the argumentation framework $AF = \langle \mathcal{A}, \mathcal{R} \cap (\mathcal{A} \times \mathcal{A}) \rangle$.*

Note that our definition of specification is similar to the notion of completion used in related work on IAFs [4,5]. Intuitively, a completion is a certain projection of a specification and is therefore not suitable for keeping track of the uncertain elements. Since the development of uncertain elements is essential for defining and studying the relevance problem (see Section 4), we use the notion of specification rather than completion.

2.3. The polynomial hierarchy

The polynomial hierarchy [9] is a hierarchy of complexity classes above NP defined using oracle machines, i.e. Turing machines that are allowed to call a subroutine (oracle), deciding some fixed problem in constant time. For a class of decision problems $\mathcal{C}$ and a class $\mathcal{X}$ defined by resource bounds, $\mathcal{X}^{\mathcal{C}}$ denotes the class of problems decidable on a Turing machine with a resource bound given by $\mathcal{X}$ and an oracle for a problem in $\mathcal{C}$.

Based on these notions, the sets Σ_k^p and Π_k^p are defined as follows: $\Sigma_0^p = \Pi_0^p = \Delta_0^p = P$, $\Sigma_{k+1}^p = \mathrm{NP}^{\Sigma_k^p}$ and $\Pi_{k+1}^p = \mathrm{CoNP}^{\Sigma_k^p}$. The canonical complete problem for Σ_k^p is k-QBF, which is the problem of deciding whether the quantified boolean formula with k alternating quantifiers, starting with an existential quantifier, is true for a formula Φ – for example, deciding if $\exists X \text{s.t.}\ \forall Y : \Phi[X,Y] = \text{True}$ is Σ_2^p-complete. The complement of a k-QBF problem, denoted by co-k-QBF, is complete for Π_k^p.

3. Justification status and stability

In order to define and study stability and relevance, we need a definition of justification status. Given an AF $\langle \mathcal{A}, \mathcal{R} \rangle$, an argument A and a semantics σ, A's justification status can be determined by either considering all σ-extensions (sceptical) or at least one σ-extension of the AF (credulous). In this context, an argument can be IN (part of all/some σ-extensions); OUT (attacked by all/some σ-extensions), or UNDEC (otherwise) [10].

Definition 5 (Argument justification status)**.** *Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework and σ some semantics in $\{\text{GR}, \text{CP}, \text{PR}, \text{ST}\}$. Let A be some argument in $\mathcal{A}$.*

- *A is σ-sceptical-*IN *(resp. σ-credulous-*IN*) iff A belongs to each (resp. some) σ-extension of AF;*
- *A is σ-sceptical-*OUT *(resp. σ-credulous-*OUT*) iff for each (resp. some) σ-extension S of AF, A is attacked by some argument in S;*
- *A is σ-sceptical-*UNDEC *(resp. σ-credulous-*UNDEC*) iff for each (resp. some) σ-extension of AF, A is not in S and not attacked by any argument in S.*

The justification statuses that we consider in this paper are $\{\text{GR}, \text{CP}, \text{PR}, \text{ST}\} \times \{\text{sceptical}, \text{credulous}\} \times \{\text{IN}, \text{OUT}, \text{UNDEC}\}$. Based on these justification statuses, we can now define stability:

Definition 6 (Stability on IAFs)**.** *Given an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, a certain argument $A \in \mathcal{A}$ and some justification status j, A is* stable-j *w.r.t. $\mathcal{I}$ iff for each specification $\mathcal{I}' = \langle \mathcal{A}', \mathcal{A}^{?\prime}, \mathcal{R}', \mathcal{R}^{?\prime} \rangle$ in $F(\mathcal{I})$, A is j w.r.t. the certain projection of $\mathcal{I}'$.*

Note that whereas the sceptical stability variants are mutually exclusive, this does not apply for the credulous stability variants, where an argument may have multiple stability statuses at the same time. Consider the following example:

Example 1. *Let $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$ be an incomplete argumentation framework with $\mathcal{A} = \{A, B\}$, $\mathcal{A}^? = \mathcal{R}^? = \emptyset$, $\mathcal{R} = \{(A,B),(B,A)\}$; note that $F(\mathcal{I}) = \{\mathcal{I}\}$. Both A and B are stable-*CP*-credulous-*IN*, stable-*CP*-credulous-*OUT *and stable-*CP*-credulous-*UNDEC*. For $\sigma \in \{\text{ST}, \text{PR}\}$, both A and B are stable-σ-credulous-*IN *and stable-σ-credulous-*OUT*. On the other hand, for semantics $\sigma \in \{\text{CP}, \text{ST}, \text{PR}\}$, A and B are not stable-σ-sceptical-*IN, -OUT *or* -UNDEC.

An alternative definition of stability on IAFs has been proposed before in [6], but this definition only takes the IN status of arguments into account. Since other justification statuses are not studied, it is not possible to distinguish for example situations in which the justification status of an argument is UNDEC in each certain projection from situations where the justification status is always either OUT or UNDEC. Our definition of stability on IAFs is more fine-grained as it also includes OUT- and UNDEC-stability. In addition, we will provide precise complexity results, refining preliminary complexity bounds from [6].

We formulate the identification of justification and stability statuses as decision problems:

j-JUSTIFICATION

Given: An argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$ and an argument $A \in \mathcal{A}$

Question: Does A's justification status in $\langle \mathcal{A}, \mathcal{R} \rangle$ equal j?

j-STABILITY

Given: An incomplete argumentation framework $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, a justification status j and an argument $A \in \mathcal{A}$

Question: Does A's stability status w.r.t. $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$ equal stable-j?

For some variants of the stability problem, we can directly use complexity results from earlier work, in particular the results on necessary sceptical and credulous acceptance presented in [5]. Specifically, for a given semantics $\sigma \in \{\text{GR}, \text{CP}, \text{PR}, \text{ST}\}$, an argument is stable-$\sigma$-sceptical-IN iff it is necessary sceptically accepted w.r.t. the corresponding IAF; similarly, the set of stable-σ-credulous-OUT arguments coincides with the necessary credulously accepted arguments (see Section 5 for a discussion). See Table 1 for an overview of these results.

σ	c/s	status	JUSTIFICATION	STABILITY	RELEVANCE
ST	c	IN/OUT	NP-c [11,12]	Π_2^p-c [5]	
ST	c	UNDEC	**Trivial (no)**	**Trivial (no)**	
ST	s	IN/OUT	CoNP-c [11]	CoNP-c (Π_2^p-c) [5]	
ST	s	UNDEC	CoNP-c [11] (**Trivial (no)**)	**CoNP-c** (**Trivial (no)**)	
CP	c	IN/OUT	NP-c [11,12]	Π_2^p-c [5]	
CP	c	UNDEC	**P-c**	**CoNP-c**	
CP	s	IN/OUT	P-c [2]	CoNP-c [5]	**NP-c**
CP	s	UNDEC	**CoNP-c**	**CoNP-c**	
GR	c	IN/OUT	P-c [2]	CoNP-c [5]	**NP-c**
GR	c	UNDEC	**P-c**	**CoNP-c**	
GR	s	IN/OUT	P-c [2]	CoNP-c [5]	**NP-c**
GR	s	UNDEC	**P-c**	**CoNP-c**	
PR	c	IN/OUT	NP-c [11,12]	Π_2^p-c [5]	
PR	c	UNDEC	**Σ_2^p-c**	**Π_3^p-c**	
PR	s	IN/OUT	Π_2^p-c [13]	Π_2^p-c [5]	
PR	s	UNDEC	**CoNP-c**	**CoNP-c**	

Table 1. Overview of all complexity results related to this paper. If a reference is specified, this complexity result is trivial from an earlier result in the literature. New results are printed bold; their proofs can be found in the appendix. Results for ST-sceptical-existence justification and stability are given in parentheses.

The complexity of OUT-STABILITY has not been studied, but can be derived from the complexity of identifying IN-STABILITY by a reduction from the corresponding IN-JUSTIFICATION problems. In the following lemma, we show that these complexities are the same for each of the semantics considered in this paper:[2]

Lemma 1. *For any given $\sigma \in \{\text{GR}, \text{CP}, \text{PR}, \text{ST}\}$ and $c \in \{sceptical, credulous\}$, the complexity of σ-c-OUT-STABILITY equals the complexity of σ-c-IN-STABILITY.*

A similar result exists for IN-JUSTIFICATION and OUT-JUSTIFICATION. The results for UNDEC-JUSTIFICATION can be found in Table 1 and the appendix. The complexity of UNDEC-STABILITY cannot be derived in a general way from e.g. IN-STABILITY: the approach depends on the chosen semantics. We can however provide a general upper bound based on the complexity of UNDEC-JUSTIFICATION:

Proposition 1 (Upper bound j-STABILITY)**.** *Given an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, a certain argument $A \in \mathcal{A}$ and a justification status j, if the complexity of j-JUSTIFICATION in $\langle \mathcal{A}, \mathcal{R} \rangle$ is $\mathcal{C}$, the j-STABILITY problem given A and $\mathcal{I}$ is in CoNP$^{\mathcal{C}}$.*

Proof. In a negative instance $(\mathcal{I}, A)$ of j-STABILITY, there is some $\mathcal{I}' \in F(\mathcal{I})$ such that A is not j in the certain projection of $\mathcal{I}'$. A polynomial-size certificate for this instance would for example be a specification $\mathcal{I}' = \langle \mathcal{A}', \emptyset, \mathcal{R}', \emptyset \rangle$ in $F(\mathcal{I})$ such that A is **not** j in $AF' = \langle \mathcal{A}', \mathcal{R}' \cap (\mathcal{A}' \times \mathcal{A}') \rangle$. Verification that $\mathcal{I}' \in F(\mathcal{I})$ can be done in polynomial time; the j-justification status check of A in AF' is done by calling a $\mathcal{C}$ oracle. □

Note that the complexity of identifying the stability or justification status for UNDEC statuses may be higher or lower than the complexity of identifying the IN or OUT status. For example, for ST semantics the credulous JUSTIFICATION and STABILITY identification of the UNDEC status is trivial: under ST semantics arguments are either in the extension or attacked by the extension, but identifying the IN/OUT status is NP (JUSTIFICATION) and even in Π_2^p (STABILITY). On the other hand, PR-credulous-UNDEC-JUSTIFICATION is on a higher level in the polynomial hierarchy than PR-credulous-IN-JUSTIFICATION: while PR-credulous-IN-JUSTIFICATION is in NP (since verification of a positive instance can be done in polynomial time, given an admissible set containing the argument as a certificate), PR-credulous-UNDEC-JUSTIFICATION is Σ_2^p-hard as it can be reduced from 2-QBF.

For other semantics, the complexity of UNDEC-STABILITY does not differ from the IN-STABILITY complexity. This is for example the case for GR semantics and CP semantics for the sceptical justification status. In order to prove this, we next give a reduction from UNSAT (co-1-QBF), which is based on the reduction from [5, Definition 14] and illustrated in the black part of Figure 1.

Definition 7 (Reduction)**.** *Let (ϕ, X) be an instance of 1-QBF or co-1-QBF and let $\phi = \bigwedge_i c_i$, where $c_i = \bigvee_j \alpha_j$ for each clause c_i in ϕ and α_j are the literals over X that occur in clause c_i. We define the corresponding IAF for this instance as $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \emptyset \rangle$, where:*

- $\mathcal{A} = \{x_i, \overline{x_i} \mid x_i \in X\} \cup \{\overline{c_i} \mid c_i \in \phi\} \cup \{\phi, \overline{\phi}\}$;
- $\mathcal{A}^? = \{g_i \mid x_i \in X\}$;

[2] Proofs of omitted results in the remainder of the paper can be found in the online appendix https://www.uu.nl/onderzoek/ai-labs/nationaal-politielab-ai/stability-relevance-iafs.

- $\mathcal{R} = \{(g_i, \overline{x_i}) \mid x_i \in X\} \cup \{(\overline{x_i}, x_i) \mid x_i \in X\} \cup \{(x_k, \overline{c_i}) \mid x_k \in c_i\} \cup \{(\overline{x_k}, \overline{c_i}) \mid \neg x_k \in c_i\} \cup \{(\overline{c_i}, \phi) \mid c_i \in \phi\} \cup \{(\phi, \overline{\phi}), (\overline{\phi}, \overline{\phi})\}$.

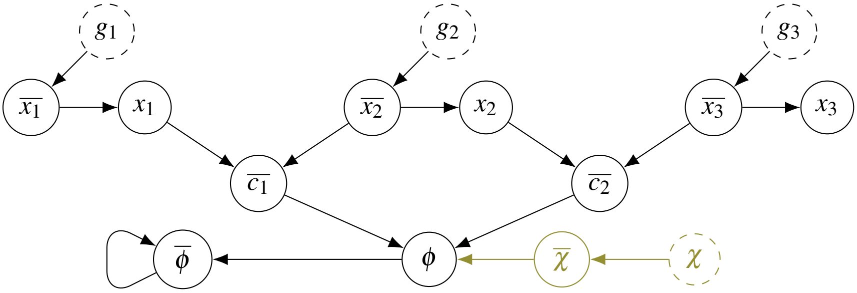


Figure 1. Visualisation of the IAF created for the clauses $c_1 = x_1 \vee \neg x_2$ and $c_2 = x_2 \vee \neg x_3$ using the reductions of Definition 7, only the black parts of the figure and Definition 11, also including the gold/gray part. We use this reduction for GR, CP and sceptical PR semantics.

The reduction is used in the proof of the following proposition.

Proposition 2. *GR-sceptical-UNDEC-STABILITY, GR-credulous-UNDEC-STABILITY and CP-credulous-UNDEC-STABILITY are CoNP-complete.*

Proof sketch. Let (ϕ, X) be a co-1-QBF (UNSAT) instance and let $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$ be the IAF according to Definition 7. As GR semantics results in a single extension, which is the intersection of all CP extensions, the problems GR-sceptical-UNDEC-STABILITY, GR-credulous-UNDEC-STABILITY and CP-credulous-UNDEC-STABILITY coincide. The argument $\overline{\phi}$ can only be stable-UNDEC if for each specification of $\mathcal{I}$, the argument ϕ is OUT, which can only be the case if there is at least one clause in ϕ such that the corresponding argument for $\overline{c_i}$ is IN. Thus the following items are equivalent:

1. (ϕ, X) is a positive UNSAT instance;
2. the argument for $\overline{\phi}$ is stable-GR-sceptical-UNDEC in $\mathcal{I}$;
3. the argument for $\overline{\phi}$ is stable-GR-credulous-UNDEC in $\mathcal{I}$;
4. the argument for $\overline{\phi}$ is stable-CP-credulous-UNDEC in $\mathcal{I}$.

CoNP-hardness from UNDEC-STABILITY follows from CoNP-hardness of UNSAT. From Proposition 1 and the fact that the corresponding JUSTIFICATION problems are in P, we conclude CoNP-completeness. □

In the following proposition, we consider the sceptical variants of the UNDEC-STABILITY under CP and PR semantics.

Proposition 3. *CP-sceptical-UNDEC-STABILITY and PR-sceptical-UNDEC-STABILITY are CoNP-complete.*

Proof sketch. CP-sceptical-UNDEC-STABILITY and PR-sceptical-UNDEC-STABILITY are in CoNP, since negative instances $(\mathcal{I}, A)$ can be verified in polynomial time given a certificate (AF', S) such that $\mathcal{I}' \in F(\mathcal{I})$, AF' is the certain projection of $\mathcal{I}'$, $A \in \mathcal{A}$ and S is an admissible set of AF' containing either A itself or an argument attacking A. For hardness, note that the corresponding (CoNP-hard) JUSTIFICATION problems can be reduced to these STABILITY problems, leaving the uncertain part empty. □

Proposition 4 states that the PR-credulous-UNDEC-STABILITY problem is on the third level of the polynomial hierarchy. For the proof, we refer to the appendix.

Proposition 4. PR-*credulous*-UNDEC-STABILITY *is* Π_3^p*-complete.*

Finally, we consider UNDEC-STABILITY under ST semantics. Recall from Definition 1 that for each ST extension S of an AF, each argument in AF is either in S or attacked by some argument in S. Consequently, an argument can only be stable-ST-sceptical-UNDEC if the AF has no ST extension. We use this property in the following proposition:

Proposition 5. ST-*sceptical*-UNDEC-STABILITY *is CoNP-complete.*

Proof sketch. The problem is in CoNP, as a no-instance $(\mathcal{I},A)$ can be verified in polynomial time given a certificate (AF', S) such that $\mathcal{I}' \in F(\mathcal{I})$, AF' is the certain projection of $\mathcal{I}'$ and S is a ST extension of AF'. If S is a ST extension then each argument in $\mathcal{A}$ is either in S or attacked by S; therefore no argument can be stable-ST-sceptical-UNDEC w.r.t. $\mathcal{I}$. For hardness, we can reduce from the CoNP-complete problem ST-sceptical-UNDEC-JUSTIFICATION. □

This result for sceptical acceptance may feel counterintuitive. Therefore, we consider an alternative version of justification and stability (based on [14]) for ST semantics, in which it is assumed that a ST extension exists.

Definition 8 (Sceptical-existent justification)**.** *Given an argumentation framework* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$*, argument* $A \in \mathcal{A}$ *and label* LAB $\in \{$IN, OUT, UNDEC$\}$*,* A *is* ST-*sceptical-existent*-LAB *w.r.t. AF iff AF has at least one* ST *extension and A is* ST-*sceptical*-LAB *in AF.*

As no AF has a ST extension S in which some argument is neither in S, nor attacked by any argument in S, ST-sceptical-existent-UNDEC-STABILITY is False for all inputs. The same holds for ST-credulous-UNDEC-STABILITY, so these problems are trivial:

Proposition 6. ST-*credulous*-UNDEC-STABILITY *and* ST-*sceptical-existent*-UNDEC-STABILITY *are trivial.*

4. Relevance

For IAFs in which a given argument is not stable, a natural follow-up question would be which uncertainties should be resolved in order to reach a point where the argument is stable: these uncertainties are *relevant* to investigate in the given IAF. In this section, we will define the problem of relevance and study its complexity. First, we give some intuition on the notion of relevance in the context of stability in the following example.

Example 2. *Let* $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$ *be the IAF illustrated in Figure 2, where only arguments A and C are certain and let j be* GR-*sceptical*-IN*; suppose that we want to know if argument A is j-stable. A is not j-stable in* $\mathcal{I}$*, but it will become j-stable in each* $\mathcal{I}' \in F(\mathcal{I})$ *such that* $\mathcal{I}' = \langle \mathcal{A}', \mathcal{A}^{?\prime}, \mathcal{R}', \mathcal{R}^{?\prime} \rangle$ *and* $D \notin \mathcal{A}' \cup \mathcal{A}^{?\prime}$ *or* $E \in \mathcal{A}'$*. Therefore, it would be relevant to investigate if D can be removed from the uncertain arguments and/or if E can be added to the certain arguments. Note that investigation of B does not contribute towards a stable situation and therefore would not be relevant.*

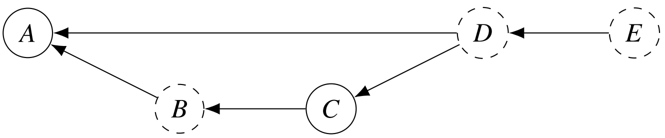


Figure 2. Visualisation of IAF $\langle\mathcal{A},\mathcal{A}^?,\mathcal{R},\mathcal{R}^?\rangle$ used to illustrate the definition of relevance. In this figure, certain arguments in $\mathcal{A}$ are depicted as nodes with solid borders (i.e. A and C), while uncertain arguments in $\mathcal{A}^?$ have dashed borders (i.e. B, D and E). The arrows between them correspond to attacks in $\mathcal{R}$.

Before proceeding to a formal definition of relevance that matches the intuitions in the example above in Definition 10, we define the notion of minimal stable specifications.

Definition 9 (Minimal stable-j specification)**.** *Given an IAF $\mathcal{I} = \langle\mathcal{A},\mathcal{A}^?,\mathcal{R},\mathcal{R}^?\rangle$, a certain argument $A \in \mathcal{A}$ and a justification status j, a minimal stable-j specification for A w.r.t. $\mathcal{I}$ is a specification $\mathcal{I}'$ in $F(\mathcal{I})$ such that A is stable-j in $\mathcal{I}'$ and there is no specification $\mathcal{I}''$ in $F(\mathcal{I})$ such that A is stable-j in $\mathcal{I}''$, $\mathcal{I}'' \neq \mathcal{I}'$ and $\mathcal{I}' \in F(\mathcal{I}'')$.*

Intuitively, the minimal stable-j specification for A is a specification in which A is stable-j, while A would not be stable-j in any specification with more uncertain elements.

Example 3. *Reconsider the IAF $\mathcal{I} = \langle\mathcal{A},\mathcal{A}^?,\mathcal{R},\mathcal{R}^?\rangle$ from Example 2:*

- *There are two minimal stable-*GR*-sceptical-*IN *specifications for A w.r.t. $\mathcal{I}$. These are $\langle\{A,C,E\},\{B,D\},\mathcal{R},\mathcal{R}^?\rangle$ and $\langle\{A,C\},\{B,E\},\mathcal{R},\mathcal{R}^?\rangle$;*
- *There are three specifications for A w.r.t. $\mathcal{I}$ for which A becomes stable-*GR*-sceptical-*OUT*: $\mathcal{I}_1 = \langle\{A,C,D\},\{B\},\mathcal{R},\mathcal{R}^?\rangle$, $\mathcal{I}_2 = \langle\{A,C,D\},\emptyset,\mathcal{R},\mathcal{R}^?\rangle$ and $\mathcal{I}_3 = \langle\{A,B,C,D\},\emptyset,\mathcal{R},\mathcal{R}^?\rangle$. Only $\mathcal{I}_1$ is minimal, because both $\mathcal{I}_2$ and $\mathcal{I}_3$ are in $F(\mathcal{I}_1)$;*
- *None of the specifications is minimal stable-*GR*-sceptical-*UNDEC.

Using the notion of minimal stable-j specifications, we can now define j-RELEVANCE.

Definition 10 (j-RELEVANCE)**.** *Given an IAF $\mathcal{I} = \langle\mathcal{A},\mathcal{A}^?,\mathcal{R},\mathcal{R}^?\rangle$, an argument $A \in \mathcal{A}$, an uncertain argument or attack $U \in \mathcal{A}^? \cup \mathcal{R}^?$ and a justification status j,*

- *Addition of U is j-relevant for A w.r.t. $\mathcal{I}$ iff there is a minimal stable-j specification $\mathcal{I}' = \langle\mathcal{A}',\mathcal{A}^{?\prime},\mathcal{R}',\mathcal{R}^{?\prime}\rangle$ for A w.r.t. $\mathcal{I}$ such that $U \in \mathcal{A}' \cup \mathcal{R}'$; and*
- *Removal of U is j-relevant for A w.r.t. $\mathcal{I}$ iff there is a minimal stable-j specification $\mathcal{I}' = \langle\mathcal{A}',\mathcal{A}^{?\prime},\mathcal{R}',\mathcal{R}^{?\prime}\rangle$ for A w.r.t. $\mathcal{I}$ such that $U \notin \mathcal{A}' \cup \mathcal{A}^{?\prime} \cup \mathcal{R}' \cup \mathcal{R}^{?\prime}$.*

In other words, addition of an uncertain element U is j-relevant if a minimal stable-j specification can be reached by moving U from the certain to the uncertain part of the IAF $\mathcal{I}$; and removal of U is j-relevant if completely removing U from $\mathcal{I}$, possibly in combination with other actions, leads to a minimal stable-j specification.

Example 4. *Recall from Example 3 that only $\mathcal{I}_1 = \langle\{A,C,D\},\{B\},\mathcal{R},\mathcal{R}^?\rangle$ is a minimal stable-*GR*-sceptical-*OUT *specification for A w.r.t. $\mathcal{I}$. Therefore, only addition of D and removal of E are* GR*-sceptical-*OUT*-relevant for A w.r.t. $\mathcal{I}$.*

Like for justification and stability status, we formulate the identification of j-RELEVANCE as a decision problem:

j-RELEVANCE of action **a**	
Given:	An incomplete argumentation framework $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, a justification status j, an action $\mathbf{a} \in \{addition, removal\}$, an argument $A \in \mathcal{A}$ and an uncertain argument or attack $U \in \mathcal{A}^? \cup \mathcal{R}^?$.
Question:	Is **a** of U j-relevant for A w.r.t. $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$?

The following lemma, proven in the appendix, shows that the relevance of adding or removing an uncertain element can be validated by checking the justification status of the certain projections of two particular future specifications. This property will be useful for proving an upper bound on *j*-RELEVANCE.

Lemma 2. *Given an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, a certain argument $A \in \mathcal{A}$ and a justification status j:*

1. *For each $U \in \mathcal{A}^?$, addition of U is j-relevant for A w.r.t. $\mathcal{I}$ iff there exists some $\mathcal{I}' = \langle \mathcal{A}', \{U\}, \mathcal{R}', \emptyset \rangle \in F(\mathcal{I})$ such that A is **not** j in the certain projection of $\mathcal{I}'$, while A is j in the certain projection of $\langle \mathcal{A}' \cup \{U\}, \emptyset, \mathcal{R}', \emptyset \rangle$.*
2. *For each $U \in \mathcal{R}^?$, addition of U is j-relevant for A w.r.t. $\mathcal{I}$ iff there exists some $\mathcal{I}' = \langle \mathcal{A}', \emptyset, \mathcal{R}', \{U\} \rangle \in F(\mathcal{I})$ such that A is **not** j in the certain projection of $\mathcal{I}'$, while A is j in the certain projection of $\langle \mathcal{A}', \emptyset, \mathcal{R}' \cup \{U\}, \emptyset \rangle$.*
3. *For each $U \in \mathcal{A}^?$, removal of U is j-relevant for A w.r.t. $\mathcal{I}$ iff there exists some $\mathcal{I}' = \langle \mathcal{A}', \{U\}, \mathcal{R}', \emptyset \rangle \in F(\mathcal{I})$ such that A is j in the certain projection of $\mathcal{I}'$, while A is **not** j in the certain projection of $\langle \mathcal{A}' \cup \{U\}, \emptyset, \mathcal{R}', \emptyset \rangle$.*
4. *For each $U \in \mathcal{R}^?$, removal of U is j-relevant for A w.r.t. $\mathcal{I}$ iff there exists some $\mathcal{I}' = \langle \mathcal{A}', \{U\}, \mathcal{R}', \emptyset \rangle \in F(\mathcal{I})$ such that A is j in the certain projection of $\mathcal{I}'$, while A is **not** j in the certain projection of $\langle \mathcal{A}', \emptyset, \mathcal{R}' \cup \{U\}, \emptyset \rangle$.*

In the following proposition, we use the results from Lemma 2 to prove a general upper bound on the complexity of *j*-RELEVANCE.

Proposition 7 (Upper bound *j*-RELEVANCE)**.** *Given an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, a certain argument $A \in \mathcal{A}$, an uncertain argument or attack $U \in \mathcal{A}^? \cup \mathcal{R}^?$ and a justification status j, if the complexity of deciding j's justification status in $\langle \mathcal{A}, \mathcal{R} \rangle$ is $\mathcal{C}$, then an upper bound on the problem of deciding if addition and/or removal of U is j-relevant for A w.r.t. $\mathcal{I}$ is $NP^{\mathcal{C}}$.*

In order to prove a lower bound of *j*-RELEVANCE for GR and CP sceptical semantics, we use the following reduction. This reduction is illustrated in Figure 1 in gold.

Definition 11 (Reduction relevance)**.** *Let (ϕ, X) be an instance of 1-*QBF *or co-1-*QBF *let $\phi = \bigwedge_i c_i$ and $c_i = \bigvee_j \alpha_j$ for each clause c_i in ϕ, where α_j are the literals over X that occur in clause c_i. Let $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \emptyset \rangle$ be the reduction from Definition 7 of this instance. We define the corresponding IAF for this instance as $\langle \mathcal{A}', \mathcal{A}^{?\prime}, \mathcal{R}', \emptyset \rangle$, where $\mathcal{A}' = \mathcal{A} \cup \{\overline{\chi}\}$; $\mathcal{A}^{?\prime} = \mathcal{A}^? \cup \{\chi\}$; and $\mathcal{R}' = \mathcal{R} \cup \{(\chi, \overline{\chi}), (\overline{\chi}, \phi)\}$.*

Using the reduction above, we can now give tight complexity bounds for GR and CP semantics for the sceptical justification status.

Proposition 8. GR-*sceptical*-IN-RELEVANCE, GR-*credulous*-IN-RELEVANCE *and* CP-*sceptical*-IN-RELEVANCE *are NP-complete.*

Proof sketch. Given an instance of 1-QBF (SAT), let $\mathcal{I}$ be the IAF constructed according to Definition 11 where $Y = \emptyset$. Addition of χ is GR-sceptical-IN-relevant for ϕ w.r.t. $\mathcal{I}$ iff there is some $\mathcal{I}' \in F(\mathcal{I})$ such that ϕ is GR-sceptical-IN, having χ in its certain arguments iff the SAT instance is True. NP-completeness follows from Proposition 8 and the fact that the corresponding JUSTIFICATION problems are in P. □

5. Related work

The computational complexity of various problems defined on argumentation frameworks is well-studied; see [15] for an overview. Most studies only identify IN arguments and do not distinguish other justification statuses; notable exceptions are [16] and [17], but neither of these works give complexity results for separate statuses, as we do.

Complexity studies on problems defined on IAFs emerged more recently. For example, variants of the verification problem on IAFs are studied in [4]. The problems of stability and relevance differ from the verification problem as they are defined on arguments rather than sets of arguments. More related is [5]: the authors study potential and necessary credulous and sceptical acceptance in IAFs, where necessary sceptical acceptance of a given argument A, for example, means that in each specification's certain projection, each extension (under a given semantics) contains A. The notions of necessary credulous and sceptical acceptance are very similar to specific stability problems: in fact, we used results regarding their complexity for proving the complexity of stable-IN statuses. Finally, the notion of stability, which was originally defined on structured argumentation frameworks in [7], is transposed to the context of IAFs in [6] and preliminary complexity results for stability under four semantics are provided. In our work, we define a more fine-grained notion of stability and provide more precise complexity characterisations.

Our notion of relevance has not been introduced or studied before. It is related to the notion of influenced sets in e.g. [18], which intuitively are sets of arguments whose justification status may change after an update. This notion is however less strict than relevance: there are situations in which some argument A would be in the influenced set of adding an uncertain attack (B,C), while addition of (B,C) is not relevant for A.

6. Conclusion

We studied the complexity of detecting stability and relevance in incomplete argumentation frameworks. First, we redefined stability [7,8,6] on IAFs. Our definition is a more fine-grained notion than the existing definition on IAFs [6], since it distinguishes between IN, OUT and UNDEC justification statuses. This distinction is appropriate in for example applications in inquiry [7,8], where a dialogue discussing a given argument should be terminated if more information cannot change the argument's (exact) justification status.

As second main contribution of this paper, we introduced the notion of relevance, which has not been studied before in the context of stability, and analysed its complexity. Returning to the application in inquiry, the identification of relevant elements can be used to select the next question, reaching a stable situation in an efficient way.

It is however unlikely that the stability and relevance problem itself can be solved efficiently for all inputs: our complexity analysis revealed that the nontrivial variants of the relevance and stability problems have a complexity ranging from the first to the third level of the polynomial hierarchy; see Table 1 for an overview. Interestingly, even within the same semantics, there are differences in the complexity of UNDEC-STABILITY problems and the corresponding IN-STABILITY problems – we consider this to be an additional reason to study a fine-grained notion of stability and relevance.

In future work we will complete Table 1, by studying the computational complexity of relevance for the other semantics and UNDEC status. In addition, to apply these theoretical concepts in practice, we plan to develop algorithms for evaluating or estimating stability and relevance. Finally, we will study stability and relevance in structured argumentation frameworks, such as a dynamic version of ASPIC$^+$, for various semantics.

References

[1] Atkinson K, Baroni P, Giacomin M, Hunter A, Prakken H, Reed C, et al. Towards artificial argumentation. AI magazine. 2017;38(3):25-36. doi:10.1609/aimag.v38i3.2704.

[2] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77:321-57. doi:10.1016/0004-3702(94)00041-X.

[3] Cayrol C, Devred C, Lagasquie-Schiex MC. Handling ignorance in argumentation: Semantics of partial argumentation frameworks. In: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Springer; 2007. p. 259-70. doi:10.1007/978-3-540-75256-1_25.

[4] Baumeister D, Neugebauer D, Rothe J, Schadrack H. Verification in incomplete argumentation frameworks. Artificial Intelligence. 2018;264:1-26. doi:10.1016/j.artint.2018.08.001.

[5] Baumeister D, Järvisalo M, Neugebauer D, Niskanen A, Rothe J. Acceptance in incomplete argumentation frameworks. Artificial Intelligence. 2021;295:103470. doi:10.1016/j.artint.2021.103470.

[6] Mailly JG, Rossit J. Stability in Abstract Argumentation. In: NMR 2020 Workshop Notes; 2020. p. 93-9. doi:10.48550/arXiv.2012.12588.

[7] Testerink B, Odekerken D, Bex F. A Method for Efficient Argument-based Inquiry. In: Proceedings of the 13th International Conference on Flexible Query Answering Systems. Springer International Publishing; 2019. p. 114-25. doi:10.1007/978-3-030-27629-4_13.

[8] Odekerken D, Borg A, Bex F. Estimating Stability for Efficient Argument-based Inquiry. In: Computational Models of Argument. Proceedings of COMMA 2020; 2020. p. 307-18. doi:10.3233/FAIA200514.

[9] Papadimitriou C. Computational Complexity. Addison-Wesley; 1994.

[10] Caminada M. On the issue of reinstatement in argumentation. In: European Workshop on Logics in Artificial Intelligence. Springer; 2006. p. 111-23. doi:10.1007/11853886_11.

[11] Dimopoulos Y, Torres A. Graph theoretical structures in logic programs and default theories. Theoretical Computer Science. 1996;170(1-2):209-44.

[12] Coste-Marquis S, Devred C, Marquis P. Symmetric argumentation frameworks. In: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Springer; 2005. p. 317-28. doi:10.1007/11518655_28.

[13] Dunne PE, Bench-Capon TJ. Coherence in finite argument systems. Artificial Intelligence. 2002;141(1-2):187-203. doi:10.1016/S0004-3702(02)00261-8.

[14] Dunne PE, Wooldridge M. Complexity of abstract argumentation. In: Argumentation in artificial intelligence. Springer; 2009. p. 85-104. doi:978-0-387-98197-0_5.

[15] Dvorák W, Dunne PE. Computational problems in formal argumentation and their complexity. Handbook of formal argumentation. 2018:631-87.

[16] Dvořák W. On the complexity of computing the justification status of an argument. In: International Workshop on Theorie and Applications of Formal Argumentation. Springer; 2011. p. 32-49. doi:10.1007/978-3-642-29184-5_3.

[17] Alfano G, Greco S, Parisi F, Trubitsyna I. Incomplete Argumentation Frameworks: Properties and Complexity. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2022. p. 5451-60. doi:10.1609/aaai.v36i5.20483.

[18] Alfano G, Greco S, Parisi F, , Simari GI, Simari GR. On the Incremental Computation of Semantics in Dynamic Argumentation. Journal of Applied Logics. 2021;8(6):1749-92.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220160

Arguing About the Existence of Conflicts[1]

Giuseppe PISANO [a,2], Roberta CALEGARI [a], Henry PRAKKEN [b,c], and
Giovanni SARTOR [a]

[a] *Alma AI—Alma Mater Research Institute for Human-Centered Artificial Intelligence,*
ALMA MATER STUDIORUM–*Università di Bologna, Italy*
[b] *Department of Information and Computing Sciences, Utrecht University, the Netherlands*
[c] *Faculty of Law, University of Groningen, the Netherlands*

Abstract. In this paper we formalise a meta-argumentation framework as an ASPIC$^+$ extension which enables reasoning about conflicts between formulae of the argumentation language. The result is a standard abstract argumentation framework that can be evaluated via grounded semantics.

Keywords. meta-argumentation, argumentation, ASPIC$^+$

1. Introduction

Meta-arguments support conclusions about other arguments, their interaction, their composition or their evaluation. For instance, a meta-argument may conclude that other arguments are in conflict or that one of them is preferred over the other, or it may provide new rules or facts that can be used in building arguments.

Meta-argumentation has received little attention thus far. As discussed in [1] there are various approaches to generate argumentation frameworks (AFs) in terms of accounts of the structure of arguments and their relations (e.g. ASPIC+, ABA, classical argumentation, DeLP). However, most of these approaches regard rule sets, specifications of conflicts and preferences as given. In the reality of adversarial debate, these things can also be argued about. Hence the importance of meta-argumentation.

In this paper we shall focus on a specific application of meta-argumentation to the conflict function of an argumentation theory, namely, assessing whether there is a conflict between two propositions in the argumentation language, i.e., whether the arguments concerning those propositions are incompatible so that accepting one of them entails rejecting the other.

Example 1 (Gender example) *To ground the discussion on a practical example, let us consider a legal example concerning a case of gender identity. Let us consider the case of Sue. She wants to compete in the woman's chess championship but the organisers argue*

[1] Research partially supported by the "CompuLaw" project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 833647).

[2] Corresponding Author: Giuseppe Pisano; E-mail: g.pisano@unibo.it.

that this would be impossible because legally she has been assigned the male gender, as proven by her passport. However, Sue is bigender – i.e. she identifies as both male and female simultaneously – and thinks that she should have the right to compete in the championship. To decide the case we should first decide on the existence of a conflict between the concepts of man and woman: are they in conflict – gender binarism discards her claim of being both man and woman at the same time –, or can the two concepts coexist according to the principle of self-determination? To encode the case at hand, the argumentation model should allow conflicts to be formalised, i.e., a meta-argumentation model is required.

It should now be more clear how the ability to include conflicts in the arguable content of a theory is fundamental in cases like the one we described above. The point of this work is whether this can be done while maintaining the compatibility with traditional argumentation methods and models—namely, Dung's semantics [2]. In this paper we focus on grounded semantics. For one reason, grounded semantics allows efficient use of the model in a real computational scenario—grounded semantics is the only one having polynomial complexity. Moreover, the use of grounded semantics only, allowed the authors – and hopefully the readers – to better focus on the fundamental ideas and mechanisms behind the proposed model, without the need to deal with the complexity of other semantics. For these reasons, the extension to other semantics is left to future work.

The main idea behind this work is to start from a standard structured argumentation framework – like ASPIC$^+$ [3] – and expand its definitions to deal with meta reasoning over conflicts. We address meta-argumentation by using the mechanism presented in [4] for preferences and adapting it to conflicts, i.e., in representing attacks and conflicts through arguments, which in their turn, may be subject to attack. In this way, we can model meta-argumentation while preserving the semantics of standard abstract argumentation [2].

The paper is organised as follows. Background notions are discussed in Section 2, while Section 3 introduces the meta argumentation framework. Section 4 presents the related work and conclusions are drawn in Section 5.

2. Abstract Argumentation and Argumentation Theories

In this section we introduce the standard definitions for argumentation frameworks based on Dung's semantics [2] and for ASPIC$^+$.

Definition 1 (Argumentation framework) *An argumentation framework AF is a tuple $< A, \rightsquigarrow >$, where A is a set of arguments and $\rightsquigarrow$ is a binary relation (attack relation) over $A \times A$. We write $X \rightsquigarrow Y$ for $(X,Y) \in \rightsquigarrow$.*

The semantics for an argumentation framework is defined as follows.

Definition 2 (Semantics) *Let $< \mathscr{A}, \rightsquigarrow >$ be an AF and $S \subseteq \mathscr{A}$. S is conflict free iff there are no $A, B \in S$ such that $A \rightsquigarrow B$. For any $X \in \mathscr{A}$, X is acceptable with respect to $S \subseteq \mathscr{A}$ iff $\forall Y \in \mathscr{A}$, $Y \rightsquigarrow X$ implies that $\exists Z \in S$ s.t. $Z \rightsquigarrow Y$. Then:*

- *S is an admissible extension iff $X \in S$ implies that X is acceptable w.r.t. S;*

- *S is a complete extension iff $X \in S$ iff X is acceptable w.r.t. S;*
- *S is the grounded extension iff S is the set-inclusion minimal complete extension.*

Definition 3 (Argumentation system) *An argumentation system is a quadruple AS=$< L, R, n, \rhd >$ where:*

- *L is a logical language;*
- *$R = R_s \cup R_d$ is a set of rules. R_d is a set of defeasible rules in the form $\phi_0, ..., \phi_n \Rightarrow \phi$, R_s is a set of strict rules in the form $\phi_0, \ldots, \phi_n \rightarrow \phi$, where $\phi_0, \ldots, \phi_n, \phi$ are well-formed formulae in the L language;*
- *n is a naming function of the form $n : R \mapsto L$*
- *$\rhd$ is a non-symmetrical conflict relation over $L \times L$. We write $\phi \rhd \psi$ for $(\phi, \psi) \in \rhd$.*

Definition 4 (Knowledge base) *A knowledge base for an AS=$< L, R, n, \rhd >$ is a set $K \subseteq L$ consisting of two disjoint subsets K_s (the axioms) and K_p (the ordinary premises).*

Definition 5 (Argumentation theory) *An argumentation theory is a tuple AT=$<AS, K>$ where AS is an argumentation system and K is a knowledge base in AS.*

Given an argumentation theory, by chaining rules from the theory we can construct arguments, as specified in the following definition; cf. [5,6,7].

Definition 6 (Argument) *Starting from an argumentation theory AT=$< AS, K >$, an argument A is any structure obtained by applying the following steps a finite number of times*

1. *ϕ if $\phi \in K$ with: Prem(A)=$\{\phi\}$; Conc(A)=ϕ; Sub(A)=$\{\phi\}$; DefRules(A)=$\emptyset$; TopRule(A)=undefined.*
2. *$A_1, \ldots, A_n \Rightarrow \psi$ if $A_1, \ldots, A_n$ are arguments s.t. $\exists$ a rule $r = Conc(A_1), \ldots, Conc(A_n) \Rightarrow \psi \in R_d$.*
   - *Prem(A)=$Prem(A_1) \cup \ldots \cup Prem(A_n)$,*
   - *Conc(A)=ψ,*
   - *Sub(A)=$Sub(A_1) \cup \ldots \cup Sub(A_n) \cup \{A\}$,*
   - *TopRule(A)=r,*
   - *DefRules(A)=$DefRules(A_1) \cup \ldots \cup DefRules(A_n) \cup \{r\}$*
3. *$A_1, \ldots, A_n \rightarrow \psi$ if $A_1, \ldots, A_n$ are arguments s.t. $\exists$ a rule $r = Conc(A_1), \ldots, Conc(A_n) \rightarrow \psi \in R_s$.*
   - *Prem(A)=$Prem(A_1) \cup \ldots \cup Prem(A_n)$,*
   - *Conc(A)=ψ,*
   - *Sub(A)=$Sub(A_1) \cup \ldots \cup Sub(A_n) \cup \{A\}$,*
   - *TopRule(A)=r,*
   - *DefRules(A)=$DefRules(A_1) \cup \ldots \cup DefRules(A_n)$*

Given an argument A we write:

- $Prem(A)$, for the set of premises from K used in the argument;
- $Conc(A)$, for the conclusion of the argument;
- $Sub(A)$, for the set of subarguments of A;
- $DefRules(A)$, for the set of rules in R_d used to build the argument;

- $TopRule(A)$, for the rule from R used in A's last inference step.

The first condition deals with arguments generated using the knowledge base K. Using the second and third ones we can recursively apply rules from R on the generated arguments to generate new arguments.

We can produce attacks starting from arguments using the notion of conflict for an argumentation language L:

Definition 7 (Direct attack) *An argument A directly attacks an argument B iff A directly undercuts, directly undermines or directly rebuts B where:*

- *A directly undercuts B iff* $Conc(A) \triangleright n(TopRule(B))$ *and* $TopRule(B) \in R_d$*;*
- *A directly rebuts argument B iff* $Conc(A) \triangleright Conc(B)$ *and* $TopRule(B) \in R_d$*;*
- *A directly undermines argument B iff* $B \in K_p$ *and* $Conc(A) \triangleright B$

Definition 8 (Attack) *We say that argument A attacks argument B if A directly attacks* $B' \in Sub(B)$.

Then we can build an abstract argumentation framework as:

Definition 9 (Abstract argumentation framework) *Let AT be an argumentation theory* $< AS, K >$*. An abstract argumentation framework defined by AT, is a tuple* $< \mathscr{A}, \rightsquigarrow >$ *where:*

- $\mathscr{A}$ *is the set of all arguments constructed from AT according to Definition 6;*
- *for any arguments X and* $Y \in \mathscr{A}$*,* $X \rightsquigarrow Y$ *iff X attacks Y*

In the following sections, we will extend this model to base the $\triangleright$ relation and consequently the $\rightsquigarrow$ relation on the content of the argumentation theory—i.e., shape the applicable conflicts inside the framework that we are evaluating.

The desiderata as a result is a standard abstract argumentation framework, thus preserving the possibility to evaluate it through the semantics given in Definition 2.

3. Reasoning with conflicts

According to Definition 3, the conflict relation is a fixed part of the argumentation system and attacks between arguments are determined by conflicts between the conclusion of the attacking argument and a premise, rule name, or conclusion of the directly attacked argument. The main idea underpinning the extension for dealing with meta-argumentation is to make the conflict relation dynamic, allowing arguments to argue for or against the existence of conflicts. In such a way we define an abstract argumentation framework that – once evaluated according to a standard Dung's semantics – produces admissible extensions containing both the arguments arguing on conflicts and arguments whose admissibility is influenced by these conflicts.

Let us start providing definitions for an argumentation language L enabling conflicts between elements of L to be stated, i.e., enabling reasoning with conflicts. We do that by introducing in the language a binary predicate – *conf* – putting in relation arbitrary formulae from the language itself. The introduced predicate will provide a way to express the content of the conflict relation $\triangleright$ and use it in the argumentation process. Furthermore, we introduces wff's att(A) for any argument constructible with any possible set of rules over the new language.

Definition 10 (Conflict-based argumentation language) *Given an argumentation language L we define a argumentation language for reasoning with conflicts L_c as the smallest argumentation language $L_c = L \cup \{conf(\psi, \phi) | \psi, \phi \in L_c\} \cup \{attA \mid A$ is constructible with any set of rules over $L_c\}$.*

Now let us consider L_c a language as in Definition 10. We can build an argumentation system AS=$< L_c, R, n, \emptyset >$ and consequently an argumentation theory AT=$< AS, K >$, and use them to build an abstract argumentation framework AF=$< \mathscr{A}, \rightsquigarrow >$ using Definition 9. Note that, since $\rhd = \emptyset$, the attack set $\rightsquigarrow$ in AF will be empty as well.

Now, let us extend the AF framework so defined to introduce attacks derived from the conflicts reified in the L_c language. In such a way the status of an attack is bound to the status of the argument claiming the conflict that generated it.

First, let us define an argument for each potential attack deriving from $conf$ predicates. Accordingly, attacks could be evaluated w.r.t. the semantics applied to the framework.

Definition 11 (Conflict-based direct attack argument) *A conflict-based direct attack argument X stating that argument W, based on conflict argument W', attacks argument Z, has the form $W, W' \Rightarrow att(Z)$ where:*

- *$Conc(W) = \phi$, $Conc(W') = conf(\phi, \psi)$ and*
  - ∗ *$n(TopRule(Z)) = \psi$ and $TopRule(Z) \in R_d$, or*
  - ∗ *$Conc(Z) = \psi$ and $TopRule(Z) \in R_d$ or*
  - ∗ *$Conc(Z) = \psi$ and $Z \in K_p$*

We define:

- *$Conc(X) = att(Z)$*
- *$Sub(X) = Sub(W) \cup Sub(W') \cup \{X\}$*

Let us write $DirectAttack(X)$ to indicate that X is a direct attack argument.

Thus to construct a *direct attack argument* $W, W' \Rightarrow att(Z)$ against Z it must be the case that two arguments are available, argument W, and argument W', the latter claiming that the conclusion of W is in conflict with the relevant element of Z (i.e., the name of Z's top rule or Z's conclusion). The status of the direct attack arguments will depend on the status of both W and W'.

We leverage direct attack arguments to build the actual *attack* set of the meta argumentation framework.

Definition 12 (Conflict-based attack) *A direct attack argument $W, W' \Rightarrow att(Z)$ attacks any argument Z' such that $Z \in Sub(Z')$.*

Thus, a direct attack argument $W, W' \Rightarrow att(Z)$ attacks not only its direct target Z, but also any argument Z' of which Z is a subargument. The success of the attack will depend not only on the status of W, but also on the status of W' which asserts that W and Z are in conflict.

These elements are merged together in a *Conflict-based Argumentation Framework*.

Definition 13 (Conflict-based Argumentation Framework) *Given an argumentation theory AT=$<< L_c, R, n, \emptyset >, K >$ with L_c being a conflict-based argumentation language, the conflict-based argumentation framework of AT is the tuple $< \mathscr{A}_1 \cup \mathscr{A}_2, \rightsquigarrow >$ where:*

- *$\mathscr{A}_1$ is the set of all arguments constructed from AT according to Definition 6;*
- *$\mathscr{A}_2$ is the set of all direct attack arguments constructed from AT and A_1 according to definition 11;*
- *$X \rightsquigarrow Z$ iff X attacks Z according to definition 12*

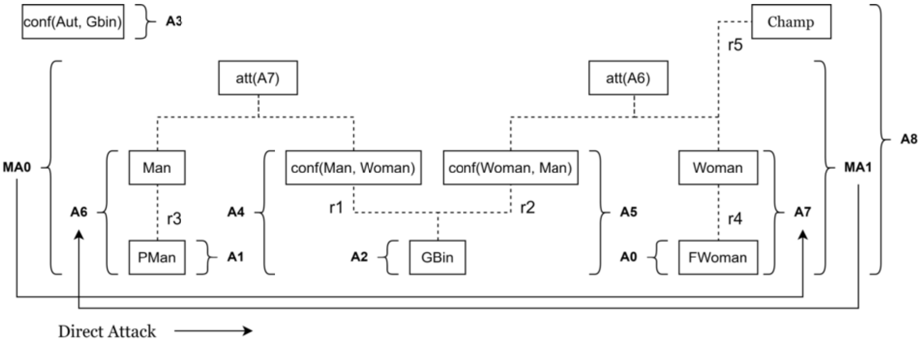
Conflict freeness, acceptability, admissible, complete, grounded extension are defined as in Definition 2.

The set $\mathscr{A}_2$ contains all attack arguments that can be generated by using the arguments in A_1, according to definition 13. For an attack argument $W, W' \Rightarrow att(Z)$ to be established according to an argumentation semantics, it is necessary that also W' is acceptable, i.e., that it is established that an acceptable conflict between W and Z exists. Only in this case W will bring an attack against Z.

Let us now provide some examples for our framework, in accordance the legal example introduced in Example 1. We use the following abbreviations:

- Champ = Sue can compete in the women's chess championship
- FWoman = Sue is bigender and identifies herself also as a woman
- PMan = Sue's passport identifies her as a man
- Aut = Every person has the right to self-determine their gender
- GBin = Every person's gender is determined by their birth sex, either male or female

Example 2 (Gender example: formalization) *Let us consider the theory where $R_d = \{$ r1 : Gbin $\Rightarrow$ conf(Man,Woman); r2 : Gbin $\Rightarrow$ conf(Woman,Man); r3 : PMan $\Rightarrow$ Man; r4 : FWoman $\Rightarrow$ Woman; r5 : Woman $\Rightarrow$ Champ $\}$ with the following facts $K_p = \{$ FWoman, PMan, GBin, conf(Aut, Gbin) $\}$, $K_s = \emptyset$, $R_s = \emptyset$. Accordingly to the above definitions, we can then build the following arguments:*



The attacks are $MA0 \rightsquigarrow A7$, $MA0 \rightsquigarrow A8$, $MA1 \rightsquigarrow A6$, $MA0 \rightsquigarrow MA1$, $MA1 \rightsquigarrow MA0$. If we apply Dung's grounded semantics to the framework we obtain the extension $\{A0, A1, A2, A3, A4, A5\}$—i.e. the incompatibility between Sue's official gender and her other perceived identity $(A4, A5)$ prevents her to compete in the championship.

Sue is not happy with the final decision and decides to appeal claiming that her right to self-determination has not been taken into due consideration. The case is evaluated again with the new information: $K'_p = \{Aut\} \cup K_p$. *Two new arguments are obtained:*

$A9\ :\ Aut$
$MA2\ :\ A9,\ A3\ \Rightarrow\ att(A2)$

The new attacks are $MA2 \rightsquigarrow A2$, $MA2 \rightsquigarrow A4$, $MA2 \rightsquigarrow A5$, $MA2 \rightsquigarrow MA0$, $MA2 \rightsquigarrow MA1$. *Applying again Dung's grounded semantics to the framework we obtain the extension* $\{A0, A1, A3, A6, A7, A8, A9, MA2\}$*—i.e. the problem on Sue's identity is resolved discarding the binary view on genders (A2, A4, A5), according to the principle of self-determination (A9). Consequently, Sue's perceived genders are both present in the extension, and she is free to compete in the championship. Indeed, the CAF was able to integrate the new knowledge and use it to revise the status of the propositional conflicts in the argumentation theory as expected. Both the original argumentation graph and the revised one are presented in Figure 1.*

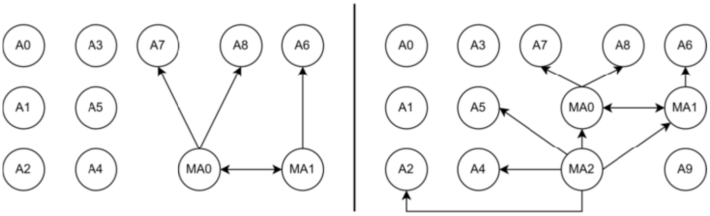


Figure 1. Conflict-based Argumentation frameworks from Example 2

3.1. Properties

We now proceed to demonstrate two important properties of the constructed framework. Intuitively, we would expect that a conflict that has been proven to exist at the meta-level – i.e. via the conflict-based framework –, indeed exists at the object level, leading to the same set of attacks and, consecutively, to the same extension. In other words, what is true according to the conflict-based framework should remain true when the verified conflicts are applied a priori as in the original ASPIC$^+$ model. To demonstrate this important property, let us introduce the notion of an *Equivalent Standard AF*.

To start, we define a way to construct a standard argumentation framework on the basis of a conflict-based argumentation framework. The basic idea is that starting with a conflict-based argumentation framework and an extension of it, we construct a standard argumentation framework having a corresponding extension according to the same semantics.

Let us consider a conflict-based argumentation framework CAF=$<\mathscr{A}, \rightsquigarrow>$ and one of its extensions E. To construct the equivalent argumentation framework EAF, we first remove from $\mathscr{A}$ (a) all attack arguments that are supported by those conflict arguments that are in the extension, and (b) all attack arguments that are supported by a conflicting argument that is attacked by the extension. Only attack arguments that are neither included in E nor attacked by it are left in the EAF's arguments set. Accordingly, the EAF's attack relation is constructed using those conflicts claimed by the arguments in E.

Definition 14 (Equivalent Standard AF) *Given a conflict based argumentation framework CAF=$< \mathscr{A}, \rightsquigarrow >$ having an extension E according to semantics σ, we define an equivalent standard argumentation framework EAF=$< \mathscr{A}', \rightsquigarrow' >$ where:*

1. $\mathscr{A}' = \mathscr{A} \setminus B \cup C$ *where:*
   (a) $B = \{a \in \mathscr{A} \mid \exists b \in E$ *such that* $Conc(b) = conf(\phi, \psi)$ *and a is a direct attack argument of the form* $W, b \Rightarrow Z\}$*;*
   (b) $C = \{a \in \mathscr{A} \mid \exists b \in \mathscr{A}$ *such that* $Conc(b) = conf(\phi, \psi)$ *and b is attacked by E and a is a direct attack argument of the form* $W, b \Rightarrow Z\}$*.*
2. $\rightsquigarrow' = \rightsquigarrow_{|\mathscr{A}' \times \mathscr{A}'} \cup \{(a,b) \mid a, b \in \mathscr{A}'$ *and* $\exists c \in E$ *such that* $Conc(c) = conf(\phi, \psi)$ *and* $\exists\, b' \in Sub(b)$ *s.t. a directly attacks* b' *(Definition 7) according to the conflict* $\phi \rhd \psi\}$*.*

Proposition 1 *Consider a finitary CAF=$< \mathscr{A}, \rightsquigarrow >$ and its corresponding EAF=$< \mathscr{A}', \rightsquigarrow' >$ built on the grounded extension E and having grounded extension E'. Then $E \cap \mathscr{A}' = E'$.*

Proof 1 *Let's consider an argumentation framework CAF $=< \mathscr{A}, \rightsquigarrow >$. We call the characteristic function of CAF the function $F : 2^{\mathscr{A}} \to 2^{\mathscr{A}}$ such that $F(Args) = \{X | \forall Y$ such that $Y \rightsquigarrow X$, then $\exists Z \in Args$ such that $Z \rightsquigarrow Y\}$ where $Args \subseteq \mathscr{A}$. Let us consider grounded extension as the minimal conflict-free fixed point of the characteristic function F—i.e. the union of a sequence $E_0, \ldots, E_n$ obtained by iterative application of the F function on the empty set, and where $E_0 = \emptyset$. We prove that $E \cap \mathscr{A}' = E'$.*

We first prove that $E \cap \mathscr{A}' \subseteq E'$. Suppose $a \in E \cap \mathscr{A}'$. We prove that $a \in E'$ as follows.

Base case: *a has no attackers in $\mathscr{A}$ according to $\rightsquigarrow$ so $a \in E_1 \cap \mathscr{A}'$. Then there can only be attackers of a in $\mathscr{A}'$ according to $\rightsquigarrow'$ if there is a relevant conflict argument b in E that says that the conclusion of some argument $x \in \mathscr{A}$ conflicts with a's conclusion. But then there exists a direct attack argument $x, b \Rightarrow att(a)$ in $\mathscr{A}$, which contradicts that a has no attackers in $\mathscr{A}$. So $x \notin \mathscr{A}'$, so a has no attackers in $\mathscr{A}'$, so $a \in E'$.*

Induction step: *Assume that all arguments in $E_{i-1} \cap \mathscr{A}'$ are in E'. Consider any $a \in E_i$. Any $b \in \mathscr{A}'$ such that $b \rightsquigarrow' a$ is such that $b \rightsquigarrow a$ or $b \not\rightsquigarrow a$. First, any such b such that $b \rightsquigarrow a$ is attacked by E_{i-1} according to $\rightsquigarrow$. Then by the induction hypothesis, if $b \in \mathscr{A}'$, then b is also attacked by E'. Next, consider any such b such that $b \not\rightsquigarrow a$. Then there is a direct attack argument $m \in \mathscr{A}$ of the form $b, X' \Rightarrow att(a')$ with $a' \in Sub(a)$. Then $m \rightsquigarrow a$ so there exists an $m' \in E_{i-1}$ such that $m' \rightsquigarrow m$. Note that m is a direct attack argument, so m is of the form $c, Z \Rightarrow att(b')$ with $b' \in Sub(b)$. By closure of E under subarguments (an easy adaptation of the same result on standard ASPIC+), c and Z are also in E_{i-1}. But then by the induction hypothesis $c \in E'$ and $c \rightsquigarrow' b$. So $a \in E'$.*

We next prove that $E' \subseteq E \cap \mathscr{A}'$. Suppose $a \in E'$. We prove that $a \in E$ as follows.

Base case: *a has no attackers in $\mathscr{A}'$ so $a \in E'_1$. Consider any $b \in \mathscr{A}$ such that $b \rightsquigarrow a$. Then b is a direct attack argument of the form $c, X \Rightarrow att(a)$, with X a conflict argument that says that the conclusion of argument $c \in \mathscr{A}$ conflicts with a's conclusion. But since a has no attackers in $\mathscr{A}'$, we have that $b \notin \mathscr{A}'$ because of either condition (1a) or condition (1b) of Definition 14. In the case of (1b), b is attacked according to $\rightsquigarrow$ on X by an argument in E. In the case of (1a), we have $X \in E$, so, according to condition (2) of Definition 14, we have that $c \rightsquigarrow' a$. But this contradicts that a has no attackers in $\mathscr{A}'$. The two cases together prove that $a \in E$.*

Induction step: *Assume that all arguments in E'_{i-1} are in E. Consider any $a \in E'_i$. Then all $b \in \mathscr{A}'$ such that $b \rightsquigarrow' a$ are attacked by E'_{i-1} according to $\rightsquigarrow'$. Then b could be either a regular argument or a direct attack argument of the form $c, X \Rightarrow att(a')$ with $a' \in Sub(a)$. In the latter case, since $b \in \mathscr{A}$ then, by the induction hypothesis, b is also attacked by E according to $\rightsquigarrow$. In the first case, since $b \rightsquigarrow' a$, there must exist a direct attack argument $m \in \mathscr{A}$ of the form $b, X \Rightarrow att(a')$ with $a' \in Sub(a)$ such that $m \rightsquigarrow a$. But, by induction hypotheses, b and m are both attacked according to $\rightsquigarrow$ by E. The two cases together prove that $a \in E$.*

Since $E \cap \mathscr{A}' \subseteq E'$ and $E' \subseteq E \cap \mathscr{A}'$ then $E' = E \cap \mathscr{A}'$.

The second property we want to demonstrate builds on top of what we have just proven. We have seen that it is possible to move the conflicts in the grounded extension at the object level without altering the results. However, the resulting *Equivalent Standard AF* still contains the meta-level attack arguments that are not in the extension or attacked by a member of it. The question is whether there are cases in which the conflict framework can be completely transformed into a regular argumentation framework. The implication of this finding would be straightforward: the Conflict-based framework would be a generalisation of a regular abstract argumentation framework. This is a fundamental property for every model trying to provide a conservative extension like ours.

The next proposition shows that a *Conflict-based Argumentation Framework* is a generalisation of a standard abstract argumentation framework.

Proposition 2 *Consider a CAF=$< \mathscr{A}, \rightsquigarrow >$ and its corresponding equivalent EAF=$< \mathscr{A}', \rightsquigarrow' >$ built on the σ extension E. If $\forall x \in \{a \in \mathscr{A} | Conc(a) = conf(\phi, \psi)\}$ we have that either $x \in E$ or $\exists (d, x) \in \rightsquigarrow$ s.t. $d \in E$, then EAF is a regular argumentation framework as in Definition 9.*

Proof 2 *By Definition 14 if all the conflict arguments are either in the extension or attacked by a member of it, then all the Direct Attack Arguments can be discarded leaving only the arguments produced using Definition 6. The attack set would then be given by the set of conflicts claimed by the argument in the extension using Definition 7. Consequently, the result is a regular argumentation framework as in Definition 9.*

Example 3 (Gender Example: propositions) *To ground Proposition 1 and 2 let us consider again the framework in Example 2.First, we have to build the Equivalent Standard AF using the results of Dung's grounded semantics. All the conflict arguments are either in the extension ($A3$) or attacked by a member of the extension ($A4$, $A5$). Accordingly, we can delete from the set of arguments the linked attack arguments ($MA0$, $MA1$, $MA2$), and use the conflict claimed by $A3$ to build the new attack set. We have only three attacks, $A9 \rightsquigarrow A2$, $A9 \rightsquigarrow A4$ and $A9 \rightsquigarrow A5$, as shown in Figure 2.*

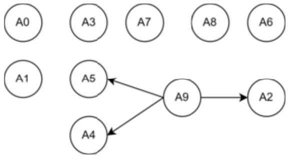


Figure 2. Equivalent Standard AF from Example 3

If we apply the grounded semantics to the framework then we obtain the extension $\{A0, A1, A3, A6, A7, A8, A9\}$*—the same as in the Conflict-based AF but without attack arguments, as claimed by Property 1. It is worth noting that we obtained a regular framework as result: all the conflicts are known a priori and the same framework could have been built using the standard ASPIC*$^+$ *definitions.*

In the general case, however, we cannot have a complete equivalency between a CAF and a regular framework. Indeed, if an argument for $conf(\psi,\phi)$ is undecided – neither in the extension nor attacked by one of its members –, then the uncertainty can be propagated to the attack argument and then to the attacked argument, thus preventing them to be part of the extension. We could not obtain the same result without considering the Direct Attack Argument, because the absence of the conflict would potentially allow the attacked argument to be accepted without considering the potential uncertainty in the state of the conflict. In other words, a CAF framework is capable of conveying more information on the state of an attack w.r.t. a standard argumentation framework, thus making the transformation to a regular framework impossible in the general case.

Example 4 (Partial Transformation) *Let us consider the theory where* $K_p = \{p, q, r, -r\}$ *and* $K_s = \{conf(r,-r), conf(-r,r)\}$ *and* $R_d = \{r => conf(p,q)\}$*. Starting from this theory we can build the Conflict-based framework and then the Equivalent one as shown in Figure 3.*

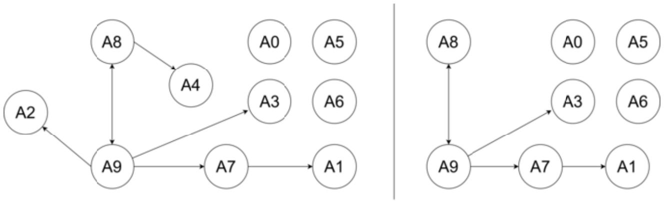
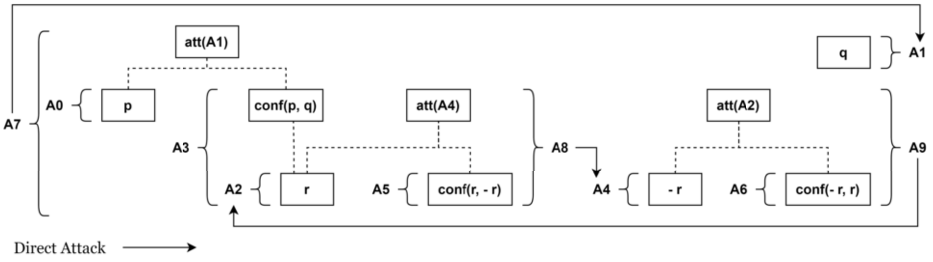


Figure 3. Conflict-based Argumentation framework from Example 4 on the left, Equivalent framework on the right.



If we apply Dung's grounded semantics to the frameworks, in both cases we obtain the extension $\{A0, A5, A6\}$*. It can be noticed that the Equivalent framework still contains an attack argument (*$A7$*) due to the uncertainty in* $A3$*'s evaluation. Indeed, without knowing if* $A3$ *is in the extension or definitely rejected – i.e. attacked by a member of the extension –, it is impossible to decide whether* $A0$ *should attack* $A1$ *or not in the Equivalent Standard AF. Consequently, every alteration of the Equivalent attack set on the basis of this conflict would lead to a possible modification in the semantics results—i.e. the attack argument* $A7$ *with the connected attacks must be preserved in the Equivalent AF.*

4. Related Research

Modgil & Bench-Capon [8] introduce the notion of meta-level argumentation frameworks. The arguments of meta-level argumentation frameworks make claims about object-level abstract argumentation frameworks according to the theory of such frameworks, for example, “A is in a preferred extension of AF” or “argument A in AF defeats argument B in AF”. Constraints are formulated on the attacks of the meta-level framework to ensure that such statements are correct with respect to the object level. For example, “y defeats x” attacks “x is justified”. This allows the formalisation of Dung’s theory of abstract argumentation frameworks in meta-level argumentation frameworks that have the same semantics as Dung’s original frameworks. Moreover, Modgil & Bench-Capon show that the same approach can be used to formalise variants of Dung-style argumentation frameworks, such as preference- and value based AFs and extended AFs. In a similar way, Boella & al. [9] develop a general methodology for instantiating Dung’s original argumentation frameworks starting from extended argumentation frameworks through a flattening technique—comparably to what is done in [10]. The resulting framework operates on meta-arguments, for example in the form “argument A is accepted” while remaining in the formal framework of Dung’s argumentation theory. While these approaches are theoretically very interesting, they do not specify the structure of arguments at the object level and therefore seem less suitable for knowledge representation.

Moving beyond abstract argumentation, [11] introduces a variant of defeasible logic, Defeasible Meta-Logic, to represent defeasible meta-theories, by proposing algorithms to compute the (meta-)extensions of such theories, and by proving their computational complexity.

Wooldridge & al. [12] develop a completely different approach for dealing with the meta-argumentative nature of argument systems. The work proposes a hierarchical first-order meta-logic, producing a three tiers argument system. Level 0 contains statements on the object domain, level 1 introduces the notion of arguments and acceptability, while level 2 is used to reason on the structure of arguments and their relations. This formalism – because of the required hierarchical representation –, although enabling a clear separation between meta- and object- level concepts, could result in decreased flexibility in the formalisation of the knowledge in the system.

A limited kind of meta-argumentation can be found in argumentation frameworks that allow for arguments about preferences. In [13] conflicts between mutually rebutting arguments are decided by preferences, which are established by arguments included in the same argumentation framework. A fix-point semantics is used to compute extensions including preference arguments. Reasoning about preferences has been recently modelled by introducing a new kind of attack, namely, a preference-based attack against attacks [14]. Dung & al. [4] expands this idea, by having a framework that includes attack arguments, as well as preference attack arguments against attacks. In this way, the framework obtained can be evaluated by using standard Dung semantics.

5. Conclusions

Our paper has presented a meta-argumentation framework for reasoning over conflicts. In particular, we have provided an ASPIC$^+$ extension allowing the encoding of conflicts between formulae in the argumentation language. The conflicts – on which can be argued in the framework – are exploited to build meta-arguments representing attacks between arguments. The result is a framework in which the set of valid attacks is dynamically connected to the acceptability status of the conflicts used to derive them. In this way, we have modelled meta-argumentation while preserving the semantics of standard abstract argumentation introduced by [2].

At the moment, this work is limited to grounded semantics only. A natural direction for a future extension is to also provide formal proofs of the framework soundness for other Dung's semantics—i.e. complete, stable, preferred. Future work will also be devoted to comparing with alternative approaches, e.g. [12] as applied in [15], and extending the model so as to include other meta-components in the framework—e.g. conditional preferences [4] and nested meta-rules [11].

References

[1] Besnard P, Garcia A, Hunter A, Modgil S, Prakken H, Simari G, et al. Introduction to structured argumentation. Argument & Computation. 2014;5(1):1-4.

[2] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995;77(2):321-58.

[3] Prakken H. An abstract framework for argumentation with structured arguments. Argument and Computation. 2010;1(2):93-124.

[4] Dung PM, Thang PM, Son TC. On Structured Argumentation with Conditional Preferences. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019. Honolulu, Hawaii, USA: AAAI Press; 2019. p. 2792-800.

[5] Caminada M, Amgoud L. On the Evaluation of Argumentation Formalisms. Artificial Intelligence. 2007;171(5—6):286-310.

[6] Modgil S, Prakken H. The *ASPIC*$^+$ framework for structured argumentation: a tutorial. Argument & Computation. 2014;5(1):31-62.

[7] Vreeswijk G. Abstract Argumentation Systems. Artif Intell. 1997;90(1-2):225-79. Available from: https://doi.org/10.1016/S0004-3702(96)00041-0.

[8] Modgil S, Bench-Capon TJM. Metalevel Argumentation. Journal of Logic and Computation. 2011;21:959-1003.

[9] Boella G, Gabbay DM, van der Torre LWN, Villata S. Meta-Argumentation Modelling I: Methodology and Techniques. Studia Logica. 2009;93(2-3):297-355.

[10] Gabbay DM. Semantics for Higher Level Attacks in Extended Argumentation Frames Part 1: Overview. Studia Logica. 2009;93(2-3):357-81.

[11] Olivieri F, Governatori G, Cristani M, Sattar A. Computing Defeasible Meta-logic. In: Logics in Artificial Intelligence. Cham: Springer International Publishing; 2021. p. 69-84.

[12] Wooldridge M, McBurney P, Parsons S. On the Meta-Logic of Arguments. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems. New York, NY, USA: Association for Computing Machinery; 2005. p. 560-7.

[13] Prakken H, Sartor G. A dialectical model of assessing conflicting arguments in legal reasoning. Artificial Intelligence and Law. 1996;4:331-68.

[14] Modgil S, Prakken H. Reasoning about Preferences in Structured Extended Argumentation Frameworks. In: Baroni P, Cerutti F, Giacomin M, Simari GR, editors. Computational Models of Argument. Proceedings of COMMA 2010. IOS; 2010. p. 347-58.

[15] van der Weide TL, Dignum F. Reasoning about and Discussing Preferences between Arguments. In: Argumentation in Multi-Agent Systems - 8th International Workshop, ArgMAS 2011, Taipei, Taiwan, May 3, 2011, Revised Selected Papers. vol. 7543 of Lecture Notes in Computer Science. Springer; 2011. p. 117-35.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220161

Formalising an Aspect of Argument Strength: Degrees of Attackability

Henry PRAKKEN

Department of Information and Computing Sciences, Utrecht University and Faculty of Law, University of Groningen, The Netherlands and European University Institute, Florence, Italy

Abstract. This paper formally studies a notion of dialectical argument strength in terms of the number of ways in which an argument can be successfully attacked in expansions of an abstract argumentation framework. The proposed model is abstract but its design is motivated by the wish to avoid overly limiting assumptions that may not hold in particular dialogue contexts or in particular structured accounts of argumentation. It is shown that most principles for gradual argument acceptability proposed in the literature fail to hold for the proposed notion of dialectical strength, which clarifies their rational foundations and highlights the importance of distinguishing between logical, dialectical and rhetorical argument strength.

Keywords. Dialectical argument strength, Structure of arguments, Nature of attack

1. Introduction

A recent trend in the formal study of argumentation is the development of gradual notions of argument acceptability, as alternatives to extension-based notions defined on top of the theory of abstract [9] or bipolar [7] argumentation frameworks. In [14] we argued that such work should make explicit which kind of argument strength or acceptability is modelled, since different kinds of strength may have different properties. In particular, we distinguished between logical, rhetorical and dialectical argument strength.

Logical argument strength in turn divides into two aspects. *Inferential argument strength* is about how well an argument's premises support its conclusion considering only the argument itself. For example, deductive arguments are stronger than defeasible arguments. *Contextual argument strength* is about how well the conclusion of an argument is supported in the context of all given arguments. Formal frameworks like Dung's theory of abstract argumentation frameworks, assumption-based argumentation, *ASPIC*$^+$ and defeasible logic programming formalise this kind of argument strength [10].

Rhetorical argument strength looks at how capable an argument is to persuade other participants in a discussion or an audience. Persuasiveness essentially is a psychological notion; although principles of persuasion may be formalised, their validation as principles of successful persuasion is ultimately psychological.

Finally, *dialectical argument strength* looks at how challengeable an argument is in the context of a critical discussion. In [15, pp. 657] this is formulated as

> (...) the (un)availability of participant moves that constrain further interlocutor moves. Minimally, argument strength thus is a function of the (un)availability of non-losing future participant moves. In this sense, the strongest proponent-argument leaves no further opponent-move except concession (i.e., retraction of either a standpoint or of critical doubt), and the weakest proponent argument constrains no opponent-move, given the "move-space".

Thus conceived, an important aspect of dialectical strength is the degree of attackability of an argument, that is, how many attacks are allowed in a given state that decrease the argument's contextual status. This reflects an intuition that many decision makers are aware of, namely, to justify one's decisions as sparsely as possible, in order to minimise the chance of successful appeal. It is this notion of dialectical strength that is the focus of the present paper.

We first propose a refined version of the notion of a normal expansion [3] of an abstract argumentation framework, designed so as to avoid overly limiting assumptions about the nature of arguments and their relations and the dialogical context. We then formalise dialectical argument strength in terms of teh number of ways to expand an argumentation framework such that the argument's contextual status decreases. We define this notion in two equivalent ways (ranking-based and weighted) and we investigate some of its formal properties. Among other things, we show that most principles for gradual argument acceptability proposed in the literature fail to hold for our notion of dialectical strength, which says something about the rational foundations of these principles.

2. Formal Preliminaries

An *abstract argumentation framework* (AF) [9] is a pair $(\mathcal{A}_{AF}, \mathcal{C}_{AF})$, where $\mathcal{A}_{AF}$ is a set of arguments and $\mathcal{C}_{AF} \subseteq \mathcal{A}_{AF} \times \mathcal{A}_{AF}$ is a relation of attack. We write $A \in AF$ as shorthand for $A \in \mathcal{A}_{AF}$ and we will omit the subscripts if there is no danger for confusion. We will sometimes in text present an AF as $A \leftarrow B \leftrightarrow C$, to denote that $\mathcal{A} = \{A, B, C\}$ and $\mathcal{C} = \{(B, A), (B, C), (C, B)\}$. Argument A is an *attacker* of argument B if $(A, B) \in \mathcal{C}$, and A is a *direct defender* of B if for some attacker C of B it holds that $(A, C) \in \mathcal{C}$. An *attack branch*, respectively, *defense branch* of an argument A_1 is a finite sequence $A_1, \ldots, A_n$ such that n is even, respectively, odd, and in both cases A_n has no attackers and for each $i < n$ it holds that A_{i+1} attacks A_i. Argument B_i is a *defender* of argument A_1 iff B_i is in an attack or defense branch of A_1 and $i > 1$ and i is odd.

The semantics of AFs [9,2] identifies sets of arguments (called *extensions*) which are internally conflict-free (no member attacks a member) and defend themselves against all attackers. In this paper we use the labelling way to define semantics for AFs. A *labelling* of a set $\mathcal{A}$ of a set of arguments in an $AF = (\mathcal{A}, \mathcal{C})$ is any triple of non-overlapping subsets (*in*,*out*,*und*) of $\mathcal{A}$ that satisfies the following constraints:

1. an argument is *in* iff all arguments attacking it are *out*;
2. an argument is *out* iff it is attacked by an argument that is *in*;
3. an argument is *und* (for 'undecided') iff it is neither *in* nor *out*.

In this paper we focus on grounded semantics, leaving generalisation to other semantics for future research. The grounded labelling of an AF minimises the set of arguments that are labelled *in* and is always unique. A set $S \subset \mathcal{A}$ is called the *grounded extension* of AF iff S is the set of all arguments labelled *in* in the grounded labelling.

3. Dialectical Argument Strength: ranking-Based Semantics

In this section we define a ranking-based semantics of dialectical strength of arguments in the form of a preorder on the set of arguments. Dialectical argument strength has both static and dynamic aspects. A static aspect is whether an argument has been successfully defended in a terminated dialogue, which is a matter of applying a notion of contextual strength at termination. Dynamic aspects concern how challengeable an argument is in a given non-final state of the dialogue. Taking the formulation of [15] quoted above in the introduction literally, it should be modelled by considering all possible ways to terminate the dialogue but in general this is infeasible sicne it will often be impossible to foresee which information is available to construct arguments, how they will be evaluated, and which procedural decisions (such as on admissibility of evidence) will be taken.

For these reasons, we propose the following approach. Imagine a dialogue participant who can extend a given AF and who wants to make a given argument F (the focus argument) dialectically as strong as possible. The participant will consider all procedurally allowed expansions AF' of AF and determine in which of these expansions F is the strongest. So in general we have to compare arguments that are in *different* AFs. Moreover, our notion of strength will not boil down to applying a notion of contextual strength to all these expansions, since we also want to determine how vulnerable F is to attack in all these expansions. To this end we will define a notion of ‘attack points’ of an argument, which are minimal sets of arguments that, if attacked in an allowed expansion, make the contextual status of the focus argument decrease.

To model these ideas, we let dialectical strength be determined by a combination of the ‘current’ contextual strength of an argument and its number of attack points as follows. To start with, we assume a ranking of contextual argument statuses, which in the present paper will be that being labelled *in* is better than being labelled *undecided*, which is better than being labelled *out*. In notation: *in* $>_c$ *und* $>_c$ *out*. (In future research this could be extended to alternative semantics, even to gradual ones, but in this paper we prefer to keep things simple to focus on the essence.) Then, given the set of allowed expansions $\{AF', AF'', \ldots\}$ of a given AF, we say that if argument $A_{AF'}$ is contextually better than argument $B_{AF''}$ then it is also dialectically better than argument $B_{AF''}$, while if $A_{AF'}$ and $B_{AF''}$ are contextually equally strong, then $A_{AF'}$ is better than $B_{AF''}$ if $A_{AF'}$ has fewer attack points than $B_{AF''}$. So this notion of dialectical strength presupposes and is a refinement of the notion of contextual strength. The primacy of contextual strength is justified by our intended application scenario, where a proponent of a focus argument F wants to move to a state where F is contextually as strong as possible. Moreover, if contextual strength has primacy, then for terminated disputes dialectical strength reduces as desired to how well an argument is defended at termination.

Consider an example $AF = A \leftarrow B$ and let A be the focus argument. Assume the proponent of A can expand AF with either C, resulting in $AF' = A \leftarrow B \leftarrow C$, or with D, resulting in $AF'' = A \leftarrow B \leftarrow D$. In both expansions A is *in* so contextually of the same strength. However, assume that C is attackable while D is unattackable. Then A has two attack points in AF', namely, $\{A\}$ and $\{C\}$, while A has only one attack point in AF'', namely, $\{A\}$. So A is dialectically stronger in AF'' than in AF', so the dialectically better choice for the proponent is to expand AF to AF'' by moving D.

An attack point must be defined as a *set* of arguments. Consider Figure 1. Attacking just C or just D is not enough to lower the status of A, so one attack point must in this

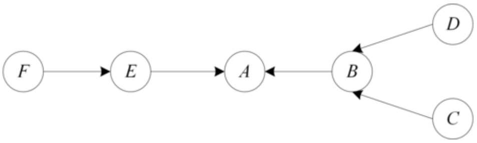


Figure 1. Multiple attack points

case be defined as $\{C, D\}$. Note, furthermore, that attacking F also lowers the status of A, so $\{F\}$ is also an attack point of A, so an argument can have multiple attack points.

To define attack points, we now first define the notion of an allowed expansion of an AF, which is a refinement of [3]'s notion of a normal expansion. The first refinement is to make expansions relative to a given background universal argumentation framework $UAF = (\mathcal{A}^u, \mathcal{C}^u)$. An important reason for doing so is to avoid implicit assumptions at the abstract level that are not always satisfied by instantiations, such as that all arguments are attackable or that all attacks are independent from each other.

Definition 1 Given a universal argumentation framework $UAF = (\mathcal{A}^u, \mathcal{C}^u)$, an *argumentation framework in* UAF is any $AF = (\mathcal{A}, \mathcal{C})$ such that $\mathcal{A} \subseteq \mathcal{A}^u$ and $\mathcal{C} \subseteq \mathcal{C}^u_{|\mathcal{A} \times \mathcal{A}}$.

That $\mathcal{C}$ is not required to equal $\mathcal{C}^u_{|\mathcal{A} \times \mathcal{A}}$ is to allow for instantiations like *ASPIC*$^+$ that use preferences to resolve attacks into defeat relations and let $\mathcal{C}$ stand for defeat.

We must also distinguish between allowed and not allowed expansions. One reason is that the dialogical protocol may impose constraints, such as admissibility of premises or of types of arguments (for example, in some systems of criminal law analogical applications of criminal provisions are not allowed). The problem context may also impose restrictions. For example, investigation procedures in which information gathering is interchanged with argument construction may have a constraint that all and only relevant arguments constructible from the gathered information are included. Finally, underlying structured accounts of argumentation may impose such constraints, for example, a closure constraint on the set $\mathcal{A}'$ of new arguments in that other arguments that can be constructed with information introduced by arguments in $\mathcal{A}'$ must also be in $\mathcal{A}'$.

We now define (allowed) expansions relative to a given UAF as follows.

Definition 2 [Expansions given a universal argumentation framework] Let $AF = (\mathcal{A}, \mathcal{C})$ and AF' be two abstract argumentation frameworks in UAF. Then AF' is an *expansion* of AF given UAF if $AF' = (\mathcal{A} \cup \mathcal{A}', \mathcal{C} \cup \mathcal{C}')$ for some nonempty $\mathcal{A}'$ disjoint from $\mathcal{A}$, such that for all A, B: if $(A, B) \in \mathcal{C}'$ then $A \in \mathcal{A}'$ or $B \in \mathcal{A}'$.

Let UAF^e be the set of all expansions of some AF given UAF. Then $aUAF^e \subseteq UAF^e$ is the set of *allowed expansions* given UAF.

A further refinement is needed. Imagine two attackable but unattacked arguments A and B such that for both of them expansions exist that lower their status. Then they both have one attack point, namely, $\{A\}$, respectively, $\{B\}$. However, if A has just one attackable premise while B has two, or A uses one defeasible rule while B uses two, then A should still be dialectically stronger than B. Accordingly, we assume that each argument A in a UAF comes with a finite set $t(A)$ of *attack targets* and we assume that each argument B attacking A attacks A on at least one of A's attack targets. Given a set S of arguments, we write S^t for the set of all pairs (A, t) such that $A \in S$ and $t \in t(A)$.

Finally, we need a notion of relevance of a set of defenders to the status of the defended argument. It adapts the dialogical notion of relevance proposed in [13] to AFs.

Definition 3 For any $AF = (\mathcal{A}, \mathcal{C})$ with $A \in \mathcal{A}$, a set $S \subseteq A$ is *relevant to* A in AF iff S is a minimal set such that the contextual status of A is lower in $AF' = (\mathcal{A}', \mathcal{C}')$ than in AF, where $\mathcal{A}' = (\mathcal{A} \cup \{X\})$ for some X not in $\mathcal{A}$ and not attacked by $\mathcal{A}$, and $\mathcal{C}' = \mathcal{C} \cup \{(X, B) \mid B \in S\}$.

So S is relevant to A in AF iff AF can be expanded with an unattacked attacker of all members of S such that A's contextual status is lowered. Note that this notion is not defined relative to a UAF. If S is relevant to A then all arguments in S are defenders of A but it can happen that a defender of A is in no set relevant to A. In Figure 2, C and G are defenders of A but attacking either of them does not lower the status of A; this only happens if either A or D is attacked, so the only sets relevant to A are $\{A\}$ and $\{D\}$.

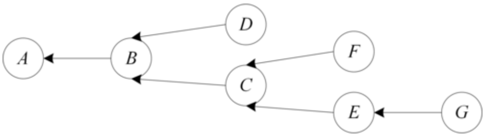


Figure 2. Relevant sets

We are now in the position to define the notion of an attack point of an argument.

Definition 4 [**Attack points**] Given an abstract argumentation framework $AF = (\mathcal{A}, \mathcal{C})$ in UAF, an *attack point* of an argument $A \in \mathcal{A}$ is any minimal set $S \subseteq \mathcal{A}^t$ relevant to A such that an allowed expansion $AF' = (\mathcal{A} \cup \mathcal{A}', \mathcal{C} \cup \mathcal{C}')$ of AF given UAF exists with

1. for all $(B, t) \in S$ there exists an argument $C \in \mathcal{A}'$ such that C attacks B on t;
2. the contextual status of A is lower in AF' than in AF.

The set of attack points of A given AF is denoted by $ap_{AF}(A)$.

It is not required that all arguments in $\mathcal{A}'$ attack some argument in S, since including an attacker of S in $\mathcal{A}'$ might require putting other arguments in $\mathcal{A}'$ as well, such as A's subarguments in systems in which arguments have subarguments. Also, Definition 4 allows for 'side effects' in that the new attackers may also attack arguments outside S or arguments in $\mathcal{A}$ but outside S may attack them. For example, an argument attacking another argument on its premise may also attack all other arguments using that premise.

We can now give our definition of dialectical argument strength, by combining the notion of contextual strength with the number of attack points of arguments. Several definitions are still possible and the ones given by us are not meant to be the final answer but instead to initiate the discussion about what are good definitions. First, we give primacy to the current contextual evaluation in that being contextually stronger implies being dialectically stronger. If two arguments are contextually equally strong, then we refine this ordering by comparing their sets of attack points.

Definition 5 [**Dialectical strength**] Let $AF = (\mathcal{A}, \mathcal{C})$ and $AF' = (\mathcal{A}', \mathcal{C}')$ be two abstract argumentation frameworks in a given UAF and let $A \in \mathcal{A}$ and $B \in \mathcal{A}'$ where the contextual status of A in AF is s and the contextual status of B in AF' is s'. We say that

$A_{AF} \geq_c B_{AF'}$ iff either $s = in$, or $s = und$ and $s' \neq in$, or $s = s' = out$. Moreover, we say that $A_{AF} \geq_d B_{AF'}$ iff

1. $A_{AF} \geq_c B_{AF'}$; and
2. if $B_{AF'} \geq_c A_{AF}$ then $|ap_{AF}(A)| \leq |ap_{AF'}(B)|$.

Below we will leave the subscripts of the arguments implicit if there is no danger of confusion. As usual, $B \leq A$ stands for $A \geq B$ while $A > B$ stands for $A \geq B$ and $B \not\geq A$, and $A \approx B$ stands for $A \geq B$ and $B \geq A$.

The condition that an attack point of A is relevant to A is to exclude examples like an AF with unattacked A and B which are both attacked by the same argument C from UAF: without the relevance condition and if expanding AF with C is allowed, then $\{(B,t)\}$ would (for a given t) be an attack point of A, which is undesirable.

We now illustrate the definition with the AFs in Figure 3. Many current gradual accounts regard A_{AF1} as stronger than A_{AF2} based on the intuition that having no attackers is better than having attackers (the principle of Void Precedence discussed in the next section). In our approach, this depends on several things. Suppose first that all of A, B and C have attackers in UAF, that A and C have one attack target, respectively, t and t', that A has other unattacked attackers in UAF besides B, and that all expansions are allowed. Then A_{AF1} has just one attack point, namely, $\{(A,t)\}$, while A_{AF2} has two attack points, namely, $\{(A,t),(C,t')\}$. So in this case having no attackers is better.

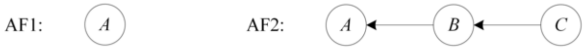


Figure 3. The reinstatement pattern

However, assume now that A has no other attackers in UAF besides B, or that A does have other attackers in UAF but that no expansion with these other attackers is allowed, perhaps for efficiency reasons. In both cases A_{AF1} still has the single attack point $\{(A,t)\}$ but A_{AF2} now also has just one attack point, namely, $\{(C,t')\}$. So here having no attackers is not better than having attackers.

Finally, suppose we change this variation by letting C have no attackers in UAF. Then A_{AF2} has no attack points, so we have a case where an argument that has attackers in one AF is better than an argument that has no attackers in another AF. In conclusion, whether having no attackers is better than having attackers depends on the nature of the arguments and their relations and on the context in which they are evaluated.

4. Properties of Dialectical Argument Strength

We now investigate some properties of our definition of dialectical argument strength. First, $\geq_d$ is a total preorder, that is, transitive and reflexive.

Proposition 1 For all arguments A, B, C and argumentation frameworks AF, AF' and AF'' in a given UAF:

1. $A_{AF} \leq_d A_{AF}$
2. If $A_{AF} \leq_d B_{AF'}$ and $B_{AF'} \leq_d C_{AF''}$ then $A_{AF} \leq_d C_{AF''}$

PROOF. (Sketch:) (1) is immediate, while (2) follows from the facts that both $\leq_c$ and a cardinality ordering on sets are total and that if $A_{AF} \approx_c B_{AF'}$, then a further comparison is made in terms of the cardinality of sets. QED

Definition 6 A UAF satisfies the *attack property* iff for all arguments A, B and C in UAF and all attack targets t that are shared by A and B it holds that C attacks A on t iff C attacks B on t.

The attack property is, for instance, satisfied by assumption-based argumentation in general and by *ASPIC*$^+$ for the case with so-called reasonable argument orderings.

Proposition 2 Consider any UAF satisfying the attack property and let AF be an argumentation framework in UAF containing arguments A and B. Then if $t(A) \subseteq t(B)$ then $A_{AF} \geq_d B_{AF}$.

PROOF. Suppose for contradiction that $A <_d B$ and suppose first that $A <_c B$. If B is *in* but A is not *in* then there exists an attacker C of A that is not *out*. But then C also attacks B so B is not *in*. If B is undecided and A is *out* then there exists an attacker C of A that is *in*. But then C also attacks B so B is *out*. Contradiction.

Suppose next that $A \approx_c B$ and suppose for contradiction that there exists an attack point of A that is not also an attack point of B. This implies that A is not *out* in AF.

Suppose first that A is *in*. Then there exists an allowed expansion AF' of AF where some attackers of A in AF are not *out* and which make that A is not *in* in AF'. By the attack property, these attackers are also attackers of B, so B is not *in* in AF', so the attack point of A is also an attack point of B. Contradiction.

Suppose next that A is *und*. Then there exists an allowed expansion AF' of AF were some attackers of A are *in* and make that A is *out* in AF'. By the attack property, these attackers are also attackers of B, so B is *out* in AF', so the attack point of A also is an attack point of B. Contradiction.

QED

Proposition 2 is what one would expect from dialectical strength as degree of attackability. Its more general version where A and B can be from different AFs does not hold. A counterexample is displayed in Figure 4. Here $\{(A, t)\}$ is (for a given t) an attack point of A in AF_1 but not in AF_2 since B protects A in AF_2 against an expansion with C.

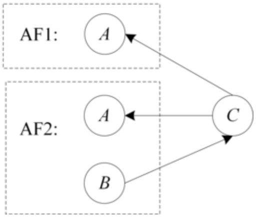


Figure 4. Counterexample to general version of Proposition 2

This illustrates that for dialectical strength the dynamic context is important.

Technically our proposal is in the class of ranking-based semantics. We therefore next investigate principles proposed in the literature on ranking-based semantics, basing ourselves on [5]. However, we should first discuss the possible objection that these principles were never intended for dialectical strength, so that investigating them would for present purposes be irrelevant. Against this, it should first be noted that authors are generally not explicit about the kind of strength for which their principles are intended. Moreover, some principles compare different AFs, just as our notion of dialectical strength does, so their underlying intuitions might involve dialectical elements. For these reasons it still makes sense to investigate whether the principles proposed in the literature are suitable for notions of dialectical argument strength. For cases where the underlying intuitions of the proposed principles are not made explicit, our investigation will reveal to which extent they can be based on intuitions concerning dialectical strength.

For reasons of space we have to present the principles discussed in [5] semiformally and we cannot (fully) discuss all of them. When giving counterexamples, we can assume that all considered expansions are allowed.

Proposition 3 Of all principles discussed by [5], Definition 5 only satisfies Attack vs Full Defense and Total.

PROOF. **Total** says that $\leq_d$ is a total ordering. This is stated by Proposition 1. **Attack vs Full Defense** says for acyclic AFs that an argument without any attack branch is ranked higher than an argument only attacked by one non-attacked argument. This holds since any argument of the former kind is *in* while any argument of the latter kind is *out*. QED

For reasons of space we can only give counterexamples to some of the other properties.

Abstraction says that different AFs of the same form should evaluate arguments having the same structural relations in the AFs equally. For a counterexample, consider AF_1 with just A and having one attack target and AF_2 with just B and having two attack targets, where UAF contains additional arguments making that all three attack targets induce the corresponding singleton set attack point. Abstraction says that A and B are of the same rank but we have $A >_d B$. Even if all arguments have the same number of attack targets, there are counterexamples. Assume that both A and B have one attack target and that UAF contains an attacker of B but not of A. Then we again have $A >_d B$.

Void precedence says that a non-attacked argument is ranked strictly higher than any attacked argument in the same AF. One counterexample was given Section 3. Another counterexample is figure 5,which depicts a UAF with an AF in UAF contained in the dotted box. Assume all arguments have a single attack target. Then A has attack point $\{(A, t)\}$ but A' has no attack points, since B protects A' against expanding AF with D attacking A.

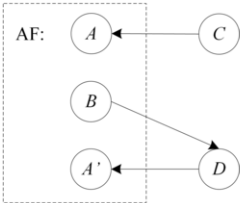


Figure 5. Counterexample to Void Precedence

For two principles that do not hold in general we have identified a special case in which they hold. The **Quality Preference** principle says that if there exists an attacker C of B such that for all attackers D of A it holds that $C >_d D$, then $A >_d B$. It holds in the following special case, since then we have $A >_d B$ so $A >_c B$.

Proposition 4 Def. 5 satisfies Quality Preference if $C >_d D$ for all D attacking A.

A weak version holds of Defence Precedence, which we call **Weak Defense Precedence**, saying that if A_{AF} and B_{AF} have the same number of attackers in AF but A_{AF} has direct defenders while B_{AF} has no direct defender, then $A_{AF} \not<_c B_{AF}$. This holds since an attacked argument with no defenders is always *out*.

Proposition 5 Definition 5 satisfies Weak Defence Precedence.

Why do most principles fail to hold? This is for two main reasons. They fail since they just consider the topology of an AF while dialectical strength also depends on the dynamic context in which an AF can evolve, and/or they fail since they make implicit assumptions on the nature of arguments and their relations that do not hold in general, such as that all arguments have an equal number of attack targets.

5. Dialectical Argument Strength: semantics for weighted AFs

We next adapt our approach to so-called weighted argumentation frameworks [1].

Definition 7 A *weighted argumentation framework* wAF is a triple $(\mathcal{A}_{wAF}, w_{wAF}, \mathcal{C}_{wAF})$ where $\mathcal{A}_{wAF}$ and $\mathcal{C}_{wAF}$ are defined as for AFs and w_{wAF} is a function from $\mathcal{A}_{wAF}$ into $[0, 1]$. A *semantics* for a wAF is another function s_{wAF} from $\mathcal{A}$ into $[0, 1]$.

As above, we omit the subscripts if they are clear from the context. Now a *weighted universal argumentation framework* $wUAF$ is a triple $(\mathcal{A}^u, w^u, \mathcal{C}^u)$, and an AF in UAF is any $wAF = (\mathcal{A}, w, \mathcal{C})$ such that $\mathcal{A} \subseteq \mathcal{A}^u$ and $w = w^u_{|\mathcal{A}}$ and $\mathcal{C} \subseteq \mathcal{C}^u_{|\mathcal{A}\times\mathcal{A}}$. The other of the above definitions for AFs then also apply to $wAFs$ by ignoring w.

We define an argument's weight in a $wUAF$ in terms of its number of attack targets:

$$w_{wUAF}(A) = \frac{1}{1+ \mid t(A) \mid}$$

Note that all weights are between 0 and 1 and that an argument without attack targets has weight 1. We next redefine dialectical argument strength for $wAFs$ as follows.

Definition 8 [**Dialectical argument strength with weights**] An argument's *attack point degree* is defined as $d_{wAF}(A) = \frac{1}{1+|ap_{wAF}A|}$. Then $s_{wAF}(A)$ is defined as follows:

- if A is *in* then $s_{wAF}(A) = \frac{d_{wAF}(A)}{2} + 0.5$;
- if A is *und* then $s_{wAF}(A) = \frac{d_{wAF}(A)}{2}$;
- if A is *out* then $s_{wAF}(A) = 0$.

It can be shown that Definition 8 induces the same ranking on arguments as Definition 5.

Lemma 6 For any wAF and any argument $A \in wAF$ it holds that A is *in* iff $s_{wAF}(a) > 0.5$; A is *und* iff $0 < s_{wAF}(a) \leq 0.5$; and A is *out* iff $s_{wAF}(a) = 0$.

Lemma 7 Let AF and AF' be equal to wAF and wAF' but without weight functions. Then if $A_{AF} \approx_c B_{AF'}$ then $|ap_{AF}(A)| \leq |ap_{AF'}(B)|$ iff $s_{wAF}(A) \leq s_{wAF'}(B)$.

Proposition 8 Let wAF and wAF' be $wAFs$, $A \in wAF$ and $B \in wAF'$ and let AF and AF' be equal to w_{wAF} and $w_{wAF'}$ but without weight functions. Then $A \leq_d B$ iff $s_{wAF}(A) \leq s_{wAF'}(B)$.

PROOF. For the only-if part assume $A \leq_d B$. Two cases must be considered. If $A <_c B$ then $s_{wAF}(A) < s_{wAF'}(B)$ by Lemma 6. If $A \approx_c B$ then $|ap_{AF}(A)| \leq |ap_{AF}(B)|$ so $s_{wAF}(A) \leq s_{wAF'}(B)$ by Lemma 7.

For the if-part assume $s_{wAF}(A) \leq s_{wAF'}(B)$. Two cases must be considered. If $A <_c B$ then $A <_d B$ so $A \leq_d B$. If $A \approx_c B$ then by Lemma 7 we have $|ap_{AF}(A)| \leq |ap_{AF'}(B)|$ so $A \leq_d B$. QED

We next investigate the principles proposed in the literature for semantics of weighted AFs, basing ourselves on [1]. As for ranking-based semantics, for space limitations we cannot discuss all principles while their presentation has to be semiformal.

Proposition 9 Of all principles discussed by [1], Definition 8 only satisfies Weakening Soundness and Compensation.

PROOF. **Weakening soundness** says that for any wAF and any $A \in wAF$, if $s_{wAF}(A) < w_{wAF}(A)$ then there exists an attacker $B \in wAF$ such that $s_{wAF}(B) > 0$. We prove this by contraposition. If there exists no such B, then all attackers of A are *out*. But then A is *in* in wAF. Then $|ap_{wAF}(A) \leq t(A)|$, so $d_{wAF}(A) \not< w_{wAF}(A)$. But then $s_{wAF}(A) \not< w_{wAF}(A)$.

Compensation says that there exist wAF in which more weak attackers compensate for fewer stronger attackers. The proof has to specify just one such wAF. Figure 6 displays a wAF with the number of attack targets of each argument indicated. Assume that all attack targets are an attack point since they have an attacker in UAF (not shown). Note that all arguments are *und*. Then $s_{wAF}(C) = s_{wAF}(D) = \frac{1}{6}$ and $s_{wAF}(E) = \frac{1}{4}$. So A has more attackers with nonzero strength than B while B has an attacker that is stronger than all attackers of A. Moreover, $s_{wAF}(A) = s_{wAF}(B) = \frac{1}{4} > 0$. QED

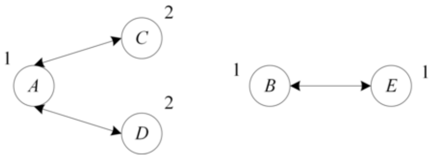


Figure 6. Proof of Compensation

Counterexamples to the other principles can be constructed as for Definition 5 by considering the context of wAF as defined by $wUAF$ or by considering arguments with sets of attack points of different cardinality. Consider **Monotony**, which says that, for any

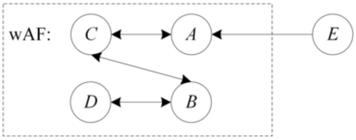


Figure 7. Counterexample to Monotony

$A, B \in wAF$, if $w_{wAF}(A) = w_{wAF}(B)$ and all attackers of A in wAF are attackers of B in wAF, then $s_{wAF}(A) \geq s_{wAF}(B)$. A counterexample is displayed in Figure 7. Here $\{(\mathcal{A}, t)\}$ is an attack point of A since expanding AF with E makes A *out* but no expansion makes B *out*. However, monotony does hold for a special case:

Proposition 10 Monotony holds if all attackers of A in $wUAF$ are attackers of B in $wUAF$ and $wUAF$ satisfies the attack property.

6. Related Research

We do not know of earlier formal work that explicitly addresses dialectical argument strength. Arguably, work on enforcing, preserving or realising a particular argument status [3,8,4], does so implicitly. Compared to this work, we are interested in how the acceptability status of an argument can decrease. A recent structured approach in *ASPIC*$^+$ is [12] (abstracted to AFs in [11]), who study whether argument and conclusion statuses can change under expansions of the knowledge base, to find out whether searching for further information makes sense. It would be interesting to investigate how all this work on argument dynamics can be combined with studies of dialectical argument strength.

As noted above, most work on gradual acceptability does not indicate which aspect(s) of argument strength is or are modelled. A recent exception is [6], who model two aspects of "persuasiveness'," i.e., of rhetorical strength. The first is *procatalepsis*, the attempt of a speaker to strengthen their argument by dealing with possible counterarguments before the audience can raise them. The second aspect is *fading*, the phenomenon that long lines of argumentation are less persuasive. Bonzon et al. claim that "current ranking-based semantics are poorly equipped to be used in a context of persuasion". Among other things, they show that procatalepsis violates the Void Precedence principle. While we agree with their observation, we note that in the end they do not give a separate model of persuasiveness but combine these two aspects with existing strength principles into an overall measure of argument strength, thereby still conflating the three kinds of argument strength. We instead prefer to separately study different notions of argument strength, since these notions may serve different purposes and may therefore evaluate the same arguments differently.

7. Conclusion

In this paper we presented the first formal study of dialectical argument strength, modelled as the number of ways in which an argument can be successfully attacked in expansions of an argumentation framework. We showed that most principles for gradual argument acceptability proposed in the literature fail to hold for the new notion, which reveals something about the possible rational foundations of these principles and highlights the importance of distinguishing between kinds of argument strength. Our model is abstract but its design is motivated by the wish to avoid overly limiting assumptions on dialogue contexts or the structure of arguments and their relations.

Are our partly negative results on satisfaction of the principles bad for our approach or for the principles? There is no easy answer to this question but we note that in the literature most principles are based on intuitions instead of on philosophical insights. Therefore it is not obvious why they should hold; it may just as well be that if a semantics based on philosophical insights and arguably reflecting good properties does not satisfy some principle, then this indicates that the principle may not be suitable for the modelled notion. Our semantics is based on [15] and arguably satisfies desirable properties. In particular, we believe that Proposition 2 and the satisfaction of Weakening Soundness and the special case of Monotony indicate that our semantics captures the ideas of [15] and the intuition that justifying a decision more sparsely is better.

In future research we want to extend our abstract model with support relations between arguments and to study structured instantiations of our model and applications to particular dialogue contexts. We also want to extend our approach to semantics other than grounded semantics, including gradual semantics.

References

[1] L. Amgoud, D. Doder, and S. Vesic. Evaluation of argument strength in attack graphs: foundations and semantics. *Artificial Intelligence*, 302:103607, 2022.

[2] P. Baroni, M. Caminada, and M. Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26:365–410, 2011.

[3] R. Baumann and G. Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. In *Proceedings of COMMA 2010*, pages 75–86, 2010.

[4] D. Baumeister, M. Järvisalo, D. Neugebauer, A. Niskanen, and J. Rothe. Acceptance in incomplete argumentation frameworks. *Artificial Intelligence*, 295:103470, 2021.

[5] E. Bonzon, J. Delobelle, S. Konieczny, and N. Maudet. A comparative study of ranking-based semantics for abstract argumentation. In *Proceedings of AAAI 2016*, pages 914–920, 2016.

[6] E. Bonzon, J. Delobelle, S. Konieczny, and N. Maudet. A parametrized ranking-based semantics compatible with persuasion principles. *Argument and Computation*, 12:49–85, 2021.

[7] C. Cayrol and M.-C. Lagasquie-Schiex. Bipolar abstract argumentation systems. In I. Rahwan and G.R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 65–84. Springer, Berlin, 2009.

[8] S. Doutre and J.-G. Mailly. Constraints and changes: A survey of abstract argumentation dynamics. *Argument and Computation*, 9:223–248, 2018.

[9] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and *n*–person games. *Artificial Intelligence*, 77:321–357, 1995.

[10] A.J. Hunter, editor. *Argument and Computation*, volume 5. 2014. Special issue with Tutorials on Structured Argumentation.

[11] J.-G. Mailly and J. Rossit. Stability in abstract argumentation. In *Proceedings of the 18th International Workshop on Nonmonotonic Reasoning*, pages 93–99, 2020.

[12] D. Odekerken, A. M. Borg, and F.J. Bex. Estimating stability for efficient argument-based inquiry. In *Proceedings of COMMA 2020*, pages 307–318, 2020.

[13] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15:1009–1040, 2005.

[14] H. Prakken. Philosophical reflections on argument strength and gradual acceptability. In *Proceedings of ECSQARU 2021*, pages 144–158, 2021.

[15] F. Zenker, K. Debowska-Kozlowska, D. Godden, M. Selinger, and S. Wells. Five approaches to argument strength: probabilistic, dialectical, structural, empirical, and computational. In *Proceedings of the 3rd European Conference on Argumentation*, pages 653–674, London, 2020. College Publications.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220162

Ordinal Conditional Functions for Abstract Argumentation

Kenneth SKIBA and Matthias THIMM
Artificial Intelligence Group, University of Hagen, Germany

Abstract. We interpret and formalise *ordinal conditional functions (OCFs)* in abstract argumentation frameworks based on ideas and concepts defined for *conditional logics*. There, these functions are used to rank interpretations, and we adapt them to rank extensions instead. Using conflict-freeness and admissibility as two essential principles to define the semantics of OCFs, we obtain a framework that allows to rank sets of arguments wrt. their plausibility. We analyse the properties of this framework in-depth, and in doing so we establish a formal bridge between the approaches of abstract argumentation and conditional logics.

Keywords. Abstract Argumentation, Ranking Functions

1. Introduction

Abstract argumentation frameworks (AF) [1] have gathered research interest as a model for rational decision-making in presence of conflicting information. Using AFs, *arguments* and *attacks* can be represented as nodes and edges, respectively, of a directed graph. In order to reason over AFs *extension semantics* were defined, which are functions such that a set of arguments can be considered jointly acceptable. Recently Skiba et al. [2] generalised this reasoning process to rank sets of arguments based on their plausibility. Another used reasoning formalism is *conditional logic*, which studies conditionals like “if A then B” written as $(B \mid A)$. So given the information that A is true it is more “believable” that B is true, than B being not true. In order to define a value of believability, *ordinal conditional functions* (OCF) (also known as *ranking functions*) were defined [3]. These functions can be used to rank possible worlds according to their plausibility. One example of an OCF is the *System Z* ranking function [4], which exhibits particularly good reasoning properties.

In recent years, the relationship between argumentation and conditional logic was investigated in [5,6,7,8] and by Weydert [9,10]. While abstract argumentation usually only provides a criterion to determine whether a set of arguments is jointly accepted or not, OCFs on the other hand can rank possible worlds according to their plausibility. In this paper, we want to use these ideas from conditional logic to reason in abstract argumentation. Our goal is to rank sets of arguments according to their plausibility, i. e., we can state not only whether a set of arguments is accepted or not, but also state that a set is more plausible than another one. In particular, we can rank sets of arguments, which are not jointly acceptable w.r.t. extension semantics, for example, we can say that out of two conflicting sets one of them is more plausible. One potential application of such a ranking is decision-making in presence of constraints, where a solution (represented as a set of arguments) satisfies constraints that cannot be satisfied by a set of arguments under extension semantics. One possible way to still make a decision would be to select the most plausible sets of arguments, which are satisfying the constraints.

To achieve such a ranking of sets of arguments, we will use the guiding principles of admissible reasoning for abstract argumentation frameworks namely *conflict-freeness* and *admissibility* to develop ordinal conditional functions for abstract argumentation. In order to connect abstract argumentation and conditional logics we interpret the set of attacks as a set conditionals. Since there can be a number of functions satisfying the defined principles, we develop a model-based reasoning technique inspired by System Z. This System Z ranking function allows us to model plausibility values for each set of arguments I being *in* while a different set O is *out*. These plausibility values can be used to rank sets of arguments, and therefore continue recent work about extension-ranking semantics started in [2].

This paper is structured as follows. We recall all necessary preliminaries about abstract argumentation and conditional logics in Section 2. Section 3 introduces OCFs for abstract argumentation. In Section 4, we look at OCFs based on System Z. A extension-ranking semantics is introduced in Section 5 as well as an in-depth investigation of the properties of that semantics is presented. We conclude this paper in Section 6 with a discussion about related work.

2. Preliminaries

In this section, we recall all necessary definitions of abstract argumentation and conditional logics.

2.1. Abstract Argumentation

Argumentation frameworks [1] are a formalism that allows the representation of conflicts between pieces of information using arguments and attacks between arguments.

Definition 1. An *abstract argumentation framework* (AF) is a directed graph $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ where A is a finite set of *arguments* and R is an *attack relation* $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$.

An argument a is said to *attack* an argument b if $(a,b) \in \mathsf{R}$. We say that an argument a is *defended by a set* $S \subseteq \mathsf{A}$ if every argument $b \in \mathsf{A}$ that attacks a is attacked by some $c \in S$. For $a \in \mathsf{A}$ define $a^- = \{b \mid (b,a) \in \mathsf{R}\}$ and $a^+ = \{b \mid (a,b) \in \mathsf{R}\}$, so the set of attackers of a and the set of arguments attacked by a. For a set of arguments $S \subseteq \mathsf{A}$ we extend these definitions to S^+ and S^- via $S^+ = \bigcup_{a \in S} a^+$ and $S^- = \bigcup_{a \in S} a^-$, respectively. For two graphs $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $\mathsf{AF}' = (\mathsf{A}', \mathsf{R}')$, we define $\mathsf{AF} \cup \mathsf{AF}' = (\mathsf{A} \cup \mathsf{A}', \mathsf{R} \cup \mathsf{R}')$.

To reason with abstract argumentation frameworks a number of different semantical notions have been developed, like the *extension-based* or the *labelling-based* approaches, for an overview see [11]. Both these approaches are handling sets of arguments, which can be considered jointly acceptable. The extension-based semantics are relying on two basic concepts: *conflict-freeness* and *defence*.

Definition 2. Given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, a set $E \subseteq \mathsf{A}$ is:

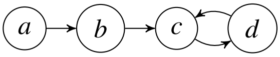


Figure 1. Abstract argumentation framework AF from Example 1.

- *conflict-free* iff $\forall a,b \in E$, $(a,b) \notin \mathsf{R}$;
- *admissible* iff it is conflict-free, and it defends its elements.

We use $\mathrm{cf}(\mathsf{AF})$ and $\mathrm{ad}(\mathsf{AF})$ for denoting the sets of conflict-free and admissible sets of an argumentation framework AF, respectively. The semantics proposed by Dung [1] are then defined as follows.

Definition 3. Given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, an admissible set $E \subseteq \mathsf{A}$ is a *complete* extension (co) iff it contains every argument that it defends; a *preferred* extension (pr) iff it is a $\subseteq$-maximal complete extension; the unique *grounded* extension (gr) iff it is the $\subseteq$-minimal complete extension; a *stable* extension (st) iff $E^+ = \mathsf{A} \setminus E$.

The sets of extensions of an argumentation framework AF, for these four semantics, are denoted (respectively) $\mathrm{co}(\mathsf{AF})$, $\mathrm{pr}(\mathsf{AF})$, $\mathrm{gr}(\mathsf{AF})$ and $\mathrm{st}(\mathsf{AF})$. Based on these semantics, we can define the status of any (set of) argument(s), namely *skeptically accepted* (belonging to each σ-extension), *credulously accepted* (belonging to some σ-extension) and *rejected* (belonging to no σ-extension). Given an argumentation framework AF and an extension semantics σ, we use (respectively) $\mathrm{sk}_\sigma(\mathsf{AF})$, $\mathrm{cr}_\sigma(\mathsf{AF})$ and $\mathrm{rej}_\sigma(\mathsf{AF})$ to denote these sets of arguments.

Example 1. Consider the argumentation framework $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ depicted as a directed graph in Figure 1, with the nodes corresponding to the arguments $\mathsf{A} = \{a,b,c,d\}$, and the edges corresponding to the attacks $\mathsf{R} = \{(a,b),(b,c),(c,d),(d,c)\}$. The sets $\{a\}$, $\{a,c\}$ and $\{a,d\}$ are complete extensions of AF, while only $\{a,c\}$ and $\{a,d\}$ are stable.

For more details about these semantics (and other ones defined in the literature), we refer the interested reader to [1,11].

2.2. Conditional Logics

In order to define the usual *propositional language* $\mathscr{L}(At)$ over At we use a set of atoms At and connectives $\wedge$ (and), $\vee$ (or) and $\neg$ (negation). The function $\omega : At \to \{T,F\}$ defines an *interpretation* (or *possible world*) ω for $\mathscr{L}(At)$. $\Omega(At)$ denotes the set of all interpretations. An interpretation ω *satisfies* an atom $a \in At$ ($\omega \models a$), iff $\omega(a) = T$. As a *conditional* we consider structures like $(\psi \mid \phi)$, which can be read as “if ϕ then (usually) ψ”. Informally speaking, an interpretation ω *verifies* a conditional $(\psi \mid \phi)$ iff it satisfies both antecedent (ϕ) and conclusion (ψ) ($(\psi \mid \phi)(\omega) = 1$); it *falsifies* iff it satisfies the antecedent but not the conclusion ($(\psi \mid \phi)(\omega) = 0$); otherwise the conditional is *not applicable*. A conditional is satisfied by ω if it does not falsify it.

We use *ordinal conditional functions (OCFs)* (also called *ranking functions*) [3], $\kappa : \Omega(At) \to \mathbb{N} \cup \{\infty\}$ to denote a plausibility degree of interpretations and define $\kappa(\phi) := min\{\kappa(\omega) \mid \omega \models \phi\}$. An OCF κ *satisfies* a set Δ of conditionals, if for each $(\psi \mid \phi) \in \Delta$, $\kappa(\phi \wedge \psi) < \kappa(\phi \wedge \neg\psi)$, i. e., verifying a conditional is more plausible than falsifying it. Since the set of all satisfying OCFs may be difficult to handle, one usually relies on model-based inference for reasoning. In this paper, we will focus on the System Z ranking function [4] as an example for model-based inference.

Definition 4. A conditional $(\psi \mid \phi)$ is *tolerated* by a finite set of conditionals Δ if there is a possible world ω, which verifies $(\psi \mid \phi)$ and does not falsify any conditional $(\psi' \mid \phi') \in \Delta$. The *Z-Partitioning* $(\Delta_0, ..., \Delta_n)$ of Δ is defined as:

- $\Delta_0 = \{\delta \in \Delta \mid \Delta \text{ tolerates } \delta\}$
- $\Delta_1, ..., \Delta_n$ is the Z-Partitioning of $\Delta \setminus \Delta_0$

For $\delta \in \Delta : Z_\Delta(\delta) = i$ iff $\delta \in \Delta_i$ and $(\delta_0, ..., \delta_n)$ is the Z-Partitioning of Δ. We define a ranking function κ^Z_Δ via $\kappa^Z_\Delta(\omega) = max\{Z_\Delta(\delta) \mid \delta(\omega) = 0, \delta \in \Delta\} + 1$, with $max\, \emptyset = -1$.

Example 2 ([4]). Consider the following set of conditionals Δ about the flying ability of penguins.

δ_1: "birds fly" $(f \mid b)$ δ_2: "penguins are birds" $(b \mid p)$
δ_3: "penguins do not fly" $(\neg f \mid p)$

The Z-Partitioning of Δ is $\Delta_0 = \{\delta_1\}$ and $\Delta_1 = \{\delta_2, \delta_3\}$, because Δ_0 can be tolerated by all conditionals, while δ_2 and δ_3 cannot be tolerated by Δ. We can calculate the plausibility value of interpretations ω. For example, a flying penguin ($\omega(p) = \omega(b) = \omega(f) = T$) receives a value of $\kappa^Z_\Delta(\omega) = 1$.

3. Ordinal Conditional Functions for Abstract Argumentation

In this section we define OCFs for abstract argumentation. We define a function κ with two parameters (I and O) to calculate a numerical plausibility value. These parameter are sets of arguments where the first set I is considered *in*, and the second set is considered *out*. So $\kappa(I,O) = 0$ means that the set I being *in* and the set O being *out* is not surprising. Note that our OCF need two parameters instead of only one, like in conditional logics, since abstract argumentation is missing the notion of negation.

Definition 5. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an AF. A *OCF* for AF is a function $\kappa : 2^{\mathsf{A}} \to \mathbb{N} \cup \{\infty\}$ with $\kappa^{-1}(0) \neq \emptyset$.

For sets $I, O \subseteq \mathsf{A}$ we abbreviate

$$\kappa(I,O) = min\{\kappa(S) | I \subseteq S, S \cap O = \emptyset\}$$
$$\kappa(I,O) = \infty \text{ if } I \cap O \neq \emptyset$$

Example 3. Consider $\mathsf{AF}_2 = (\{a,b\}, \{(a,b)\})$. One exemplary OCF $\kappa^C(I,O)$ returns the number of conflicts in I, i. e., for $\{a,b\}$ $\kappa^C(\{a,b\}, \emptyset) = 1$. For any other set $S \subset \{a,b\}$ like $S = \{a\}$ we have $\kappa^C(S, \emptyset) = \kappa^C(S, \{b\}) = 0$ and $\kappa^C(S,S) = \infty$.

The following definitions are inspired by OCFs in conditional logic. However, while conditional logic semantics follow a single principle regarding conditional acceptance ("a conditional is accepted if its verification is more plausible than its violation"), for admissible reasoning in abstract argumentation we have two guiding principles:

- An argument should not be accepted if one of its attackers is accepted.

i	$\kappa^{-1}(i)$
3	$(\{a,b\},\emptyset)$
2	$(\emptyset,\{a\})$, $(\emptyset,\{b\})$, $(\emptyset,\{a,b\})$
1	$(\emptyset,\emptyset)$, $(\{b\},\{a\})$, $(\{b\},\emptyset)$
0	$(\{a\},\emptyset)$, $(\{a\},\{b\})$

Table 1. Example OCF for Example 4. Note κ is only partially defined.

- An argument should be accepted if all its attackers are not accepted.

The first principle is also called *conflict-freeness*, i. e., a set does not contain two arguments, which share an attack. So conflicting sets are less plausible than conflict-free sets. The second principle is *admissibility*, so a set, which defends itself from all possible attackers, is at least as plausible as set not defending itself. Implementing these two principles for OCFs gives us:

Definition 6. Let $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ be an AF and $a,b \in \mathsf{A}$.

- An OCF κ *accepts* an attack (a,b) with $a \neq b$ if $\kappa(\{a\},\{b\}) < \kappa(\{a,b\},\emptyset)$.
- An OCF κ *possibly reinstates* an argument $a \in \mathsf{A}$ if $\kappa(S \cup \{a\}, a^-) \leq \kappa(S,\{a\} \cup a^-)$ for all $S \subseteq \mathsf{A}$ with $S \cap (a^- \cup a^+) = \emptyset$.

Intuitively, for an OCF to accept an attack (a,b) means that it is more plausible that argument a is *in* and b is *out*, than both a and b being *in* at the same time. For an OCF to possibly reinstate an argument a means that if all attackers of a are *out*, then it is at least as plausible that a is *in* than *out*.

Next we want to denote when an AF is satisfied by an OCF, i.e. when we can define an OCF satisfying all principles defined above for an AF.

Definition 7. An OCF κ *satisfies* an argumentation framework $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ if it accepts all attacks in R and possibly reinstates all arguments in A.

Example 4. Consider $\mathsf{AF}_2 = (\{a,b\},\{(a,b)\})$. So the following statements have to hold:

1. $\kappa(\{a\},\{b\}) < \kappa(\{a,b\},\emptyset)$
2. $\kappa(\{a\},\emptyset) \leq \kappa(\emptyset,\{a\})$
3. $\kappa(\{b\},\{a\}) \leq \kappa(\emptyset,\{a,b\})$

Table 1 depicts an OCF that satisfies AF_2.

Note that if an AF contains a self-attacking argument a, then there is no OCF that satisfies it. Because to accept attack (a,a) it has to hold that $\kappa(\{a\},\{a\}) < \kappa(\{a\},\emptyset)$, which is impossible, since $\kappa(\{a\},\{a\}) = \infty$.

4. The System Z Ranking Function for Abstract Argumentation

In this section, we want to define an OCF inspired by System Z. The basic idea of System Z is that a conditional $(B \mid A)$ is *tolerated* by a set of conditionals, if it is confirmed by a world ω and no other conditional is refuted. In our setting of abstract argumentation we interpret an attack from argument a to argument b as the conditional relationship "if a is acceptable then b should not be acceptable". So the whole attack relation can be interpreted as a set of conditionals. To tolerate an attack, we have to find a set of arguments, which verifies an attack while not violating any other attack. In addition, we use a similar idea to the admissible semantics from Dung. Recall, a set is admissible iff all arguments contained in the set are defended by the set. We add another condition for a set S to tolerate an attack, namely *attack admissibility*, which states that if all attackers of an argument are not in S, then this argument should be included in S.

We begin with defining, when an attack is satisfied by a set S.

Definition 8. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an argumentation framework.

- A set $S \subseteq \mathsf{A}$ *verifies* an attack (a,b) iff $a \in S$ and $b \notin S$.
- A set $S \subseteq \mathsf{A}$ *violates* an attack (a,b) iff $a \in S$ and $b \in S$.
- A set $S \subseteq \mathsf{A}$ *satisfies* an attack (a,b) iff it does not violate it.

Intuitively speaking, a set satisfies an attack if this set does not contain any two conflicting arguments. So for an AF $\mathsf{AF}_3 = (\{a,b,c\}, \{(a,b),(b,c)\})$, we can observe, that the set $S_1 = \{a\}$ verifies the attack (a,b) and does not violate the attack (b,c), while the set $S_2 = \{a,b\}$ verifies the attack (b,c), however S_2 violates attack (a,b).

To satisfy attack admissibility of an argument, we know that, if all the attackers of the argument are *out*, then the argument itself should be *in*.

Definition 9. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an argumentation framework.

- A set $S \subseteq \mathsf{A}$ *verifies attack admissibility* of $a \in \mathsf{A}$ iff $a \in S$ and $b \notin S$ for all $b \in a^-$.
- A set $S \subseteq \mathsf{A}$ *violates attack admissibility* of $a \in \mathsf{A}$ iff $a \notin S$ and $b \notin S$ for all $b \in a^-$.
- A set $S \subseteq \mathsf{A}$ *satisfies attack admissibility* of $a \in \mathsf{A}$ iff it does not violate it.

We recall $\mathsf{AF}_3 = (\{a,b,c\}, \{(a,b),(b,c)\})$, we see that the set $S_3 = \{a,c\}$ verifies attack admissibility of argument c, because the only attacker of c, b is not part of S_3 and one of b's attackers is contained in S_3.

Now we combine both these definitions and define when an attack can be tolerated.

Definition 10. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an argumentation framework. A set $P \subseteq \mathsf{R}$ *tolerates* an attack (a,b) iff there is a set $S \subseteq \mathsf{A}$ that

1. verifies (a,b),
2. satisfies each attack in P, and
3. satisfies attack admissibility of each $c \in \mathsf{A}$

So in order to tolerate an attack, we need to find a set of arguments S s.t. S is conflict-free and every argument contained in S has to be defended. Recall $\mathsf{AF}_3 = (\{a,b,c\}, \{(a,b),(b,c)\})$, then the attack (b,c) is not tolerated by $\{(a,b),(b,c)\}$. Because, for (b,c) to be verified for any set S, it has to hold that $b \in S$. Then to not violate (a,b) a is not allowed to be contained in S. However, then we have the problem that S does not contain any attackers of a, meaning that attack admissibility of a is violated.

With these definitions, we can define the OCF κ^Z for an AF AF.

Definition 11. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an argumentation framework. Then the Z-attack-Partitioning $(\mathsf{R}_0, \ldots, \mathsf{R}_n)$ with $\mathsf{R}_0 \cup \ldots \cup \mathsf{R}_n \subseteq \mathsf{R}$ is defined as

i	$\kappa^{-1}(i)$
2	$(\{b,c\},X),(\{a,b,c\},X),(\{b,c,d\},X),(\{a,b,c,d\},X)$
1	$(\{a,b\},X),(\{c,d\},X),(\{a,b,d\},X),(\{a,c,d\},X)$
0	$(\emptyset,X),(\{a\},X),(\{b\},X),(\{c\},X),(\{d\},X),(\{a,c\},X),(\{b,d\},X),(\{a,d\},X)$

Table 2. The OCF κ^Z, where for every pair (I,X) $X \subseteq \mathsf{A}$ is any set s.t. $I \cap X = \emptyset$.

- $\mathsf{R}_0 = \{r \in \mathsf{R} \mid \mathsf{R} \text{ tolerates } r\}$
- $(\mathsf{R}_1,\ldots,\mathsf{R}_n)$ is the Z-attack-Partitioning of $\mathsf{R} \setminus \mathsf{R}_0$

For $r \in \mathsf{R}$ define $Z_{\mathsf{R}}(r) = i$ if $r \in \mathsf{R}_i$ and

$$\kappa^Z(S,X) = \max\{Z(r) \mid S \text{ violates } r\} + 1$$

where $X \subseteq \mathsf{A}$ is any set s.t. $S \cap X = \emptyset$.

So all attacks in R_0 are tolerated by the set of attacks of AF, while attacks in R_1 are only tolerated if we remove all attacks from R_0. Now we can state when a set of arguments is more plausible than another one, i. e., if the first set violates either no attacks or only attacks which are in lower levels. In a sense these levels represent the impact of each attack in an AF. Hence, it is more important to satisfy a single highly ranked attack than to satisfy multiple lowly ranked attacks.

Example 5. Consider Example 1 again. The Z-attack-Partitioning of R is $(\mathsf{R}_0,\mathsf{R}_1)$ with

$$\mathsf{R}_0 = \{(a,b),(c,d),(d,c)\}$$
$$\mathsf{R}_1 = \{(b,c)\}$$

Table 2 depicts κ^Z_{AF} for AF from Example 1.

Next, we want to prove, that the function κ^Z satisfies an AF if κ^Z is defined.

Theorem 1. If κ^Z is defined, then κ^Z satisfies AF.

Proof. Let $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ be an AF. In order to show that κ^Z satisfies AF, we need to prove, that κ^Z satisfies both principles of an OCF, i. e. acceptance of attacks and possibly reinstating an argument.

Case: accept attack. Let $(a,b) \in \mathsf{R}$ with $a \neq b$ an attack, it has to hold that $\kappa^Z(\{a\},\{b\}) < \kappa^Z(\{a,b\},\emptyset)$. We know that $\kappa^Z(\{a\},\{b\}) = 0$, because $\{a\}$ can only violate the attack (a,a), which can not exist. Hence, it is enough to show, that $\kappa^Z(\{a,b\},\emptyset) > 0$. Since, (a,b) exists, we know that $\{a,b\}$ violates this attack, and therefore $\kappa^Z(\{a,b\},\emptyset) > 0$.

Case: argument possibly reinstated. Let $a \in \mathsf{A}$ be an argument. Assume $\kappa^Z(S \cup a, a^-) > \kappa^Z(S, a \cup a^-)$ for some $S \subseteq \mathsf{A}$ with $S \cap (a^- \cup a^+) = \emptyset$. This is only possible, if $S \cup \{a\}$ violates an attack $r \in \mathsf{R}$ and S does not violate r. So, there is one argument $b \in S$ s.t. $r = (a,b)$ or $r = (b,a)$ and $a \neq b$. Hence, $b \in a^- \cup a^+$. However, because of $S \cap (a^- \cup a^+) = \emptyset$ we know that, there can not be such an argument $b \in S$ with $b \in a^- \cup a^+$. Therefore $\kappa^Z(S \cup a, a^-) > \kappa^Z(S, a \cup a^-)$ is impossible. □

Besides being undefined for AF with self-attacks, κ^Z is also undefined for AFs without a stable extension. Let $\mathsf{AF}_4 = (\{a,b,c\},\{(a,b),(b,c),(c,a)\})$ be an AF. If we try to tolerate (a,b) by $\{(a,b),(b,c),(c,a)\}$, then we know that, we need to verify (a,b) so $a \in S$. However, this also means that $b,c \notin S$, which entails that attack admissibility of c is violated. Similar we can show, that (b,c) and (c,a) cannot be tolerated either. So, we cannot define a Z-attack-Partitioning for AF_4. Next, we show that in general it holds that if an AF does not have a stable extension, then κ^Z is undefined.

Theorem 2. κ^Z is undefined for AF if $\mathrm{st}(\mathsf{AF}) = \emptyset$.

Proof. Let $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ be an AF. We will show the contraposition, so if κ^Z is defined for AF, then $\mathrm{st}(\mathsf{AF}) \neq \emptyset$. Let κ^Z be defined. So we can find a Z-attack-Partitioning $(\mathsf{R}_0,...,\mathsf{R}_n)$. For every attack r in R_0 we know that there is a set S s.t. r is verified, every attack is satisfied and attack admissibility of every argument $a \in \mathsf{A}$ is satisfied. We show that S is stable. First, S has to be conflict-free, otherwise there is an attack, which is violated. Next we show that $S \cup S^+ = \mathsf{A}$, so we need to show, that every argument not in S is attacked by S. Let $b \notin S$, then because attack admissibility of b is satisfied we know that there is an argument $c \in b^-$ with $c \in S$, hence we have found an attacker of b which is part of S. □

Looking at the levels of a Z-attack-Partitioning in detail, we observe, that if an attack (a,b) is in R_0, then a is credulously admissible accepted in AF.

Theorem 3. For any AF $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ and Z-attack-Partitioning $(\mathsf{R}_0,...,\mathsf{R}_n)$. If $(a,b) \in \mathsf{R}_0$, then a is credulous accepted wrt. admissible semantics.

Proof. Let $\mathsf{AF} = (\mathsf{A},\mathsf{R})$ be an AF, $(\mathsf{R}_0,...,\mathsf{R}_n)$ a Z-attack-Partitioning of R and $(a,b) \in \mathsf{R}_0$, then (a,b) is tolerated by R, meaning that there is an $S \subseteq \mathsf{A}$ s.t. (a,b) is verified, each attack in R is satisfied by S and attack admissibility of each argument $c \in \mathsf{A}$ is satisfied. In order to verify (a,b), we know that $a \in S$ and $b \notin S$. Also it has to hold for all $c \in a^-$ that $c \notin S$. So all attackers of a are *out*. Next, we will show that S is admissible. S is conflict-free, otherwise, one attack would be violated. We know for every $d \in S$ that no attacker of d is in S. In order to not violate attack admissibility, we know for every $e \notin S$ that at least one attacker of e has to be in S, meaning that S attacks every argument not contained in S. Hence, for every attacker b of an argument $a \in S$ we have an argument $d \in S$ s.t. d attacks b. So S is admissible, and therefore a is part of some admissible extension of AF making a credulous accepted w.r.t. admissible semantics. □

5. Extension-ranking Semantics based on System Z

First, we recall the definitions from [2] for *extension-ranking semantics*.

5.1. Extension-Ranking Semantics

Extension-ranking semantics defined in [2] are a generalisation of extension-based semantics. Using them, we can state not only that a set of arguments is jointly accepted or not, but also we can say whether a set E_1 is more plausible than a set E_2.

$$\begin{aligned}\emptyset \cong^{\kappa^Z}_{\mathsf{AF}} \{a\} \cong^{\kappa^Z}_{\mathsf{AF}} \{b\} \cong^{\kappa^Z}_{\mathsf{AF}} \{c\} \cong^{\kappa^Z}_{\mathsf{AF}} \{d\} \cong^{\kappa^Z}_{\mathsf{AF}} \{a,c\} \cong^{\kappa^Z}_{\mathsf{AF}} \{b,d\} \cong^{\kappa^Z}_{\mathsf{AF}} \{a,d\}\\ \preceq^{\kappa^Z}_{\mathsf{AF}} \{a,b\} \cong^{\kappa^Z}_{\mathsf{AF}} \{c,d\} \cong^{\kappa^Z}_{\mathsf{AF}} \{a,b,d\} \cong^{\kappa^Z}_{\mathsf{AF}} \{a,c,d\}\\ \preceq^{\kappa^Z}_{\mathsf{AF}} \{b,c\} \cong^{\kappa^Z}_{\mathsf{AF}} \{a,b,c\} \cong^{\kappa^Z}_{\mathsf{AF}} \{b,c,d\} \cong^{\kappa^Z}_{\mathsf{AF}} \{a,b,c,d\}\end{aligned}$$

Table 3. The ranking for AF based on $\preceq^{\kappa^Z}_{\mathsf{AF}}$.

Definition 12. An *extension ranking* on AF is a preorder[1] over the powerset of arguments 2^{A}. An *extension-ranking semantics* τ is a function that maps each AF to an extension ranking $\preceq^{\tau}_{\mathsf{AF}}$ on AF.

For an extension-ranking semantics τ, an extension ranking $\preceq^{\tau}_{\mathsf{AF}}$, $E, E' \subseteq \mathsf{A}$, and for $E \preceq^{\tau}_{\mathsf{AF}} E'$ we say that E is *at least as plausible as* E' by τ in AF.

Using the OCF κ^Z, we can define an *extension-ranking semantics*. So we can state that a set of arguments E is more plausible than another one E', if the OCF κ^Z returns a lower value for E than for E'.

Definition 13. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an AF and $E, E' \subseteq \mathsf{A}$. Define the *System Z extension-ranking semantics* $\preceq^{\kappa^Z}_{\mathsf{AF}}$ via

$$E \preceq^{\kappa^Z}_{\mathsf{AF}} E' \text{ iff } \kappa^Z(E, \mathsf{A} \setminus E) \leq \kappa^Z(E', \mathsf{A} \setminus E')$$

So E is at least as plausible as E', if E being considered *in* and all arguments not in E being *out*, is more plausible than E' being considered *in* and all arguments not in E' being *out*.

Example 6. Consider again AF from Example 1. Then Table 3 depicts the ranking corresponding to $\preceq^{\kappa^Z}_{\mathsf{AF}}$. We see, that all conflict-free sets are part of the most plausible sets, while sets with conflicts are ranked worse. Also, the number of conflicts is not as important as for the approaches of [2]. In their approaches, it always holds that $\{b,c\}$ is ranked strictly better than $\{b,c,d\}$. While for κ^Z these two sets are ranked equally.

5.2. Study of the System Z Extension-ranking Semantics

Next, we want to evaluate $\preceq^{\kappa^Z}_{\mathsf{AF}}$ based on principles defined by [2].

We begin with σ-*generalisation*, which states that sets of arguments, which satisfies extension semantics σ should also be ranked best by an extension-ranking semantics and every set not satisfying σ should be ranked worse. In Example 6, we can see that $\preceq^{\kappa^Z}_{\mathsf{AF}}$ violates σ-*generalisation* for $\sigma \in \{\mathrm{ad}, \mathrm{co}, \mathrm{pr}, \mathrm{gr}, \mathrm{st}\}$, because the set $\{b,d\}$ is not admissible, however, it is ranked as a most plausible set. Therefore $\preceq^{\kappa^Z}_{\mathsf{AF}}$ cannot satisfy σ-*generalisation* for any admissible based semantics σ.

The next properties (*composition* and *decomposition*) states that unconnected arguments should not influence a ranking.

Theorem 4. $\preceq^{\kappa^Z}_{\mathsf{AF}}$ satisfies *composition*. Where τ satisfies *composition* if for every AF such that $\mathsf{AF} = (\mathsf{A}_1, \mathsf{R}_1) \cup (\mathsf{A}_2, \mathsf{R}_2)$ and $E, E' \subseteq \mathsf{A}_1 \cup \mathsf{A}_2$:

[1] A preorder is a (binary) relation that is *reflexive* ($E \preceq E$ for all E) and *transitive* ($E_1 \preceq E_2$ and $E_2 \preceq E_3$ implies $E_1 \preceq E_3$)

$$\text{if } \left\{ \begin{matrix} E \cap \mathsf{A}_1 \preceq^{\tau}_{\mathsf{AF}_1} E' \cap \mathsf{A}_1 \\ E \cap \mathsf{A}_2 \preceq^{\tau}_{\mathsf{AF}_2} E' \cap \mathsf{A}_2 \end{matrix} \right\} \text{ then } E \preceq^{\tau}_{\mathsf{AF}} E'.$$

Proof. Let $\mathsf{AF} = (\mathsf{A}_1,\mathsf{R}_1) \cup (\mathsf{A}_2,\mathsf{R}_2)$ be an AF and $E,E' \subseteq \mathsf{A}_1 \cup \mathsf{A}_2$. For composition we need to show that, if $\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E')$ and $\kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E')$ then $\kappa^Z(E, \mathsf{A} \setminus E) \leq \kappa^Z(E', \mathsf{A} \setminus E')$. By definition of κ^Z we know that $\kappa^Z(E, \mathsf{A} \setminus E)$ is the maximal value between $\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E)$ and $\kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E)$, because if an attack r_1 is violated by E, then r_1 is also violated by either $E \cap \mathsf{A}_1$ or $E \cap \mathsf{A}_2$. Similar holds for $\kappa^Z(E', \mathsf{A} \setminus E')$. So we have to check four possible cases for $max(\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E), \kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E)) \leq max(\kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E'), \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E'))$.

1. $\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E')$
2. $\kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E')$
3. $\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E')$
4. $\kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E')$

Case 1 and 2 are clear via definition. For case 3 we know that $\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E) \geq \kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E)$ and $\kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E') \leq \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E')$, but we also know that $\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E')$, which proves case 3. Case 4 is similar to case 3. □

For *decomposition*, we see that $\preceq^{\kappa^Z}_{\mathsf{AF}}$ violates it. Recall that τ satisfies *decomposition* if for every AF such that $\mathsf{AF} = (\mathsf{A}_1,\mathsf{R}_1) \cup (\mathsf{A}_2,\mathsf{R}_2)$ and $E,E' \subseteq \mathsf{A}_1 \cup \mathsf{A}_2$:

$$\text{if } E \preceq^{\tau}_{\mathsf{AF}} E' \text{ then } \left\{ \begin{matrix} E \cap \mathsf{A}_1 \preceq^{\tau}_{\mathsf{AF}_1} E' \cap \mathsf{A}_1 \\ E \cap \mathsf{A}_2 \preceq^{\tau}_{\mathsf{AF}_2} E' \cap \mathsf{A}_2 \end{matrix} \right\}.$$

Example 7. Let $\mathsf{AF}_5 = (\{a,b,c,d,e\}, \{(a,b),(b,c),(d,e)\})$ be an AF. This AF can be split into two disjoint AFs $\mathsf{AF}_{5.1} = (\{a,b,c\}, \{(a,b),(b,c)\})$ and $\mathsf{AF}_{5.2} = (\{d,e\}, \{(d,e)\})$. The Z-attack-Partitioning of R_5 is $\mathsf{R}_{5\,0} = \{(a,b),(d,e)\}$ and $\mathsf{R}_{5\,1} = \{(b,c)\}$. Let $E = \{a,b,d,e\}$ and $E' = \{b,c,d\}$, then $\kappa^Z(E, \mathsf{A}_5 \setminus E) = 1$ and $\kappa^Z(E', \mathsf{A}_5 \setminus E') = 2$. However, we have $\kappa^Z(E \cap \mathsf{A}_{5.2}, \mathsf{A}_{5.2} \setminus E) = 1$ and $\kappa^Z(E' \cap \mathsf{A}_{5.2}, \mathsf{A}_{5.2} \setminus E') = 0$. This shows, that decomposition is violated.

Decomposition is violated, because $\preceq^{\kappa^Z}_{\mathsf{AF}}$ focuses on a global view. Violating (b,c) is worse, than violating any other attack. However, $\preceq^{\kappa^Z}_{\mathsf{AF}}$ satisfies a weak version of decomposition, where instead of satisfying $E \cap \mathsf{A}_1 \preceq^{\tau}_{\mathsf{AF}_1} E' \cap \mathsf{A}_1$ for both disjoint AFs, it is enough if κ^Z_{AF} satisfy this for one AF.

Definition 14 (Weak Decomposition)**.** Let τ be an extension-ranking semantics. τ satisfies *weak decomposition* if for every AF such that $\mathsf{AF} = (\mathsf{A}_1,\mathsf{R}_1) \cup (\mathsf{A}_2,\mathsf{R}_2)$ and $E,E' \subseteq \mathsf{A}_1 \cup \mathsf{A}_2$: if $E \preceq^{\tau}_{\mathsf{AF}} E'$ then $E \cap \mathsf{A}_1 \preceq^{\tau}_{\mathsf{AF}_1} E' \cap \mathsf{A}_1$ or $E \cap \mathsf{A}_2 \preceq^{\tau}_{\mathsf{AF}_2} E' \cap \mathsf{A}_2$.

Theorem 5. $\preceq^{\kappa^Z}_{\mathsf{AF}}$ *satisfies weak decomposition.*

Proof. Let $\mathsf{AF} = (\mathsf{A}_1,\mathsf{R}_1) \cup (\mathsf{A}_2,\mathsf{R}_2)$ be an AF and $E,E' \subseteq \mathsf{A}_1 \cup \mathsf{A}_2$. In order to prove weak decomposition we have to show, that if $\kappa^Z(E, \mathsf{A} \setminus E) \leq \kappa^Z(E', \mathsf{A} \setminus E')$ then $\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E')$ or $\kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E')$. By definition we know that $\kappa^Z(E, \mathsf{A} \setminus E) = max(\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E), \kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E))$ and similar for E'. So, we have $max(\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E), \kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E)) \leq max(\kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E'), \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E'))$. Hence, we have four cases to check.

1. $\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E')$
2. $\kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E')$
3. $\kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E')$
4. $\kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E')$

Case 1 and 2 are clear via definition. For case 3 we know that $\kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E) \leq \kappa^Z(E \cap \mathsf{A}_1, \mathsf{A}_1 \setminus E)$ and therefore also $\kappa^Z(E \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E) \leq \kappa^Z(E' \cap \mathsf{A}_2, \mathsf{A}_2 \setminus E')$. Hence, weak decomposition is satisfied. Case 4 can be proven similar to case 3. □

The final properties we want to recall are the reinstatement ones, which state that if an argument is defended and does not add conflicts into a set, then the addition of this argument into a set should not lower the plausibility, respectively should raise the plausibility of the set.

Theorem 6. $\preceq^{\kappa^Z}_{\mathsf{AF}}$ satisfies *weak reinstatement*. Where τ satisfies *weak reinstatement* iff $a \in F_{\mathsf{AF}}(E)$, $a \notin E$ and $a \notin (E^- \cup E^+)$ implies $E \cup \{a\} \preceq^{\tau}_{\mathsf{AF}} E$.

Proof. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an AF and $E \subseteq \mathsf{A}$. Assume $a \notin E$ and $a \notin (E^- \cup E^+)$. We have to show that $\kappa^Z(E \cup \{a\}, \mathsf{A} \setminus E \cup \{a\}) \leq \kappa^Z(E, \mathsf{A} \setminus E)$. We know that $E \cup \{a\}$ violates the same attacks as E, because E and $\{a\}$ are not in a conflict with each other. This means that $\kappa^Z(E \cup \{a\}, \mathsf{A} \setminus E \cup \{a\})$ can not be greater than $\kappa^Z(E, \mathsf{A} \setminus E)$. □

For *strong reinstatement* i. e., adding an argument into an AF, which is defended by a set and does not create more conflicts, should raise the plausibility, we can look at Example 6. We see, that $\{c\}$ is equally ranked to $\{a, c\}$ despite it holds that $a \in F_{\mathsf{AF}}(\{c\})$, $a \notin \{c\}$ and $a \notin (\{c\}^- \cup \{c\}^+)$. So *strong reinstatement* is violated.

Even though a number of properties are violated by $\preceq^{\kappa^Z}_{\mathsf{AF}}$ this does no lower the impact of this semantics, since $\preceq^{\kappa^Z}_{\mathsf{AF}}$ focuses on a global view. The semantics identifies important attacks in the AF and ensures, that these attacks are satisfied. So it is worse to not satisfy a single highly ranked attacked, than not satisfying multiple lower ranked attacks. Another difference of this semantics to the semantics of Skiba et al. [2] is the fact, that the number of conflicts a set contains in not important just the fact, that the set is not conflict-free is significant.

6. Discussion

In this work, we continue the research of investigating the relationship of conditional logics and abstract argumentation, by using concepts for conditional logics to reason in abstract argumentation. In particular, we defined a formalism of OCFs to rank sets of arguments. It turns out that these preorders are in line with current work about extension-ranking semantics and produce a ranking for the powerset of arguments for an argumentation framework.

One use of conditional logics is belief change. Where preorders are used to update beliefs with information inconsistent with them. There are a number of different works investigating belief change involving preorders over extensions of an argumentation framework [12,13,14]. However, all these works tackle a different problem. To summarise, given an AF and an extension semantics σ, the AF will be changed using a preorder to satisfy new information. This paper talks about using OCFs to reason over sets of argument, while not changing AFs. Weydert [9] investigates a different idea to define extension rankings using conditionals, his definitions could be used to define an extension-ranking semantics similar to Section 5. However, his semantics cannot differentiate conflicting sets. All conflicting sets have the same rank of infinity. A full investigation of the properties of the resulting extension-ranking semantics will be done in future work. A noteworthy mention is that System Z and *rational closure* by Lehmann and Magidor [15] use the same construction. So our work also allows us to draw connections between argumentation and non-monotonic inference. Additionally OCFs with natural numbers and an infinity level are really close to possibilistic logics [16].

As there are more possible OCFs satisfying our proposed principles, we can define more extension-rankings semantics, like for example an extension-ranking semantics based on *c-representations* [17].

Acknowledgements. The research reported here was supported by the Deutsche Forschungsgemeinschaft under grant 423456621.

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995.

[2] Skiba K, Rienstra T, Thimm M, Heyninck J, Kern-Isberner G. Ranking Extensions in Abstract Argumentation. In: Proc. of IJCAI'21; 2021. .

[3] Spohn W. Ordinal conditional functions: A dynamic theory of epistemic states. In: Causation in decision, belief change, and statistics. Springer; 1988. .

[4] Goldszmidt M, Pearl J. Qualitative probabilities for default reasoning, belief revision, and causal modeling. Artificial Intelligence. 1996.

[5] Heyninck J. Relations Between Assumption-based Approaches in Nonmonotonic Logics and Formal Argumentation. FLAP. 2019.

[6] Heyninck J, Kern-Isberner G, Thimm M, Skiba K. On the correspondence between abstract dialectical frameworks and nonmonotonic conditional logics. Ann Math Artif Intell. 2021.

[7] Kern-Isberner G, Simari GR. A Default Logical Semantics for Defeasible Argumentation. In: Proc. of FLAIRS'11; 2011. .

[8] Thimm M, Kern-Isberner G. On the Relationship of Defeasible Argumentation and Answer Set Programming. In: Proc. of COMMA'08; 2008. .

[9] Weydert E. A Plausibility Semantics for Abstract Argumentation Frameworks. CoRR. 2014.

[10] Weydert E. On the Plausibility of Abstract Arguments. In: Proc. of ECSQARU'13; 2013. .

[11] Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018.

[12] Booth R, Kaci S, Rienstra T, van der Torre L. A logical theory about dynamics in abstract argumentation. In: Proc. of SUM'13; 2013. .

[13] Coste-Marquis S, Konieczny S, Mailly JG, Marquis P. On the revision of argumentation systems: Minimal change of arguments statuses. In: Proc. of KR'14; 2014. .

[14] Diller M, Haret A, Linsbichler T, Rümmele S, Woltran S. An extension-based approach to belief revision in abstract argumentation. International Journal of Approximate Reasoning. 2018.

[15] Lehmann D, Magidor M. What does a Conditional Knowledge Base Entail? Artif Intell. 1992.

[16] Dubois D, Prade H. Possibilistic Logic - An Overview. In: Siekmann JH, editor. Computational Logic. Handbook of the History of Logic. Elsevier; 2014. .

[17] Kern-Isberner G. Conditionals in nonmonotonic reasoning and belief revision: considering conditionals as agents. Springer; 2001.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220163

Strongly Accepting Subframeworks: Connecting Abstract and Structured Argumentation[1]

Markus ULBRICHT [a] and Johannes P. WALLNER [b]
[a] *Department of Computer Science, Leipzig University, Germany*
[b] *Institute of Software Technology, Graz University of Technology, Austria*
ORCiD ID: Markus Ulbricht https://orcid.org/0000-0002-0773-7510, Johannes P. Wallner https://orcid.org/0000-0002-3051-1966

Abstract. Computational argumentation is primed to strengthen the current hot research field of Explainable Artificial Intelligence (XAI), e.g., by dialectical approaches. In this paper, we extend and discuss a recently proposed approach of so-called strong acceptance on abstract argumentation that aims to support explaining argumentative acceptance. Our goal is to push these results into the realm of structured argumentation. In this setting, a knowledge base induces an abstract argumentation framework (AF) via instantiation. We investigate how and under which conditions it is possible to transfer results regarding strong acceptance between the given knowledge base and the induced AF. To this end we consider generic functions formalizing the interaction of the AF and the knowledge base. This approach helps us to infer rather general results making basic assumptions rather than dealing with the technical details of several structured argumentation formalisms. Along the way, we apply our techniques to the concrete approach of assumption-based argumentation (ABA) which constitutes one of the primal structured argumentation formalisms.

Keywords. Structured argumentation, Assumption-based argumentation, Strong acceptance

1. Introduction

Computational argumentation is a thriving research area within the broader field of knowledge representation and reasoning and the landscape of AI research [1]. With application avenues in, e.g., legal and medical research [2], a key contribution of computational argumentation are ways of specifying argument structures and argumentative acceptance forming the basis for, e.g., automated argumentative reasoning. Central to these formalizations are prescribed workflows: from a knowledge base argument structures are instantiated, together with inter-argument relations, upon which argumentation semantics define criteria of acceptance [3,4,5,6]. Importantly it was shown that after instantia-

[1] This work was supported by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig

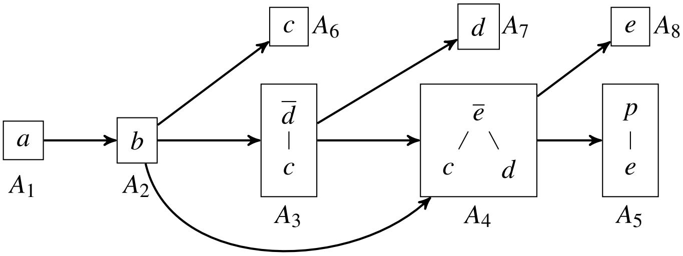


Figure 1. Arguments $\{A_3, A_5\}$ are strongly accepting p on the AF side. On the KB side, $\{(\overline{d} \leftarrow c, e), (p \leftarrow e), e\}$ is strongly accepting p.

tion, an abstract view of arguments and their relations suffices for deriving acceptance of arguments for several use cases.

Computational argumentation is actively contributing to the area of explainable artificial intelligence (XAI) [7], e.g., by providing dialectical grounds of acceptance or rejection of arguments and claims. Several approaches that support explainability arising from argumentation have been proposed and studied. We focus here on ways of supporting explanations by investigating what arguments, or parts of the knowledge base, are sufficient to show acceptance of a target conclusion or argument.

In monotonic formalisms, a common way of looking at parts that entail acceptance is to look at minimal parts (of a knowledge base) that entail the result. In non-monotonic approaches, such as virtually all approaches to argumentation, an adapted notion was presented in order to account for the fact that parts of a knowledge base might entail a certain claim, but as a whole it does not. Formally, given some set B, we can only be sure that a subset B' suffices to entail a certain piece of information, whenever this is the case for each B'' with $B' \subseteq B'' \subseteq B$.

Recently, these approaches were extended to the field of abstract argumentation [8,9,10]. However, as observed from various other aspects [11,12], connecting results from abstract to the non-abstract view is not always immediate. Consider a simple example in the structured argumentation formalism called assumption-based argumentation (ABA) [4], where, briefly put, arguments are derivations starting off from assumptions via a given set of derivation rules. A (possibly asymmetric) contrary relation decides conflicts between arguments. Altogether arguments and directed conflicts (attacks) are referred to as argumentation frameworks (AFs) [13].

Example 1. *Consider an ABA framework consisting of five assumptions $\{a, b, c, d, e\}$ and three rules: $(p \leftarrow e)$, $(\overline{e} \leftarrow c, d)$, and $(\overline{d} \leftarrow c)$. That is, from assumption e we can derive p, from assumptions c and d we can derive $\overline{e}$, and from assumption c we can derive $\overline{d}$. As for (asymmetric) contraries, let $\overline{d}$ be the contrary of d, $\overline{e}$ be the contrary of e, b be the contrary of c, and finally a the contrary of b. These ingredients lead to eight different arguments that can be instantiated from this ABA framework. All of these directly correspond to derivations based on the assumptions via the rules. Figure 1 shows all eight arguments and their directed conflicts. For instance, argument A_3 attacks A_4 because the former concludes $\overline{d}$, the contrary of d, which is an assumption in A_4.*

Let us consider ways of argumentative acceptance of atom p and the prominent approach of admissibility and credulous acceptance. Reasoning on ABA frameworks can be defined via arguments: finding a set of non-conflicting arguments that defends itself *against counterarguments and concludes p. It holds that $\{A_1,A_3,A_5\}$ constitutes an admissible set that concludes p, e.g., the attack from A_2 onto A_3 is countered by A_1.*

Following recent work [8,9,10], one can look at so-called strongly accepting subframeworks that show parts that are sufficient for acceptance. This notion is defined on the level of arguments. For instance here $\{A_3,A_5\}$ is a strongly accepting subframework for p. Let us look at the subframework consisting only of these two arguments: there are no conflicts and p can be concluded. "Re-adding" A_4 leads to a conflict with A_5 (the argument we need to defend), but it holds that A_3 defends A_5 against A_4. Adding A_2 leads to the case that A_2 defends A_5 against A_4 (again making up an admissible set in the subframework concluding p). Adding further arguments again leads to the overall picture: if one commits to A_3 and A_5 we can safely find admissible sets concluding p in the subframeworks in-between the one with only these two arguments and all arguments.

On the other hand, when looking at the structured ingredients needed to conclude p, we find that assumption e and two rules $(\overline{d} \leftarrow c)$ and $(p \leftarrow e)$ are sufficient: with these components we can only instantiate argument A_5 (for all others some components are missing), and adding any further rules or assumptions leads to cases where we still find an admissible set concluding p. For instance, re-adding components to instantiate A_4 requires assumption c, with which, together with $(\overline{d} \leftarrow c)$, leads to the case that A_3 can be instantiated, again leading to the case that A_3 defends A_5 against A_4. However, here we see a mismatch: considering all arguments that can be instantiated from the strongly accepting assumption e and the two rules leads to the set $\{A_5\}$ of arguments. This set does not constitute a strongly accepting subframework when looking only at the argument-level, since we can add A_4 (and no other argument), which defeats A_5.

We address strongly accepting subframeworks in terms of parts of knowledge bases and AFs, and provide the following main contributions.

- We introduce strongly accepting sub-bases both for a general notion of structured knowledge bases, and for the concrete example of ABA.
- We show that strongly accepting subframeworks of a corresponding AF induce a strongly accepting sub-base on the side of the knowledge base, for a generic structured argumentation approach.
- At the same time, as exemplified above, we show that the converse does not hold in general, and point out that by considering "closed" AFs (containing all arguments and attacks from base), and an adaption of strong acceptance on AFs addresses this issue.
- We show that under mild assumptions, a strongly accepting subframework can be bound polynomially on the argument-level in terms of the original knowledge base. This indicates that even though AFs corresponding to a knowledge base might be large (exponentially-sized), a "witness" for acceptability can be bound polynomially in size.

2. Preliminaries

Abstract Argumentation. We fix a non-finite background set $\mathscr{U}$. An argumentation framework (AF) [13] is a directed graph $F = (A,R)$ where $A \subseteq \mathscr{U}$ represents a set of arguments and $R \subseteq A \times A$ models *attacks* between them. Let $\mathbb{F}$ be the set of all AFs over

$\mathscr{U}$. For two arguments $x,y \in A$, if $(x,y) \in R$ we say that x *attacks* y as well as x *attacks* (the set) E given that $y \in E \subseteq A$. A set $E \subseteq A$ attacks $a \in A$ if $\exists b \in E$ with $(b,a) \in R$. We let $E_F^+ = \{x \in A \mid E \text{ attacks } x\}$ for a set $E \subseteq A$. For two AFs $F = (A,R)$ and $G = (B,S)$ we define the $\subseteq$ relation component-wise, i.e. $F \subseteq G$ if $A \subseteq B$ and $R \subseteq S$.

A set $E \subseteq A$ is *conflict-free* in F iff for no $x,y \in E$, $(x,y) \in R$. We say E *defends* an argument x if E attacks each attacker of x. A conflict-free set E is *admissible* in F ($E \in ad(F)$) iff it defends all its elements. Given an AF $F = (A,R)$ a *semantics* σ returns a set of subsets of A. These subsets are called σ-*extensions*. In this paper we consider so-called *complete*, *grounded*, *preferred*, and *stable* semantics (abbr. *co*, *gr*, *pr*, *stb*).

Definition 1. *Let $F = (A,R)$ be an AF and $E \in ad(F)$.*

- *$E \in co(F)$ iff E contains all arguments it defends;*
- *$E \in gr(F)$ iff E is $\subseteq$-minimal in $co(F)$;*
- *$E \in pr(F)$ iff E is $\subseteq$-maximal in $co(F)$;*
- *$E \in stb(F)$ iff $E^+ = A \setminus E$.*

Assumption-based Argumentation. We assume a deductive system $(\mathscr{L},\mathscr{R})$, where $\mathscr{L}$ is a formal language, i.e. a set of sentences, and $\mathscr{R}$ is a set of inference rules over $\mathscr{L}$. A rule $r \in \mathscr{R}$ has the form $a_0 \leftarrow a_1,\dots,a_n$ with $a_i \in \mathscr{L}$. We denote the head of r by $head(r) = a_0$ and the (possibly empty) body of r with $body(r) = \{a_1,\dots,a_n\}$.

Definition 2. *An ABA framework is a tuple $(\mathscr{L},\mathscr{R},\mathscr{A},\overline{})$, where $(\mathscr{L},\mathscr{R})$ is a deductive system, $\mathscr{A} \subseteq \mathscr{L}$ a non-empty set of assumptions, and a contrary function $\overline{} : \mathscr{A} \to \mathscr{L}$.*

Assumption 1. *In this work, we focus on ABA frameworks which are* flat*, i.e., for each rule $r \in \mathscr{R}$, $head(r) \notin \mathscr{A}$ (no assumption can be derived), and* finite*, i.e., $\mathscr{L}$, $\mathscr{R}$, $\mathscr{A}$ are finite; moreover, each rule is stated explicitly (given as input).*

Given an ABA framework $D = (\mathscr{L},\mathscr{R},\mathscr{A},\overline{})$, a tree-based argument is a finite labeled rooted tree t, also denoted by $A \vdash_{R'} p$ with $A \subseteq \mathscr{A}$, $R' \subseteq \mathscr{R}$, and $p \in \mathscr{L}$, s.t. the root is labeled with p. Moreover, each leaf is labeled by an assumption $a \in \mathscr{A}$ or a dedicated symbol $\top \notin \mathscr{L}$. The set of all labels of leaves is A, and each internal node is labeled with the head of a rule $r \in R'$ s.t. the set of labels of children of this node is equal to the body of r or $\top$ if the body is empty. For each $r \in R'$ there must be such a corresponding internal node. We write $A \vdash p$ if there is some $R' \subseteq \mathscr{R}$ s.t. $A \vdash_{R'} p$. An ABA framework induces an AF as follows.

Definition 3. *The associated AF $F_D = (A,R)$ of an ABA $D = (\mathscr{L},\mathscr{R},\mathscr{A},\overline{})$ is given by $A = \{S \vdash p \mid \exists R' \subseteq \mathscr{R} : S \vdash_{R'} p\}$ and attack relation $(S \vdash p, S' \vdash p') \in R$ iff $p \in \overline{S'}$.*

Semantics of ABA frameworks can then be direct taken as σ-extensions of the associated AFs.

Notion of Credulous Acceptance As for acceptance, we consider credulous acceptance. Given an ABA framework and a semantics σ, an atom $p \in \mathscr{L}$ is (credulously) accepted under σ if there is a σ-extension on the associated AF with an argument $A \vdash p$ in the extension. When looking only at the argument-level, an atom is, likewise, (credulously) accepted if there is a σ-extension concluding the atom. Formally, we let $crd_\sigma(F) = \bigcup_{E \in \sigma(F)} conc(E)$ for a semantics σ and AF F, and with $conc(E) = \{p \mid (A \vdash p) \in E\}$ (i.e., collecting all conclusions of arguments). For an ABA framework D and its associated AF F_D, we let $crd_\sigma(D) = crd_\sigma(F_D)$, i.e. the semantics of ABA frameworks is defined via AF instantiation.

3. A General View on Structured Argumentation

Before delving into defining the notion of strongly accepting subframeworks on ABA, we first define a general view on structured argumentation in order to broaden our scope. We consider a general approach to structured argumentation formalisms in line with ABA. For our purposes, three ingredients are important:

- a definition of structured knowledge bases,
- a translation to AFs,
- a function extracting components of the knowledge base from an instantiated AF.

Formally, a knowledge base is an (abstract) structure B which we simply define as a set for ease of presentation. That is, B can be seen as a set composed of ingredients making up the knowledge base. By definition of sets, we arrive at sub-bases by referring to the $\subseteq$ relation. For instance, $\emptyset$ is the empty knowledge base. Given a knowledge base, we need a function that instantiates the knowledge base as an AF. We denote this function as af. We also consider a function extracting a knowledge base (back) from an AF, denoted by kb. These mappings af and kb were used in a similar fashion before [12], but not presented in the same depth and not connected to strong acceptance.

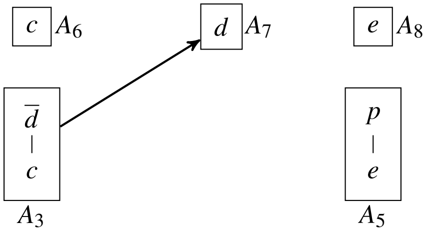
Definition 4. *A knowledge base is a set B. Define the mapping $af : 2^B \to \mathbb{F}$. We define $\mathbb{F}_B = \{F \mid F \subseteq af(B)\}$ as the set of sub-frameworks of the AF instantiated from B.*

In order to apply our general definitions to ABA frameworks D, we identify $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, ^{-})$ with the set $\mathscr{R} \cup \mathscr{A}$ of rules and assumptions. Recall that each ABA framework D induces an associated AF F_D as defined above. We can thus apply our general proposal to ABA in a naturally way by letting $af_{ABA}(D) := F_D$. Given a fixed ABA framework D, the mapping $af_{ABA} : 2^D \to \mathbb{F}$ formalizes all conceivable possibilities to instantiate D using only a subset of the rules and assumptions. Thereby, we will sometimes work with a technically different ABA framework containing fewer assumptions. With a little notational abuse, we always assume that the contrary function $^{-}$ is suitably restricted to the considered set of assumptions.

Example 2. *In this example, we extend the ABA framework from our motivating example $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, ^{-})$, i.e. we have $\mathscr{A} = \{a, b, c, d, e\}$, $\overline{c} = b$ as well as $\overline{a} = b$ and rules*

$$p \leftarrow e. \qquad \overline{e} \leftarrow c, d. \qquad \overline{d} \leftarrow c. \qquad p \leftarrow r.$$

where "$p \leftarrow r.$" is a novel rule which is not applicable. For brevity, we use $\overline{d}$ and $\overline{e}$ to mean fresh symbols without explicating them. The mapping af_{ABA} applied to D returns the argumentation framework depicted in Figure 1. If we restrict $\mathscr{R}$ to the set of rules $\{(p \leftarrow e.), (\overline{d} \leftarrow c.)\}$ and $\mathscr{A}$ to $\{c, d, e\}$, then the corresponding AF is the sub-graph consisting of A_3, A_5, A_6, A_7, and A_8:



For the usual instantiation procedures known from the literature, not every AF $F \in \mathbb{F}_B$ corresponds directly to some subset of the knowledge base. Consider e.g. our running example. There is no subset of $\mathscr{R} \cup \mathscr{A}$ resulting in the AF containing the argument A_5 only since constructing A_5 requires assumption e which would induce A_8 as well.

Moving from a knowledge base to an induced AF is a standard procedure in structured argumentation formalisms. For our investigation, we need to connect AFs and structured bases in both directions. Therefore, we require a formal tool to extract (parts of) the given knowledge base from (parts of) the instantiated AF.

Definition 5. *Let B be a knowledge base. Define the mapping $kb : \mathbb{F}_B \to 2^B$.*

That is, for a knowledge base B, each sub-framework $F \in \mathbb{F}_B$ is mapped to some subset $kb(F) = B' \subseteq B$ of the original knowledge base. Intuitively, one may e.g. think of those parts of the knowledge base which are necessary to construct the arguments occurring in F.

As we already mentioned, not every AF $F \in \mathbb{F}_B$ is induced by some $B' \subseteq B$. Therefore, it holds that *af* and *kb* are in general not inverse to each other. However, for some of our technical results we require the two mappings to correspond to each other in a certain sense. As already mentioned, the intuitive idea is that when considering a sub-framework $F \in \mathbb{F}$, in $kb(F)$ we collect all components of the knowledge base which are necessary to construct F. If we then apply *af* again obtaining $af(kb(F))$, we expect F (and potentially further arguments) to be constructible again, for otherwise our selected components in $kb(F)$ would not suffice for our intended purpose.

Definition 6. *Let B be a knowledge base. We call af and kb* well-behaved *if for all $F \in \mathbb{F}_B$ it holds that $F \subseteq af(kb(F))$.*

An illustration of this notion is depicted in Figure 2. Inspecting the relationship $F \subseteq af(kb(F))$ reveals that if this inclusion is proper, i.e., $F \subsetneq af(kb(F))$, we find a kind of (apparent) "closure" operator, i.e. the composition of *af* and *kb*. Intuitively, if $af(kb(F))$ contains more arguments than F itself, then further arguments can be constructed without the necessity to make use of additional components of the knowledge base.

Definition 7. *Let B be a knowledge base. We call an AF F* closed *if there is some $B' \subseteq B$ s.t. $F = af(B')$. We call af and kb* strictly well-behaved *if they are well behaved and in addition $F = af(kb(F))$ holds for all closed AFs F.*

Let us now demonstrate how we can define a natural mapping *kb* in the context of ABA. The attentive reader may realize that for *kb* to be reasonably defined we need to be able to extract the knowledge base from the given instantiated AF. This is possible

$$\begin{tikzcd}
B \arrow[r, "af"] \arrow[d, phantom, "\rotatebox{90}{$\subseteq$}"] & af(B) \arrow[d, phantom, "\rotatebox{90}{$\subseteq$}"] \\
kb(F) \arrow[r, "af"] & af(kb(F)) \arrow[d, phantom, "\rotatebox{90}{$\subseteq$}"] \\
& F \arrow[ul, "kb"]
\end{tikzcd}$$

Figure 2. For a given knowledge base B, function $af(B)$ results in the associated AF of B. For an AF F that is a sub AF of $af(B)$, inspecting its components ($kb(F)$) leads to a sub part of B (potentially proper). Applying af on the sub part may lead to a potential super framework of F (namely $af(kb(F))$). The composite function $af(kb(\cdot))$ can be interpreted as a closure operation.

for ABA as long as we can be sure the knowledge base D does not contain any hidden information which is not reflected in F_D.

The following notion suffices to ensure that all information included in the ABA framework are made explicit in the selection of all arguments. It simply states that for each atom $p \in \mathscr{L}$, there is at least one tree-based argument inferring it.

Definition 8. *We call an ABA framework* $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, ^-)$ trim *if* $\mathscr{L} = Th_D(\mathscr{A})$.

Interestingly, assuming that D is trim already suffices to rebuild the whole ABA framework by inspecting the constructed arguments. From a technical point of view, we want to emphasize however that in our definition of an argument, the whole tree is stored.

Proposition 1. *Let* $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, ^-)$ *be a trim ABA framework and let* $F_D = (A, R)$ *be the associated AF. Then*

- $\mathscr{L}$ *is the union of all labels of roots occurring in A,*
- $\mathscr{A}$ *is the union of all labels of leaves except* $\top$ *occurring in A,*
- $a_0 \leftarrow a_1, \ldots, a_n \in \mathscr{R}$ *iff there is some* $t \in A$ *s.t.* a_0 *is a label of a node in t and its children are labeled* $a_1, \ldots, a_n$.

Proof. ($\mathscr{L}$) We have $\bigcup_{t \in A} root(t) = Th_D(\mathscr{A}) = \mathscr{L}$ where the first "=" holds by definition and the second since D is trim.

($\mathscr{A}$) We show $\bigcup_{t \in A} leaves(t) \setminus \{\top\} = \mathscr{A}$. The inclusion $\subseteq$ is clear. For the other direction note that each assumption $a \in \mathscr{A}$ induces a tree-based argument $\{a\} \vdash a$.

($\mathscr{R}$) The ($\Leftarrow$)-direction follow from the way argument trees are constructed. Regarding ($\Rightarrow$) let $r = a_0 \leftarrow a_1, \ldots, a_n \in \mathscr{R}$. Since D is trim, there are tree-based arguments $t_1, \ldots, t_n \in A$ with $root(t_i) = a_i$ for $1 \leq i \leq n$. Therefore, there is a tree-based argument t stemming from $t_1, \ldots, t_n$ and the rule r, i.e. $t \in A$ s.t. $root(t) = a_0$ and the children of the root are labeled with $a_1, \ldots, a_n$. □

Inspired by Proposition 1, for a given AF F we let $kb_{ABA}(F)$ be the set $\mathscr{R} \cup \mathscr{A}$ of rules and assumptions as described in the proposition, i.e. $a_0 \leftarrow a_1, \ldots, a_n \in kb_{ABA}(F)$ iff there is some $t \in A$ s.t. a_0 is a label of a node in t and its children are labeled $a_1, \ldots, a_n$; $a \in kb_{ABA}(F)$ iff $a \neq \top$ and there is some leave labelled a. With a little notational abuse we denote the induced ABA framework with $(\mathscr{L}, kb_{ABA}(F_D), ^-)$.

Interestingly, D does not need to be trim in order to find all the necessary components of the knowledge base, since we can simply ignore rules which are not applicable. In the following we establish that af_{ABA} and kb_{ABA} are strictly well-behaved, even without restricting our attention to trim ABA frameworks.

Proposition 2. *Let $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, \overline{})$ be an ABA framework and let D' be induced by $kb_{ABA}(F)$, i.e. $D' = (\mathscr{L}, kb_{ABA}(F_D), \overline{})$. Then $F_D = F_{D'}$.*

Proof. Set $F_D = (A,R)$ and $F_{D'} = (A',R')$. We have $kb_{ABA}(D) \subseteq \mathscr{R} \cup \mathscr{A}$ (part of the proof of Proposition 1 which does not require D to be trim). Moreover, $kb_{ABA}(D) \cap \mathscr{A} = \mathscr{A}$ is clear since each assumption induces some argument. By definition of the instantiation procedure, $F_{D'} \subseteq F_D$. Since attacks are uniquely determined by the tree-based arguments, it suffices to show that $A \subseteq A'$.

Suppose the contrary, i.e. take $t \in A \setminus A'$. Without loss of generality, assume that each proper sub-argument in t occurs in A'. Since D and D' share the same assumptions, the root label of t is no assumption, say a_0. Let $a_1, \ldots, a_n$ be the label of the children. By definition $a_0 \leftarrow a_1, \ldots, a_n \in kb_{ABA}(D)$ and hence $t \in A'$; contradiction. □

Example 3. *For D with the AF F_D from Example 2, $kb_{ABA}(F_D)$ consists of the rules*

$$p \leftarrow e. \qquad \overline{e} \leftarrow c,d. \qquad \overline{d} \leftarrow c.$$

and assumptions $\mathscr{A} = \{a,b,c,d,e\}$ which can be extracted from the tree-based instantiated arguments occurring in the AF (see Figure 1); the dummy rule "$p \leftarrow r.$" is lost. Nonetheless, $af_{ABA}(kb_{ABA}(F_D)) = F_D$.

Consider an AF F' consisting only of argument A_5 (with assumption e and rule $p \leftarrow e$). This AF is not closed. It holds that $F' \subsetneq af_{ABA}(kb_{ABA}(F')) = F''$ with F'' having the two arguments A_5 and A_8. Confirming our intuition, F'' is now closed.

Corollary 1. *The mappings af_{ABA} and kb_{ABA} are strictly well-behaved.*

We now have settled the syntax of our approach. Regarding the semantics, let us consider a generic mapping *acc* that returns for both knowledge bases and AFs a set of "accepted" atoms.

Definition 9. *The mappings acc and af are called* compatible *if it holds that $a \in acc(B)$ iff $a \in acc(af(B))$ for each knowledge base B.*

Intuitively, compatibility simply states that when instantiating a knowledge base, the corresponding semantics are preserved. It is well-known that this is the case for ABA.

Proposition 3. *For the ABA and AF semantics we consider, crd_σ and af_{ABA} are compatible (for fixed $\mathscr{L}$, and $\overline{}$).*

We remark that our general view and the condition of *af* and *kb* being well-behaved ($F \subseteq af(kb(F))$) have a certain relation to Galois connections. Highlighting only the essential, if $kb(F) \subseteq B$ implies $F \subseteq af(B)$ (*), one can show that $F \subseteq af(kb(F))$ follows. Condition (*) can be seen as "one direction" of the requirement for Galois connections. Intuitively, condition (*) appears plausible for structured argumentation: if B contains all ingredients to instantiate F, then $af(B)$ should contain all of F, as well.

Proposition 4. *If $kb(F) \subseteq B$ implies $F \subseteq af(B)$, it holds that $F \subseteq af(kb(F))$.*

Proof. By definition, $kb(F) \subseteq kb(F)$ holds and $F \subseteq af(kb(F))$ holds by assumption. □

4. Strongly Accepting Subframeworks

Next we discuss our notion of strong acceptance, utilizing earlier works [8,10].

4.1. Basic Definitions

The idea behind a strongly accepting sub-framework is that one aims to find a sub-framework $B' \subseteq B$ accepting a certain atom (argument), but accounts for non-monotonicity by requiring that this property survives moving to supersets of B' within B. We first define this idea on knowledge bases.

Definition 10. *Let B a knowledge base. A set $B' \subseteq B$* strongly accepts *a if $a \in acc(B'')$ for all B'' such that $B' \subseteq B'' \subseteq B$.*

As mentioned earlier, strong acceptance can be defined on AFs [8,10].

Definition 11. *Let B a knowledge base and $F = af(B)$. A sub-AF $F' \in \mathbb{F}_B$* strongly accepts *a if $a \in acc(F'')$ for all F'' such that $F' \subseteq F'' \subseteq F$.*

Example 4. *Consider our running Example 2. As we already discussed in the introduction, the sub-AF consisting of $\{A_3, A_5\}$ strongly accepts A_5 (and hence the atom p). From the point of view of the knowledge base, we require the assumptions $\mathscr{A}' = \{c, e\}$ and rules $\mathscr{R}' = \{(\bar{d} \leftarrow c, e.), (p \leftarrow e.)\}$ to construct these arguments. The reader may verify that indeed, $\mathscr{A}' \cup \mathscr{R}'$ strongly accepts p (note that this is a superset to the one discussed in the introduction; we come back to a smaller one below).*

The observation we made in the previous example is no coincidence: given a strongly accepting sub-graph of $af_{ABA}(B)$, under mild conditions we can find a strongly accepting subset of B by applying the mapping kb_{ABA}. The following proposition formalizes this result.

Proposition 5. *Let B be a knowledge base, and $F' \in \mathbb{F}_B$. Suppose af is $\subseteq$-monotone, and af, kb, and acc are well-behaved and compatible. If F' strongly accepts a, then $kb(F')$ strongly accepts a.*

Proof. Assume that F' strongly accepts a and $kb(F') = B'$ does not. Then there is a B'' with $B' \subseteq B'' \subseteq B$ with $a \notin acc(B'')$. It holds that $af(B') \subseteq af(B'')$ (by monotonicity of af) and $F' \subseteq af(kb(F'))$ (by af and kb being well-behaved). Then $F' \subseteq af(kb(F')) = af(B') \subseteq af(B'')$. Moreover, since $af(B)$ is $\subseteq$-maximal, it follows that $F' \subseteq af(B'') \subseteq af(B)$. By assumption that F' is strongly accepting a, it follows that $a \in acc(af(B''))$, which implies $a \in acc(B'')$, a contradiction. □

The opposite direction does not hold in general, also not for concrete instantiations (e.g., ABA and ASPIC$^+$ [3]) as discussed partially before and more concretely in the following counter-example.

Example 5. *From our running example ABA D consider the rules $\mathscr{R}'$:*

$$\bar{d} \leftarrow c. \qquad\qquad p \leftarrow e.$$

and $\mathscr{A}' = \{e\}$. The set $D' = \mathscr{A}' \cup \mathscr{R}'$ strongly accepts p in D which can be seen as follows. In order to prevent p from being acceptable, the argument A_4 is required. However, for this we would need to add assumptions c and d as well which in turn allows to construct A_3. We already know however that presence of A_3 and A_5 suffice to strongly accept p. On the other hand, the AF $af_{ABA}(D')$ consists of the argument A_5 from Figure 1 only which does not suffice to strongly accept p: we can simply add A_4.

However, if *af* is a bijection and *kb* its inverse, then the converse is also true.

Proposition 6. *Let B a knowledge base, and $F' \in \mathbb{F}_B$ as well as $B' \subseteq B$. Suppose af and kb are $\subseteq$-monotone, and af, kb, and acc are compatible s.t. $af : 2^B \to \mathbb{F}_B$ is a bijection and $kb = af^{-1}$. If F' strongly accepts a, then $kb(F')$ strongly accepts a. If B' strongly accepts a, then $af(B')$ strongly accepts a.*

To summarize, the notion of strong acceptance can be naturally defined for both the AFs as well as the underlying knowledge base. Since not every AF $F \in \mathbb{F}_B$ is induced by some subset of the knowledge base B, it is in general not true that strong acceptance is preserved when applying *af* resp. *kb*. However, as formalized by Proposition 5, under mild conditions it can be translated back from the AF to the knowledge base. As our results regarding ABA demonstrate, the preconditions for Proposition 5 hold for ABA.

If one restricts strong acceptance on the AF side to only closed AFs in $\mathbb{F}_B$, in addition to having *af* being a bijective with *kb* its inverse, we can apply Proposition 6 to conclude that strong acceptance transfers in both ways.

4.2. Strong Acceptance and Size of Subframeworks

In the general case, an associated AF may not be small w.r.t. the structured knowledge base it was instantiated from. For instance, an AF associated to an ABA framework may be exponential in size [14] (the example given there applies to ABA as well). Nevertheless, as we show, in many cases one can bound strongly accepting subframeworks *both* on the knowledge base and argument level.

To formalize this idea, we will make use of claim-augmented AFs (CAFs) [15]. A CAF is a triple $\mathscr{F} = (A, R, cl)$ where $F = (A, R)$ is an AF and $cl : A \to \mathscr{C}$ assigns a claim $c \in \mathscr{C}$ to each argument in A; $\mathscr{C}$ is a countably-infinite set. We let $cl(E) = \{cl(e) \mid e \in E\}$ for a set E of arguments. In the literature, semantics for CAFs have been introduced and formally investigated, but we are only interested in the claims as additional information. We thus let $\sigma(\mathscr{F}) = \sigma(F)$ for each semantics σ considered in this paper. We assume that our mappings *af*, *kb*, and *acc* naturally extend to CAFs.

Example 6. *Our running example ABA frameworks yields a CAF $\mathscr{F} = (A, R, cl)$ where the underlying $F = (A, R)$ corresponds to the AF from Figure 1. Claims of arguments correspond to their conclusions, i.e. we let $cl(A_3) = \bar{d}$ and analogously for the other A_i.*

When constructing arguments corresponding to an ABA framework *D*, out-going attacks are naturally characterized by the conclusions of an argument. Viewing the AF as a CAF by defining *cl* as shown above, this procedure yields a well-formed CAF. In the following, we assume that the CAFs we work with possess this feature.

Assumption 2. *For a given KB B, $af(B) = \mathscr{F} = (A, R, cl)$ is well-formed in the sense that $cl(a) = cl(b)$ implies $a^+ = b^+$ for each argument $a, b \in A$.*

We remark that this assumption holds for many structured argumentation approaches, such as ABA, but not all: in case of preferential approaches (e.g., ASPIC$^+$ [3]) one can find counterexamples. We make use of one additional technical assumption that holds for several structured argumentation formalisms.

Assumption 3. *For a given KB B, it holds that the set of claims in $af(B)$ is bounded polynomially by $|B|$.*

This assumption does not hold, e.g., if there is an underlying logic or deductive system of a knowledge base which gives rise to claims not present in original knowledge base (e.g., if a leads to $a \vee b$ leads to $a \vee b \vee c$, ...). Observe however that for our running example of non-preferential ABA instantiations, we have $|cl(A)| \leq |\mathscr{L}|$ and for each framework D, F_D is well-formed.

In the following, we will show that any strongly accepting sub-framework can be reduced to at most $|C|$ arguments, where C is the set of claims occurring in F. The crucial observation is formalized in the following theorem. It states an admissible extension $E \in ad(F)$ with more than $|C|$ claims can be reduced in size. The proof proceeds by removing arguments which do not contribute any novel claim.

Theorem 1. *Let $\mathscr{F} = (A,R,cl)$ be well-formed, $a \in A$, and the set of claims $C = cl(A)$ of $\mathscr{F}$ finite. If a set of arguments E is admissible in F and contains a, then there exists an admissible set E' with $a \in E'$ and $|E'| \leq |C|$.*

Proof. Let E be admissible in F and $a \in E$. If $|E| \not\leq |C|$ then there is a claim $\alpha \in C$ such that more than one argument in E has claim α. Pick any $a \in E$ with $claim(a) = \alpha$. We claim that $E' = E \setminus \{x \in E \mid claim(x) = \alpha, x \neq a\}$ is admissible in F. It holds that E' is conflict-free (since conflict-freeness holds for any subset of a conflict-free set). Suppose there is a $b \in E'$ such that there is a $c \in A$ with $(c,b) \in R$ and there is no $d \in E'$ with $(d,c) \in R$. Since $E' \subseteq E$ it holds that $b \in E$. Since E is admissible there is a $d' \in E$ such that $(d,c) \in R$. Since $d' \in E \setminus E'$, by construction of E', it holds that $claim(d') = \alpha = claim(a)$. By well-formedness, it holds that $(a,c) \in R$, contradicting the assumption that b is not defended by E'. We conclude that E' is admissible in F. Define $C' \subseteq C$ as the set of claims of arguments in E. Finally, pick for each claim $\alpha \in C'$ an argument $x_\alpha \in E$ with the exception that a_α is chosen for the claim of a, and set $E^* = \{x_\alpha \mid x_\alpha \in E, \alpha \in C'\}$. By the statements above it holds that E^* is admissible and contains a. By construction, there is exactly one argument per claim in C'. □

For the other semantics $\sigma \in \{co, gr, pr, stb\}$ we might move to a superset in order to fulfill the semantic-specific requirements, but we can be sure that an admissible set of small size can be extended in a suitable way.

Corollary 2. *Let $\mathscr{F} = (A,R,cl)$ be well-formed, $a \in A$, and the set of claims $C = cl(A)$ of F finite. Let $\sigma \in \{co, gr, pr, stb\}$. If $E \in \sigma(F)$, then there is some $E_0 \in ad(F)$ with $|E_0| \leq |C|$ and $E_0 \subseteq E$.*

Since extensions induce strongly accepting subsets as formalized in [8] we can now infer that the size of such subsets can be trimmed down to size of at most $|C|$ arguments.

Corollary 3. *Let $\mathscr{F} = (A,R,cl)$ be well-formed, $a \in A$, and the set of claims $C = cl(A)$ of F finite. Let $\sigma \in \{co, gr, pr, stb\}$. If there is a strongly accepting sub-framework for a, then here is also a strongly accepting sub-framework containing at most $|C|$ arguments.*

5. Conclusions

In this paper we revisited strongly accepting subframeworks [10,8,9] (see also [16]), and investigated their connection to structured argumentation frameworks. Based on a generic notion of such structured frameworks, we showed that strongly accepting subframeworks are applicable and generalizable from abstract AFs to structured frameworks, with the concrete formalism ABA being presented here. There is an apparent mismatch between notions on the abstract and structured level, but which can be addressed with careful consideration of, e.g., closed AFs. Moreover, we considered properties connecting to strongly accepting frameworks, in particular we showed that such strongly accepting subframeworks can be bounded polynomially, even if a "full" AF is not bounded polynomially, indicating that notions supporting explanations, based on strongly accepting subframeworks, exhibit interesting size bounds, and open up further investigation.

The most apparent future work directions include the investigation of other concrete structured argumentation formalisms and finding natural and mild conditions ensuring the converse of Proposition 5. Moreover, studying relations to other forms explainability is an intriguing avenue of future research.

References

[1] Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018.

[2] Atkinson K, Baroni P, Giacomin M, Hunter A, Prakken H, Reed C, et al. Towards Artificial Argumentation. AI Magazine. 2017;38(3):25-36.

[3] Modgil S, Prakken H. A general account of argumentation with preferences. Artificial Intelligence. 2013;195:361-97.

[4] Bondarenko A, Dung PM, Kowalski RA, Toni F. An Abstract, Argumentation-Theoretic Approach to Default Reasoning. Artificial Intelligence. 1997;93:63-101.

[5] García AJ, Simari GR. Defeasible Logic Programming: An Argumentative Approach. Theory and Practice of Logic Programming. 2004;4(1-2):95-138.

[6] Besnard P, Hunter A. Elements of Argumentation. MIT Press; 2008.

[7] Cyras K, Rago A, Albini E, Baroni P, Toni F. Argumentative XAI: A Survey. In: Proc. IJCAI. ijcai.org; 2021. p. 4392-9.

[8] Ulbricht M, Wallner JP. Strong Explanations in Abstract Argumentation. In: Proc. AAAI. AAAI Press; 2021. p. 6496-504.

[9] Niskanen A, Järvisalo M. Smallest Explanations and Diagnoses of Rejection in Abstract Argumentation. In: Proc. KR; 2020. p. 667-71.

[10] Saribatur ZG, Wallner JP, Woltran S. Explaining Non-Acceptability in Abstract Argumentation. In: Proc. ECAI. vol. 325 of Frontiers in Artificial Intelligence and Applications; 2020. p. 881-8.

[11] Prakken H, Winter MD. Abstraction in Argumentation: Necessary but Dangerous. In: Proc. COMMA. vol. 305 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2018. p. 85-96.

[12] Wallner JP. Structural constraints for dynamic operators in abstract argumentation. Argument & Computation. 2020;11(1-2):151-90.

[13] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77(2):321-57.

[14] Strass H, Wyner A, Diller M. *EMIL*: Extracting Meaning from Inconsistent Language: Towards argumentation using a controlled natural language interface. International Journal of Approximate Reasoning. 2019;112:55-84.

[15] Dvořák W, Woltran S. Complexity of abstract argumentation under a claim-centric view. Artificial Intelligence. 2020;285:103290.

[16] Brewka G, Thimm M, Ulbricht M. Strong inconsistency. Artificial Intelligence. 2019;267:78-117.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220164

Reasoning With and About Norms in Logical Argumentation

Kees VAN BERKEL [a,1] and Christian STRASSER [b]
[a] *Institute of Logic and Computation, TU Wien, Austria*
[b] *Institute for Philosophy II, Ruhr University Bochum, Germany*

Abstract. Normative reasoning is inherently defeasible. Formal argumentation has proven to be a unifying framework for representing nonmonotonic logics. In this work, we provide an argumentative characterization of a large class of Input/Output logics, a prominent defeasible formalism for normative reasoning. In many normative reasoning contexts, one is not merely interested in knowing whether a specific obligation holds, but also in why it holds despite other norms to the contrary. We propose sequent-style argumentation systems called Deontic Argument Calculi (DAC), which serve transparency and bring meta-reasoning about the inapplicability of norms to the object language level. We prove soundness and completeness between DAC-instantiated argumentation frameworks and constrained Input/Output logics. We illustrate our approach in view of two deontic paradoxes.

Keywords. Nonmonotonic logic, Argumentation, Deontic logic, Normative reasoning

1. Introduction

Obligations and norms fulfil a crucial role in a variety of fields, including law, ethics, AI, and everyday life [1]. The logical study of normative reasoning investigates reasoning with such concepts in formal systems of logic, e.g., deontic logics. Its importance increases with the development of intelligent autonomous systems. Complex normative systems often require reasoning with normative conflicts, exceptions, preferences and priorities [1]. A central challenge is to provide transparent formal models of the underlying reasoning processes, e.g., by means of nonmonotonic logics.

Over the past decades, abstract argumentation has proven to be a unifying framework for the representation of large classes of nonmonotonic logics [2]. Formal argumentation provides both a natural and a transparent model of conflicts and their resolution in terms of conflicting arguments. In this way, it provides a promising basis for tackling the challenging requirements of normative reasoning. The logical analysis of normative reasoning is well-established [1] with the Input/Output framework (I/O) being one of the central approaches [3]. Nonmonotonicity is captured in constrained I/O logics through considering maximal consistent families of norms. In recent years, also argumentative representations of deontic logics have attracted increasing interest [4,5,6,7,8,9].

[1] Corresponding Authors: Kees van Berkel (e-mail: kees@logic.at) and Christian Straßer (e-mail: christian.strasser@rub.de). Acknowledgments: This work is supported by WWTF and FWF projects MA16-028 and W1255-N23. We thank Leon van der Torre for discussions that significantly shaped this work.

This paper is the first to provide argumentative characterizations for a significant class of I/O logics, including all original logics from [3]. In this way, we are able to combine the advantages of I/O logic with those of formal argumentation. On the one hand, I/O is a highly expressive and robust framework with two decades of developments, including many applications (e.g., priorities, constitutive norms, cognitive modeling, causal reasoning [1,10]). On the other hand, it does not provide the level of transparency that comes with the explicit representation of conflicts in formal argumentation.

In particular, I/O leaves some central challenges of normative reasoning unaddressed. When answering the question as to *why* an obligation holds, one must state *reasons*. Moreover, often it does not suffice to know why a specific obligation holds, one must know why other obligations to the contrary *do not hold*. E.g., in order to understand why "I am permitted to overtake on the left, *despite* having to drive on the right" one must know how the first norm relates to the second. In this case, the first is an exception that renders the latter *inapplicable* in the context of "overtaking another vehicle". Common approaches to I/O logics—as well as deontic logics—do not provide means for making explicit the reasons why certain obligations are not derivable. Despite their central role in ethics and explanation [11], a general lack of explicit modeling of reasons in formal systems has been recently identified [12] (with some exceptions, e.g., [13]). Support and defeat relations are central in the context of reasons as well as in formal argumentation, which makes the latter an ideal framework to reason with and about reasons.

We address these problems by introducing a class of rule-based proof systems called *Deontic Argument Calculi* (DAC) for normative reasoning by means of argumentation.

Our conceptual contribution is twofold: First, we use labels on formulae to make the presentation transparent on the object level, i.e., we can syntactically distinguish between facts, obligations, and constraints without "burdening" the logics with modalities [10]. Second, we internalize some of the meta-reasoning in the I/O formalism by referring to the inapplicability of norms on the object language level. Consequently, our calculi generate both arguments that provide *explicit reasons* for obligations and arguments that defeat other arguments by giving explicit reasons for why certain norms are inapplicable. The second type of arguments concerns the nonmonotonicity of normative reasoning. The possibility to reason about the inapplicability of norms on the object language level distinguishes our work from other systems such as [14,15]. We illustrate the utility of our approach using the notion of *related admissibility* [16] to explain why some obligation holds *despite* certain norms to the contrary.

The technical contribution of this work contains two types of completeness results: First, we show adequacy between DAC and a significant class of monotonic I/O consequence relations. Second, we prove that formal argumentation frameworks instantiated with DAC arguments characterize a large class of nonmonotonic I/O logics. This makes our work the first to argumentatively characterize I/O logic. Moreover, DAC enjoys a modularity particularly suitable for expansions and our calculi are modular with respect to a large class of base logics. Last, our work contributes to previous representation results in formal argumentation concerning systems based on maximal consistent sets [2].

2. Basic Terminology and Benchmark Examples

We introduce basic terminology by considering two examples. Developments in deontic logic are driven by challenging examples [17]. Here, we focus on *contrary-to-duty* rea-

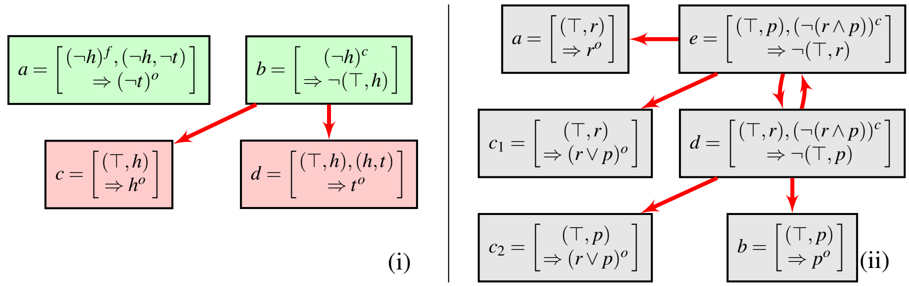

$a = \left[\begin{smallmatrix}(\neg h)^f, (\neg h, \neg t)\\ \Rightarrow (\neg t)^o\end{smallmatrix}\right]$
$b = \left[\begin{smallmatrix}(\neg h)^c\\ \Rightarrow \neg(\top, h)\end{smallmatrix}\right]$
$c = \left[\begin{smallmatrix}(\top, h)\\ \Rightarrow h^o\end{smallmatrix}\right]$
$d = \left[\begin{smallmatrix}(\top, h), (h, t)\\ \Rightarrow t^o\end{smallmatrix}\right]$
(i)
$a = \left[\begin{smallmatrix}(\top, r)\\ \Rightarrow r^o\end{smallmatrix}\right]$
$e = \left[\begin{smallmatrix}(\top, p), (\neg(r \wedge p))^c\\ \Rightarrow \neg(\top, r)\end{smallmatrix}\right]$
$c_1 = \left[\begin{smallmatrix}(\top, r)\\ \Rightarrow (r \vee p)^o\end{smallmatrix}\right]$
$d = \left[\begin{smallmatrix}(\top, r), (\neg(r \wedge p))^c\\ \Rightarrow \neg(\top, p)\end{smallmatrix}\right]$
$c_2 = \left[\begin{smallmatrix}(\top, p)\\ \Rightarrow (r \vee p)^o\end{smallmatrix}\right]$
$b = \left[\begin{smallmatrix}(\top, p)\\ \Rightarrow p^o\end{smallmatrix}\right]$
(ii)

Figure 1. Defeasible normative reasoning examples: (i) The Chisholm scenario (Example 1). Arrows denote defeat relations between arguments, relative to $\mathscr{C}' = \{\neg h^c\}$ (Example 2). (ii) A deontic conflict (Example 3). Argument e defends $\{b, c_2, e\}$, whereas argument d defends $\{a, c_1, d\}$.

soning and *deontic conflicts*. Both can be effectively addressed using nonmonotonic reasoning [10] (for alternative approaches see [1]). The language $\mathscr{L}$ is defined as follows:

$$\varphi ::= p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

with $p \in \mathsf{Atoms}$. All connectives are primitive in order to be modular with respect to the base logic (Section 3). We use $p, q, r, \ldots$ for atoms, and reserve $\varphi, \psi, \theta, \ldots$ for arbitrary formulae of $\mathscr{L}$. In order to increase transparency we *label* formulae of $\mathscr{L}$, i.e., $\mathscr{L}^i = \{\varphi^i \mid \varphi \in \mathscr{L}\}$ for $i \in \{f, o, c\}$. We have formulae expressing *facts* $\mathscr{L}^f$, *obligations* $\mathscr{L}^o$, and *constraints* $\mathscr{L}^c$. Moreover, we employ pairs of formulae $\mathscr{L}^n = \{(\varphi, \psi) \mid \varphi, \psi \in \mathscr{L}\}$. A pair (φ, ψ) represents a *norm*: i.e., "given fact φ, it is obligatory that ψ" [3].

We work with knowledge bases of the type $\langle \mathscr{F}, \mathscr{N}, \mathscr{C} \rangle$, where $\mathscr{F} \subseteq \mathscr{L}^f$ constitutes the factual context, $\mathscr{N} \subseteq \mathscr{L}^n$ denotes a system of norms, and $\mathscr{C} \subseteq \mathscr{L}^c$ represents the constraints with which output must be consistent. The basic idea is that facts (input) trigger norms, from which obligations are detached (output). Moreover, constraints control the output to ensure consistency. The above is in the spirit of constrained I/O logic [3].

Suppose we have a single fact $\mathscr{F} = \{p^f\}$, a norm system $\mathscr{N} = \{(p, q)\}$, and no constraints, then an argument concluding that q is obligatory is of the following form,

$$p^f, (p, q) \Rightarrow q^o$$

The left-hand side (lhs) gives reasons for the conclusion on the right-hand side (rhs).

Example 1 (Chisholm scenario [1], Figure 1-i)**.** *Jones is under the obligation to go and help her neighbors* $(\top, h)$.[2] *Furthermore, Jones knows if she goes to help, she must tell them she goes* (h, t). *Now, if Jones does not go, she ought not to tell them she goes* $(\neg h, \neg t)$. *It turns out that Jones does not go to help* $\neg h^f$. *Clearly, Jones has violated her primary obligation to go and help. Let the knowledge base be* $\mathscr{F} = \{\neg h^f\}$ *and* $\mathscr{N} = \{(\top, h), (h, t), (\neg h, \neg t)\}$. *Figure 1-i presents arguments a, c, and d that can be constructed from the knowledge base (we explain the meaning of b and the arrows in Example 2); e.g., in argument a, the reasons for not telling* $\neg t^o$ *are the fact* $\neg h^f$ *and the norm* $(\neg h, \neg t)$. *What must Jones do in this* contrary-to-duty *scenario? The desired answer is that she ought not to tell the neighbors she goes* $\neg t^o$. *Formalizations of this scenario cause problems for (monadic) deontic logics, e.g., both t and* $\neg t$ *become obligatory.*

[2] A norm $(\top, \varphi)$ with a precondition $\top$ is triggered by default, that is, even by an empty factual context.

Arguments do not only provide reasons in support of an obligation, but also defend them from potential *defeaters*. A rebuttal defeat opposes the conclusion of an argument without pinpointing the reason as to why. In contrast, attacks on reasons—i.e., *undercuts*—are arguments that express *which reasons* are *inapplicable* given some context. We adopt undercuts since they are more transparent about attacks. Recall that constraints are consistency requirements and suppose $\mathscr{C} = \{\neg q^c\}$.[3] Then, a defeating argument

$$p^f, \neg q^c \Rightarrow \neg(p,q)$$

expresses that, if output is to be consistent with the constraint $\neg q^c$, in context p^f the norm (p,q) cannot be consistently asserted as a reason (since it would detach q^o). Hence, $\neg(p,q)$ expresses that this norm is *inapplicable* given $\mathscr{F}$ and $\mathscr{C}$. An argumentation framework is then simply a set of arguments with defeat relations holding between them.

Example 2 (Example 1 cont.)**.** *We want to know what Jones must do in the light of her violation $\neg h^f$. Thus, we impose the constraint that the output must be consistent with the fact that Jones does not help $\mathscr{C} = \{\neg h^c\}$ (i.e., $\mathscr{C} = \mathscr{F}$ modulo relabelling). This constraint gives us the argument $b : \neg h^c \Rightarrow \neg(\top, h)$ expressing that given consistency requirement $\neg h$, the norm $(\top, h)$ may not be asserted as a reason (it would output the inconsistent h^o). This argument serves as a defeater of any argument which appeals to $(\top, h)$ in its reasons, in this case both c and d; see the defeat arrows in Figure 1-i. So, that Jones ought not to tell $\neg t^o$ is explained by argument a together with the fact that arguments c and d concluding helping h^o, respectively telling t^o, cannot be defended in view of b. Namely, c and d both employ reasons that are inapplicable given $\mathscr{C}$.*

What makes this approach more transparent is the use of labels in arguments to indicate different types of information (factual, obligations, constraints), the internalized meta-reasoning about inapplicability of norms, and the argumentation framework revealing the contrastive dimension of defeasible reasoning. In Figure 1-i, the question "why shouldn't Jones help, *despite* argument c?" is answered by "since argument b attacks c and b is not attacked." These notions will be made precise in subsequent sections.

Example 3 (Deontic Conflict [1], Figure 1-ii)**.** *Suppose Smith has an obligation to return a borrowed weapon to a colleague $(\top, r)$. Smith knows the colleague is planning to commit a crime with this weapon and Smith is under the obligation to prevent crime $(\top, p)$. Furthermore, the constraint is that Smith cannot secure both r and p. What should Smith do? This is a* deontic conflict. *The knowledge base is $\mathscr{F} = \emptyset$, $\mathscr{N} = \{(\top, r), (\top, p)\}$, and $\mathscr{C} = \{\neg(r \wedge p)^c\}$. Suppose we reason classically, e.g., p entails $p \vee r$. The arguments that can be constructed are presented in Figure 1-ii. The two defeating arguments, d and e, express that given the constraints one of either two norms cannot be asserted.*

Intuitively, the defensible set $\{a, c_1, d\}$ justifies the obligation that Smith ought to return the weapon in Example 3, whereas $\{b, c_2, e\}$ does this for the prevention of crime. Likewise, one can justify the floating conclusion $(r \vee p)^o$ in Figure 1-ii, by arguing that in every defensible stance *either* c_1 *or* c_2 is selected (cf. disjunctive response [13]). However, following a more skeptical reasoning style one can argue why $r \vee p$ is not obligatory since there is no single argument concluding $(r \vee p)^o$ that is selected in *every* defensible stance. Defeasible reasoning by means of argumentation gives rise to various reasoning styles, including the aforementioned. We will discuss these in Section 5.

[3] Since we do not allow for formulae with mixed labels, we can safely omit brackets w.r.t. using labels.

$$\frac{}{(\top,\top)}\,\mathrm{T} \qquad \frac{}{(\varphi,\varphi)}\,\mathrm{ID} \qquad \frac{(\varphi,\psi) \qquad \psi\vdash\gamma}{(\varphi,\gamma)}\,\mathrm{WO} \qquad \frac{(\varphi,\psi) \qquad (\varphi,\gamma)}{(\varphi,\psi\wedge\gamma)}\,\mathrm{AND}$$

$$\frac{(\varphi,\psi) \qquad \gamma\vdash\varphi}{(\gamma,\psi)}\,\mathrm{SI} \qquad \frac{(\varphi,\psi) \qquad (\varphi\wedge\psi,\gamma)}{(\varphi,\gamma)}\,\mathrm{CT} \qquad \frac{(\varphi,\psi) \qquad (\gamma,\psi)}{(\varphi\vee\gamma,\psi)}\,\mathrm{OR}$$

Figure 2. Rules for constructing $\mathsf{deriv}_{\mathscr{R},\mathsf{L}}$ proof-systems. The minimal set of deriv-rules is $\{\mathrm{WO},\mathrm{AND},\mathrm{SI}\}$.

3. Constrained Input/Output Logic

We briefly recall the basics of *Constrained Input/Output logic*, the systems for which we provide argumentative characterizations. The formalism was developed by Makinson and van der Torre [3] and is particularly suitable for normative reasoning [10]. Its central feature is the employment of syntactic objects of the form (φ,ψ), called *norms*.

I/O logics are construed over the *non*-labelled propositional language $\mathscr{L}$ (Section 2) and a base logic L. We use capital Greek letters $\Delta,\Gamma,\ldots$ for finite sets of $\mathscr{L}$-formulae and write $\bigwedge\Delta$ to denote the conjunction of elements of Δ. Let $\vdash$ denote the consequence relation of the base logic L. We assume that $\vdash$ is reflexive ($\Gamma,\varphi\vdash\varphi$), transitive ($\Gamma\vdash\psi$ and $\Gamma',\psi\vdash\varphi$ implies $\Gamma,\Gamma'\vdash\varphi$) and monotonic ($\Gamma\vdash\varphi$ implies $\Gamma,\Gamma'\vdash\varphi$). We also assume the presence of a conjunction $\wedge$, for which $\Gamma\vdash\psi\wedge\varphi$ iff $\Gamma\vdash\psi$ and $\Gamma\vdash\varphi$, a negation $\neg$, for which $\Gamma\vdash\varphi$ iff $\Gamma,\neg\varphi\vdash$, a disjunction $\vee$, for which $\Gamma,\varphi_1\vdash\psi$ and $\Gamma,\varphi_2\vdash\psi$ iff $\Gamma,\varphi_1\vee\varphi_2\vdash\psi$, and a falsum constant $\bot$ for which $\bot\vdash\varphi$ and $\varphi,\neg\varphi\vdash\bot$. We assume L has an adequate *sequent calculus* LC, i.e., $\Delta\vdash\varphi$ iff the sequent $\Delta\Rightarrow\varphi$ is LC-derivable.

Constrained I/O logics work with knowledge bases of the type $\langle\mathscr{F},\mathscr{N},\mathscr{C}\rangle$, where $\mathscr{F}\subseteq\mathscr{L}$ is the *factual* input, $\mathscr{N}\subseteq\mathscr{L}\times\mathscr{L}$ a *normative system*, and $\mathscr{C}\subseteq\mathscr{L}$ a set of *constraints* containing the formulae with which output must be consistent. We assume $\mathscr{F}$ and $\mathscr{C}$ to be consistent, i.e., $\mathscr{F}\nvdash\bot$ and $\mathscr{C}\nvdash\bot$. The traditional I/O proof systems are only available for a class of *monotonic* I/O logics [10]. The system is referred to as "deriv" and contains inference rules that derive I/O pairs from other I/O pairs (Figure 2).

Definition 1. *Let* $\mathsf{deriv}_{\mathscr{R},\mathsf{L}}$ *be a proof-system, with* $\mathscr{R}$ *a set of rules from Table 2. Let* L *be the base logic, and let* $\mathscr{N}\subseteq\mathscr{L}^n$. A derivation *of* $(\varphi,\psi)\in\mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\mathscr{N})$ *is a* tree *of rule-applications of* $\mathscr{R}$ *where the leaves are either members of* $\mathscr{N}$ *or instances of T and ID (if T, ID* $\in\mathscr{R}$*), all members of* $\mathscr{N}$ *are among the leaves, and the root is* (φ,ψ).

We say ψ *is* obligatory *(detached) under* $\mathscr{N}$ *and* $\mathscr{F}$ *if* $(\varphi,\psi)\in\mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\mathscr{N}')$ *with* $\mathscr{F}\vdash\varphi$ *and* $\mathscr{N}'\subseteq\mathscr{N}$. *We write* $\psi\in\mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\Delta,\mathscr{N})$ *if* $(\bigwedge\Delta,\varphi)\in\mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\mathscr{N})$.

Paradigmatic I/O logics are characterized by the sets of rules $\mathscr{R}_1=\{\mathrm{T},\mathrm{WO},\mathrm{SI},\mathrm{AND}\}$, $\mathscr{R}_2=\{\mathrm{OR}\}\cup\mathscr{R}_1$, $\mathscr{R}_3=\mathscr{R}_1\cup\{\mathrm{CT}\}$, and $\mathscr{R}_4=\mathscr{R}_2\cup\mathscr{R}_3$. The system $\mathscr{R}_1$ represents a *single deontic detachment* procedure which allows for weakening of the output (WO), combining output (AND), and strengthening of the input (SI). All propositional tautologies are among the output (T). System $\mathscr{R}_2$ extends $\mathscr{R}_1$ with *reasoning by cases* (OR), i.e., if both φ and γ generate output ψ, then $\varphi\vee\gamma$ generates ψ too. System $\mathscr{R}_3$ extends $\mathscr{R}_1$ with reusability (CT) allowing for iterations of *successive deontic detachment* (cf. chaining reasons in Example 5). Last, $\mathscr{R}_4$ combines $\mathscr{R}_2$ and $\mathscr{R}_3$. The above systems may be closed under *throughput* (ID), i.e., input is 'put through' as output. We write $\mathscr{R}_i^+=\mathscr{R}_i\cup\{\mathrm{ID}\}$ for $i\in\{1,2,3,4\}$. The resulting eight systems are sound and complete with respect to their semantic characterizations [10]. We omit the semantics here.

The above systems are still monotonic. As Example 1 and 3 demonstrate, we require defeasible detachment. Constrained I/O logics enable this [3]. Constrained I/O logics work with maximal families of norms $\mathscr{N}' \subseteq \mathscr{N}$ under which the output remains consistent with the constraints $\mathscr{C}$. If the output is required to be consistent *per se*, we let $\mathscr{C} = \emptyset$. If the output is to be consistent with the input, we take $\mathscr{F} \subseteq \mathscr{C}$ (e.g., Example 2).

Definition 2. *Let* $\mathsf{deriv}_{\mathscr{R},\mathsf{L}}$ *be a system from Figure 2 and let* $\mathscr{K} = \langle \mathscr{F}, \mathscr{N}, \mathscr{C} \rangle$ *be a knowledge base. The set of* maximal consistent families *of* $\mathscr{N}$ *(*maxfam*) is defined as:*

- $\mathsf{maxfam}_{\mathscr{R},\mathsf{L}}(\mathscr{K})$ *is the set of* max-elements *of* $\{\mathscr{N}' \subseteq \mathscr{N} \mid$ *for all* $(\varphi, \psi) \in \mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\mathscr{N}')$*, if* $\mathscr{F} \vdash \varphi$*, then* $\mathscr{C}, \psi \nvdash \bot\}$*.*

We define sceptic nonmonotonic inference $\mid\!\sim^{s}$ *for* constrained I/O logic *as follows:*

- $\mathscr{K} \mid\!\sim^{s}_{\mathscr{R},\mathsf{L}} \varphi$ *iff* $\forall\, \mathscr{N}' \in \mathsf{maxfam}_{\mathscr{R},\mathsf{L}}(\mathscr{K})$*,* $\exists (\psi, \varphi) \in \mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\mathscr{N}')$ *with* $\mathscr{F} \vdash \psi$*.*

Example 4 (Example 1 cont.)**.** *Consider* $\mathscr{R}_3$ *with* L *a classical logic,* $\mathscr{F} = \{\neg h\}$*, and* $\mathscr{N} = \{(\top,h),(h,t),(\neg h,\neg t)\}$*. For* $\mathscr{C} = \emptyset$*, we have* $\mathsf{maxfam}_{\mathscr{R}_3,\mathsf{L}}(\mathscr{F},\mathscr{N},\mathscr{C}) = \{\{(\top,h),(h,t)\},\{(\neg h,\neg t),(\top,h)\},\{(\neg h,\neg t),(h,t)\}\}$*. We derive* $(\top, h \wedge t) \in \mathsf{deriv}_{\mathscr{R}_3,\mathsf{L}}(\{(\top,h),(h,t)\})$ *as follows:*

$$\dfrac{(\top,h) \qquad \dfrac{(\top,h) \qquad \dfrac{(h,t) \qquad \top \wedge h \vdash h}{(\top \wedge h, t)}\ \mathrm{SI}}{(\top,t)}\ \mathrm{CT}}{(\top, h \wedge t)}\ \mathrm{AND}$$

with $\mathscr{F} \vdash \top$ *and* $\mathscr{C}, h \wedge t \nvdash \bot$*. However, once we set the constraints to Jones' violation, i.e.,* $\mathscr{C}' = \mathscr{F}$*, we obtain a single* maxfam *member* $\mathscr{N}' = \{(\neg h,\neg t),(h,t)\}$ *since now* $\mathscr{C}', h \vdash \bot$ *whereas* $\mathscr{C}', \neg t \nvdash \bot$ *(note that* $(h,t) \in \mathscr{N}'$ *cannot be triggered by* $\mathscr{F}$*). Given* $\mathscr{C}'$*, Jones is obliged to not tell, i.e.,* $\mathscr{K} \mid\!\sim^{s}_{\mathscr{R},\mathsf{L}} \neg t$*, and is not obliged to help, i.e.,* $\mathscr{K} \not\mid\!\sim^{s}_{\mathscr{R},\mathsf{L}} h$*.*

First, maxfam sets (of arbitrary size) do not provide formal ways of pinpointing the reasons why some norms are inapplicable, e.g., why $(\top,h)$ in Example 4 is inapplicable given $\mathscr{C} = \mathscr{F}$. Second, deriv is unsuitable for generating transparent arguments, e.g., as a certificate the derivation in Example 4 may justify *that* $(\top, h \wedge t)$ is derivable, its conclusion does not explain *why* $h \wedge t$ is obligatory. In fact, although in general a derivation is a justification, it is not necessarily an explanation. Our calculi address both challenges.

4. Deontic Argument Calculi (DAC)

In order to generate more transparent I/O arguments, we label propositional formulae as facts $\mathscr{L}^f$, obligations $\mathscr{L}^o$, and constraints $\mathscr{L}^c$ (Section 2). What is more, we allow for Boolean operations over the more complex meta-logical objects (φ, ψ) denoting norms. Operations over these higher-order syntactic objects enable undercuts that explain why certain norms should (not) be applied. For the present work, it suffices to consider negation only. Let $\overline{\mathscr{L}^n} = \{\neg(\varphi,\psi) \mid (\varphi,\psi) \in \mathscr{L}^n\}$. The *language of norms* is defined as $\mathscr{L}^n \cup \overline{\mathscr{L}^n}$. Furthermore, let $\mathscr{L}^{io} = \mathscr{L}^f \cup \mathscr{L}^o \cup \mathscr{L}^c \cup \mathscr{L}^n \cup \overline{\mathscr{L}^n}$ be the full labelled I/O language. In $\mathscr{L}^{io}$, norms are integrated into the object-level language. We write φ for an arbitrary formula of $\mathscr{L}^{io}$ and write Δ^i to denote that $\Delta^i \subseteq \mathscr{L}^i$ for $i \in \{f,o,c,n\}$.

We introduce *Deontic Argument Calculi* (DAC) for I/O logic. These calculi are *sequent-style* calculi, which are rule-based proof systems employing syntactic objects of the form $\Delta \Rightarrow \Gamma$, with $\Delta, \Gamma \subseteq \mathscr{L}^{io}$ and '$\Rightarrow$' as a sequent arrow. We call $\Delta \Rightarrow \Gamma$ a sequent or an argument, where Δ denotes the reasons for Γ (Section 2). Furthermore, Δ is inter-

Ax $\vdash_{\mathsf{LC}} \Delta^i \Rightarrow \Gamma^i$, for $i \in \{f,o,c\}$ **Taut** $\Rightarrow (\top,\top)$ **Detach** $\varphi^f,(\varphi,\psi) \Rightarrow \psi^o$ **TP** $\varphi^f \Rightarrow \varphi^o$

$$\textbf{R-C}\ \frac{\Delta \Rightarrow \varphi^o}{\Delta, (\neg\varphi)^c \Rightarrow} \qquad \textbf{R-N}\ \frac{\Delta, (\varphi,\psi) \Rightarrow}{\Delta \Rightarrow \neg(\varphi,\psi)} \qquad \textbf{L-CT}^a\ \frac{\varphi^f, \Delta \Rightarrow \Theta}{\varphi^o, \Delta \Rightarrow \Theta}$$

$$\textbf{L-OR}^b\ \frac{\Delta, \varphi^f \Rightarrow \Theta \qquad \Delta', \psi^f \Rightarrow \Theta}{\Delta, \Delta', (\varphi \vee \psi)^f \Rightarrow \Theta} \qquad \textbf{Cut}^c\ \frac{\Delta \Rightarrow \varphi \qquad \varphi, \Delta' \Rightarrow \Theta}{\Delta, \Delta' \Rightarrow \Theta}$$

Figure 3. Rules for building $\mathsf{DAC}_{\mathscr{S}}$ (Definition 3). The upper level contains initial sequents and the lower level logical and structural rules. Side-condition (a) denotes $\Delta \cap \mathscr{L}^n \neq \emptyset$; (b) denotes that if **TP** $\notin \mathscr{S}$, then $\Delta \cap \mathscr{L}^n \neq \emptyset$ and $\Delta' \cap \mathscr{L}^n \neq \emptyset$; and (c) that $\varphi \in \mathscr{L}^{io}$.

preted as a regular finite set and Γ is restricted to at most one formula. The use of regular sets instead of multi-sets is more modular w.r.t. the base logic L. Let LC be an adequate sequent calculus for the base logic L, then, intuitively, DAC takes *labelled* versions of any LC-derivable $\Delta \Rightarrow \Gamma$ as an initial sequent (i.e., $\Delta^i \Rightarrow \Gamma^i$ for each $i \in \{f,o,c\}$) and contains logical- and structural rules for transforming labelled formulae of $\mathscr{L}^{io}$ (see Figure 3).

Definition 3. *Let* DAC *be the base system with the underlying logic* L*, containing the rules* **Ax**, **Detach**, **R-C**, **R-N***, and* **Cut** *from Figure 3. The calculus* $\mathsf{DAC}_{\mathscr{S}}$ *extends* DAC *with the set of rules* $\mathscr{S} \subseteq \{$**Taut**,**TP**,**L-OR**,**L-CT**$\}$*, leading to 16* DAC*-axiomatizations.*

A $\mathsf{DAC}_{\mathscr{S}}$*-derivation of* $\Delta \Rightarrow \Gamma$ *is a tree whose leafs are initial sequents of* $\mathsf{DAC}_{\mathscr{S}}$*, whose root is* $\Delta \Rightarrow \Gamma$*, and whose rule-applications are instances of the rules of* $\mathsf{DAC}_{\mathscr{S}}$*. We write* $\vdash_{\mathscr{S}} \Delta \Rightarrow \Gamma$ *(resp.* $\vdash^n_{\mathscr{S}} \Delta \Rightarrow \Gamma$*) if* $\Delta \Rightarrow \Gamma$ *is* $\mathsf{DAC}_{\mathscr{S}}$*-derivable (in at most n steps).*

Since $\mathsf{DAC}_{\mathscr{S}}$ takes labelled LC-derivable sequents as initial sequents, the rules of LC are not part of $\mathsf{DAC}_{\mathscr{S}}$. Still, LC rules can be straightforwardly shown admissible in DAC due to the presence of **Cut**. The rule **Taut** ensures that all propositional tautologies are considered as output. The rule **Detach** is an initial explanatory argument stating that the fact φ and the norm (φ,ψ) are reasons for the obligation ψ. Instead of deriving pairs from other pairs (as in deriv), we keep norms as primitive reasons from a given normative code $\mathscr{N}$ and only modify facts, obligations, and constraints. This gives us some explanatory advantages (see **R-C** and **R-N** below). The rule **TP** corresponds to throughput. The rule **L-CT** corresponds to successive detachment, expressing that a norm may likewise be triggered by the output of some other norm (cf. Example 5). **L-OR** reflects reasoning by cases over input. The side-condition on **L-OR** is dropped for **TP** $\in \mathscr{S}$ due to reasoning by cases with throughput. **Cut** suffices as the only structural rule.

More interesting are the rules **R-C** and **R-N**. Concerning **R-C**, think of a sequent with an empty right-hand side as an argument expressing inconsistent reasons. For instance, an argument $\varphi^f, (\varphi,\psi), (\neg\psi)^c \Rightarrow$ explains that the fact φ and the norm (φ,ψ) (which are reasons for ψ) are inconsistent whenever the output must be consistent with $\neg\psi$. What is more, whenever such an argument expresses inconsistent reasons, we know at least one of its norms is inapplicable. The rule **R-N** expresses this: from $\varphi^f, (\varphi,\psi), (\neg\psi)^c \Rightarrow$ we obtain the defeating argument $\varphi^f, (\neg\psi)^c \Rightarrow \neg(\varphi,\psi)$. Hence, φ^f and $(\neg\psi)^c$ are reasons for the *inapplicability* of the norm (φ,ψ). $\mathsf{DAC}_{\mathscr{S}}$ sequents will be the building blocks for the desired argumentative characterizations (Section 5).

Example 5 (Example 1 cont.)**.** *The* DAC*-argument d (Figure 1-i), concluding that Jones should tell her neighbors she is coming to help, is derived through chaining* $(\top,h)$ *and* (h,t)*. The following* $\mathsf{DAC}_{\mathscr{S}}$*-derivation (left) shows this, where* **L-CT** $\in \mathscr{S}$*:*

$$\dfrac{\dfrac{}{\top^f,(\top,h)\Rightarrow h^o}\,\textbf{Detach} \qquad \dfrac{\dfrac{}{h^f,(h,t)\Rightarrow t^o}\,\textbf{Detach}}{h^o,(h,t)\Rightarrow t^o}\,\textbf{L-CT}}{\top^f,(\top,h),(h,t)\Rightarrow t^o}\,\textbf{Cut} \qquad \dfrac{\dfrac{\dfrac{}{\top^f,(\top,h)\Rightarrow h^o}\,\textbf{Detach}}{\top^f,(\neg h)^c,(\top,h)\Rightarrow}\,\textbf{R-C}}{\top^f,(\neg h)^c\Rightarrow\neg(\top,h)}\,\textbf{R-N}$$

Given $\mathscr{C}'=\{\neg h^c\}$, "why should Jones not tell, despite *argument d?" is answered by the (right) derivable argument b (Figure 1-i). The fact $\top^f$ is omitted by a* **Cut** *with $\Rightarrow\top^f$.*

Example 6 (Example 3 cont.). *In the dilemma, Smith cannot both return the weapon and prevent the crime. So, we find $(\top,r)$ applicable if and only if $(\top,p)$ is inapplicable. This is expressed by arguments e and f. The* $\mathsf{DAC}_{\mathscr{S}}$*-derivations of e and f from Figure 1-ii are obtained similarly to argument b in Example 5, using* **Detach** *twice, the* DAC*-admissible rule from* LC *for right conjunction introduction,* **R-C**, and **R-N** *consecutively.*

5. Argumentation and DAC-Instantiations

DAC arguments are of two types: they either give reasons for obligations, or they give reasons for why certain norms are inapplicable, i.e., defeated. The latter arguments capture the defeasibility of normative reasoning and define the interaction among arguments. We define DAC-induced *argumentation frameworks* (*AF*s) to model this interaction.

Definition 4. *Let* $\mathsf{DAC}_{\mathscr{S}}$ *be a calculus and $\mathscr{K}=\langle\mathscr{F},\mathscr{N},\mathscr{C}\rangle$ a labelled knowledge base (i.e., $\mathscr{F}\subseteq\mathscr{L}^f,\mathscr{N}\subseteq\mathscr{L}^n$, and $\mathscr{C}\subseteq\mathscr{L}^c$). We define $\mathrm{AF}_{\mathscr{S}}(\mathscr{K})=\langle\mathsf{Arg},\mathsf{Att}\rangle$ as follows:*

- $\Delta\Rightarrow\Gamma\in\mathsf{Arg}$ *iff $\Delta\Rightarrow\Gamma$ is* $\mathsf{DAC}_{\mathscr{S}}$*-derivable, $\Delta\subseteq\mathscr{F}\cup\mathscr{N}\cup\mathscr{C}$, and $\Gamma\subseteq\mathscr{L}^{io}$;*
- *a* defeats *b, i.e., $(a,b)\in\mathsf{Att}$ iff $a=\Delta\Rightarrow\neg(\varphi,\psi)$ and $b=\Gamma,(\varphi,\psi)\Rightarrow\Theta$.*

We write $\mathsf{Arg}(\Sigma)$ to denote the set of $\mathsf{DAC}_{\mathscr{S}}$*-arguments $\Delta\Rightarrow\Gamma$ for which $\Delta\subseteq\Sigma\subseteq\mathscr{L}^{io}$.*

For an $AF_{\mathscr{S}}(\mathscr{K})$ it suffices to only consider arguments relevant to $\mathscr{K}$, i.e., $\mathsf{Arg}(\mathscr{F}\cup\mathscr{N}\cup\mathscr{C})$. We are interested in what combinations of arguments (*extensions*) can be collectively accepted given an *AF*. For our purpose, stable extensions suffice.

Definition 5. *Let $\langle\mathsf{Arg},\mathsf{Att}\rangle$ be an* AF *and let $\mathscr{E}\subseteq\mathsf{Arg}$:*

- $\mathscr{E}$ defeats *an argument $a\in\mathsf{Arg}$ if there is a $b\in\mathscr{E}$ that defeats a, i.e., $(b,a)\in\mathsf{Att}$;*
- $\mathscr{E}$ *is* conflict-free *if it does not defeat any of its own elements;*
- $\mathscr{E}$ *is* stable *if it is conflict-free and defeats all $b\in\mathsf{Arg}\setminus\mathscr{E}$.*

Let `Stable` *be the set of stable extensions of* AF. *We define sceptic (s), sceptic* (s*), and credulous (c) nonmonotonic inference as follows:*

- $\mathrm{AF}\mid\!\sim^{s}_{\text{stable}}\varphi$ *iff for each $\mathscr{E}\in$* `Stable`*, there is an $a\in\mathscr{E}$ concluding φ;*
- $\mathrm{AF}\mid\!\sim^{s^*}_{\text{stable}}\varphi$ *iff there is an $a\in\bigcap$*`Stable` *concluding φ;*
- $\mathrm{AF}\mid\!\sim^{c}_{\text{stable}}\varphi$ *iff there is a $\mathscr{E}\in$* `Stable` *s.t. there is an $a\in\mathscr{E}$ concluding φ.*

The use of DAC-arguments introduces nuances in sceptic inference: e.g., the distinction between s and s^* corresponds to the discussion of floating conclusions in Section 2.

Example 7 (Example 3 cont.). *Smith is in a dilemma of conflicting duties. The* AF *of Figure 1-ii represents this conflict, where $\mathsf{Arg}=\{a,b,c_1,c_2,d,e\}$ and $\mathsf{Att}=\{(e,a),(e,c_1),$*

Table 1. Lemmas for $\vdash_{\mathscr{S}}$. Let $\Delta^{\downarrow}$ and $\varphi^{\downarrow}$ be the set of formulae in Δ, resp. φ stripped from any labels.

Lemma :	if	then
1	$\vdash_{\mathscr{S}} \Delta, \Gamma_1^c \Rightarrow \Sigma$ and $\mathscr{C} \vdash \bigwedge \Gamma_1$	$\exists \Gamma_2 \subseteq \mathscr{C} : \vdash_{\mathscr{S}} \Delta, \Gamma_2^c \Rightarrow \Sigma$ and $\Gamma_2 \vdash \bigwedge \Gamma_1$
2	$\vdash^n_{\mathscr{S}} \Delta \Rightarrow \neg(\varphi, \psi)$	$\vdash^n_{\mathscr{S}} \Delta, (\varphi, \psi) \Rightarrow$
3		$\Delta^{\downarrow} \vdash \gamma^{\downarrow}$, where $\Delta \subseteq \mathscr{L}^f \cup \mathscr{L}^o \cup \{(\top, \top)\}$, $\gamma \in \mathscr{L}^f \cup \mathscr{L}^o$
4		$\vdash \gamma^{\downarrow}$, where $\mathbf{TP} \notin \mathscr{S}$, $\Delta \subseteq \mathscr{L}^f \cup \{(\top, \top)\}$, $\gamma \in \mathscr{L}^o$
5	$\vdash^n_{\mathscr{S}} \Delta \Rightarrow$	$\vdash^n_{\mathscr{S}} \Delta \setminus \mathscr{L}^c \Rightarrow \varphi^o$ s.t. $\varphi \vdash \neg \bigwedge (\Delta \cap \mathscr{L}^c)^{\downarrow}$, where $\neg \bigwedge \emptyset =_{\mathsf{df}} \bot$

$(e,d), (d,e), (d,c_2), (d,b)\}$. *It has two* stable *extensions* $\{a, c_1, d\}$ *and* $\{b, c_2, e\}$, *defending the views that Smith ought to return the weapon, resp. prevent the crime. Hence,* $\mathrm{AF} \mid\!\sim^{c}_{\text{stable}} r^o, p^o$, *whereas* $\mathrm{AF} \mid\!\not\sim^{c}_{\text{stable}} (r \wedge p)^o$, $\mathrm{AF} \mid\!\not\sim^{s}_{\text{stable}} r^o$, *and* $\mathrm{AF} \mid\!\not\sim^{s}_{\text{stable}} p^o$. *For the floating conclusion* $(r \vee p)^o$ *we have* $\mathrm{AF} \mid\!\sim^{s}_{\text{stable}} (r \vee p)^o$ *but* $\mathrm{AF} \mid\!\not\sim^{s^*}_{\text{stable}} (r \vee p)^o$. *(The* AF *of Example 2 in Figure 1-i has one stable extension* $\{a, b\}$, *and so* $\mathrm{AF} \mid\!\sim^{s,s^*,c}_{\text{stable}} (\neg t)^o$.*)*

To illustrate the utility of our approach, we consider the notion of related admissibility [16]. An extension $\mathscr{E}$ is *admissible* if it is conflict-free and $\mathscr{E}$ defeats all arguments defeating some $a \in \mathscr{E}$. An argument *a defends b* iff $a = b$, or there is a c s.t. a defeats c and c defeats b, or there is a c s.t. a defends c and c defends b. A set $\mathscr{E}_a \subseteq \mathsf{Arg}$ is *related admissible with topic a* iff $a \in \mathscr{E}_a$, for all $b \in \mathscr{E}_a$, b defends a, and $\mathscr{E}_a$ is admissible. Thus, a related admissible set $\mathscr{E}_a$ identifies the relevant arguments that justify the acceptability of a. Let $\mathscr{E}^+ = \{a \in \mathsf{Arg} \mid \mathscr{E} \text{ defeats } a\}$ and $\mathscr{E}^- = \{a \in \mathsf{Arg} \mid a \text{ defeats some } b \in \mathscr{E}\}$. In Example 3, the answer to "why is Smith obliged to prevent crime (b)?" is given by the related admissible set $\mathscr{E}_b = \{b, e\}$ where $\mathscr{E}_b^- = \{d\}$ and $\{d\}^- \cap \mathscr{E}_b = \{e\}$ explain that the only counterargument to b is d which is defeated by e expressing that the norm $(\top, r)$ used in d is inapplicable given the reasons $(\top, p)$ and $\neg(r \wedge p)^c$ offered in e. Hence, using only undercuts enables a more refined analysis of the *relevant* norms explaining the (non-)acceptability of certain arguments and obligations. The DAC approach is therefore more precise compared to using maximal consistent families of norms in traditional I/O.

6. Metatheory: Soundness and Completeness

We demonstrate two soundness and completeness results: First, we prove adequacy between I/O proof systems and DAC (Theorem 1). Second, we prove adequacy between constrained I/O logics and DAC-based argumentation frameworks (Theorem 2). We provide explicit proofs of the main results. Table 1 lists several technical lemmas whose proofs are omitted: Lemma 1 follows by the compactness of L, while Lemmas 2 to 5 are proven by a straightforward induction on the length of the derivation.

We first show adequacy between deriv and DAC. Both systems are modular and correspondence between the rules of these systems is defined in Table 2. In referring to $\mathsf{deriv}_{\mathscr{R}\mathsf{L}}$ and $\mathsf{DAC}_{\mathscr{S}}$ we assume this correspondence. We state the two directions of Theorem 1 separately, we prove Lemma 7, and omit the similar proof of Lemma 6.

Lemma 6. *Let* $\Theta \subseteq \mathscr{L}^n$, *If* $\vdash_{\mathscr{S}} \Delta^f, \Theta \Rightarrow \varphi^o$, *then* $\varphi \in \mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\Delta, \Theta)$.

Lemma 7. *If* $(\varphi, \psi) \in \mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\Theta)$, *then* $\vdash_{\mathscr{S}} \varphi^f, \Theta \Rightarrow \psi^o$.

Table 2. Correspondence between $\mathsf{deriv}_{\mathcal{R},\mathsf{L}}$ rules and $\mathsf{DAC}_{\mathcal{S}}$ rules with the underlying logic L. For instance, $\{\text{ID}, \text{OR}\} \subseteq \mathcal{R}$ iff $\{\textbf{TP}, \textbf{L-OR}\} \subseteq \mathcal{S}$. The first column represents the minimal sets the systems must contain.

Rules of $\mathsf{deriv}_{\mathcal{R},\mathsf{L}}$	$\{$WO, AND, SI$\}$	T	ID	CT	OR
Rules of $\mathsf{DAC}_{\mathcal{S}}$	$\{$**Ax**, **Detach**, **R-C**, **R-N**, **Cut**$\}$	**Taut**	**TP**	**L-CT**	**L-OR**

Proof. By induction on the length of the $\mathsf{deriv}_{\mathcal{R},\mathsf{L}}$-derivation of (φ, ψ). *Base case.* Case $\{(\varphi, \psi)\} = \Theta$. By **Detach**, $\vdash_{\mathcal{S}} \varphi^f, (\varphi, \psi) \Rightarrow \psi^o$. Case $(\top, \top)$ is derived by T with $\Theta = \emptyset$. By **Detach**, $\vdash_{\mathcal{S}} \top^f, (\top, \top) \Rightarrow \top^o$ and by **Taut**, $\vdash_{\mathcal{S}} \Rightarrow (\top, \top)$. By **Cut**, $\vdash_{\mathcal{S}} \top^f \Rightarrow \top^o$. Case (φ, φ) is derived by ID with $\Theta = \emptyset$. By **TP**, $\varphi^f \Rightarrow \varphi^o$.

Inductive step. To illustrate, we consider the case of CT. The other cases are similar or straightforward. Suppose that (φ, ψ) is derived from $(\varphi, \sigma) \in \mathsf{deriv}_{\mathcal{R},\mathsf{L}}(\Theta_1)$ and $(\varphi \wedge \sigma, \psi) \in \mathsf{deriv}_{\mathcal{R},\mathsf{L}}(\Theta_2)$ by CT, where $\Theta = \Theta_1 \cup \Theta_2$. By the IH, $\vdash_{\mathcal{S}} \varphi^f, \Theta_1 \Rightarrow \sigma^o$ and $\vdash_{\mathcal{S}} (\varphi \wedge \sigma)^f, \Theta_2 \Rightarrow \psi^o$. By R$\wedge$2, $\varphi, \sigma \vdash \varphi \wedge \sigma$. By **Ax**, $\vdash_{\mathcal{S}} \varphi^f, \sigma^f \Rightarrow (\varphi \wedge \sigma)^f$ and by **Cut**, $\varphi^f, \sigma^f, \Theta_2 \Rightarrow \psi^o$. Then, if $\emptyset \neq \Theta_2$, by **L-CT**, $\vdash_{\mathcal{S}} \varphi^f, \sigma^o, \Theta_2 \Rightarrow \psi^o$ and by **Cut**, $\vdash_{\mathcal{S}} \varphi^f, \Theta \Rightarrow \psi^o$. Else, $\Theta_2 = \emptyset$ (and hence $\Theta = \Theta_1$). We consider: (i) $\textbf{TP} \in \mathcal{S}$ and (ii) $\textbf{TP} \notin \mathcal{S}$. Ad (i). By Lemma 3.1, $\varphi, \sigma \vdash \psi$ and by **Ax**, $\vdash_{\mathcal{S}} \varphi^o, \sigma^o \Rightarrow \psi^o$. By **TP**, $\vdash_{\mathcal{S}} \varphi^f \Rightarrow \varphi^o$ and by twice **Cut**, $\vdash_{\mathcal{S}} \varphi^f, \Theta \Rightarrow \psi^o$. Ad (ii). By Lemma 3.2, $\vdash \psi$ and so $\sigma \vdash \psi$. By **Ax**, $\vdash_{\mathcal{S}} \sigma^o \Rightarrow \psi^o$. By **Cut**, $\vdash_{\mathcal{S}} \varphi^f, \Theta \Rightarrow \psi^o$. □

Theorem 1. *Let $\Delta \subseteq \mathcal{L}$, $\psi \in \mathcal{L}$, and $\Theta \subseteq \mathcal{L}^n$. Then, $\vdash_{\mathcal{S}} \Delta^f, \Theta \Rightarrow \psi^o$ iff $\psi \in \mathsf{deriv}_{\mathcal{R},\mathsf{L}}(\Delta, \Theta)$.*

Proof. ($\Rightarrow$) This is Lemma 6. ($\Leftarrow$) Suppose $\psi \in \mathsf{deriv}_{\mathcal{R},\mathsf{L}}(\Delta, \Theta)$. So, $(\bigwedge \Delta, \psi) \in \mathsf{deriv}_{\mathcal{R},\mathsf{L}}(\Theta)$. By Lemma 7, $\vdash_{\mathcal{S}} (\bigwedge \Delta)^f, \Theta \Rightarrow \psi^o$. Since $\Delta \vdash \bigwedge \Delta$, $\vdash_{\mathcal{S}} \Delta^f \Rightarrow (\bigwedge \Delta)^f$ by **Ax**. By **Cut**, $\vdash_{\mathcal{S}} \Delta^f, \Theta \Rightarrow \psi^o$. □

We now prove our second adequacy result concerning constrained I/O logics and DAC-instantiated argumentation frameworks.

Theorem 2. *Let $\mathcal{K} = \langle \mathcal{F}, \mathcal{N}, \mathcal{C} \rangle$ be a knowledge base. Let $\mathcal{R}$ be a set of* deriv*-rules and $\mathcal{S}$ the set of corresponding* DAC*-rules (Table 2). Let $\mathrm{AF} = \mathrm{AF}_{\mathcal{S}}(\mathcal{K}) = \langle \mathsf{Arg}, \mathsf{Att} \rangle$.*

1. *If $\mathcal{N}' \in \mathsf{maxfam}_{\mathcal{R},\mathsf{L}}(\mathcal{K})$ then $\mathsf{Arg}(\mathcal{F}^f \cup \mathcal{N}' \cup \mathcal{C}^c)$ is stable in AF.*
2. *If $\mathcal{A}$ is stable in AF then there is a $\mathcal{N}' \subseteq \mathcal{N}$ such that $\mathcal{N}' \in \mathsf{maxfam}_{\mathcal{R},\mathsf{L}}(\mathcal{K})$ for which $\mathcal{A} = \mathsf{Arg}(\mathcal{F}^f \cup \mathcal{N}' \cup \mathcal{C}^c)$.*

Proof. (1) Let $\mathcal{N}' \in \mathsf{maxfam}_{\mathcal{R},\mathsf{L}}(\mathcal{K})$ and $\mathcal{A} = \mathsf{Arg}(\mathcal{F}^f \cup \mathcal{N}' \cup \mathcal{C}^c)$. For conflict-freeness assume towards a contradiction that there are $a = \Delta^f, \Theta, \Gamma^c \Rightarrow \neg(\varphi, \psi) \in \mathcal{A}$ (where $\Theta \subseteq \mathcal{N}'$) and $b = \Omega, (\varphi, \psi) \Rightarrow \Sigma \in \mathcal{A}$ such that a attacks b. By Lemma 2 and since $(\varphi, \psi) \in \mathcal{N}'$, we have, $\Delta^f, \Theta, \Gamma^c, (\varphi, \psi) \Rightarrow \in \mathcal{A}$. By Lemma 5, $\Delta^f, \Theta, (\varphi, \psi) \Rightarrow \sigma^o \in \mathcal{A}$ for some σ for which $\sigma \vdash \neg \bigwedge \Gamma$. By Theorem 1, $\sigma \in \mathsf{deriv}_{\mathcal{R},\mathsf{L}}(\Delta, \Theta \cup \{(\varphi, \psi)\})$, which contradicts the $\mathcal{C}$-consistency of $\mathcal{N}'$.

For $\mathcal{A}$ defeats all $b \in \mathsf{Arg} \setminus \mathcal{A}$ let $a = \Delta_1^f, \Theta_1, \Gamma_1^c \Rightarrow \Sigma \in \mathsf{Arg} \setminus \mathcal{A}$, where $\Theta_1 \subseteq \mathcal{L}^n$. So, there is a $(\varphi, \psi) \in \Theta_1 \setminus \mathcal{N}'$. By the maximal consistency of $\mathcal{N}'$, $\mathcal{N}' \cup \{(\varphi, \psi)\}$ is inconsistent with $\mathcal{C}$. So, there is a $\theta \in \mathsf{deriv}_{\mathcal{R},\mathsf{L}}(\Delta_2, \Theta_2)$ for some $\Delta_2 \subseteq \mathcal{F}$ and $\Theta_2 \subseteq \mathcal{N}' \cup \{(\varphi, \psi)\}$ such that $\mathcal{C} \vdash \neg\theta$. By Theorem 1, $\Delta_2, \Theta_2 \Rightarrow \theta^o \in \mathsf{Arg}$. Note that $(\varphi, \psi) \in \Theta_2$ since otherwise $\Theta_2 \subseteq \mathcal{N}'$ in contradiction to the consistency of $\mathcal{N}'$. By **R-C** and **R-N**, $\vdash_{\mathcal{S}} \Delta_2, \Theta_2 \setminus \{(\varphi, \psi)\}, (\neg\theta)^c \Rightarrow \neg(\varphi, \psi)$. By Lemma 1, $b = \Delta_2, \Theta_2 \setminus \{(\varphi, \psi)\}, \Gamma_2^c \Rightarrow \neg(\varphi, \psi) \in \mathsf{Arg}$ for some $\Gamma_2 \subseteq \mathcal{C}$ for which $\Gamma_2 \vdash \neg\theta$. Note, $b \in \mathcal{A}$ and b attacks a.

(2) Let $\mathscr{A}$ be a stable extension of $AF(\mathscr{K})$. Let $\mathscr{N}' = \{(\varphi,\psi) \in \mathscr{N} \mid \neg\exists a = \Delta \Rightarrow \neg(\varphi,\psi) \in \mathscr{A}\}$. We first show that $\mathscr{A} = \mathsf{Arg}(\mathscr{F}^f \cup \mathscr{N}' \cup \mathscr{C}^c)$:

"($\supseteq$)" Let $a \in \mathsf{Arg}(\mathscr{F}^f \cup \mathscr{N}' \cup \mathscr{C}^c)$. By the definition of $\mathscr{N}'$ there is no $b \in \mathscr{A}$ that attacks a and since $a \in \mathsf{Arg}$ and by the stability of $\mathscr{A}$, $a \in \mathscr{A}$. "($\subseteq$)" Let $a \in \mathsf{Arg} \setminus \mathsf{Arg}(\mathscr{F}^f \cup \mathscr{N}' \cup \mathscr{C}^c)$ with $a = \Delta \Rightarrow \Gamma$. So, there is a $(\varphi,\psi) \in \Delta$ for which there is a $b \in \mathscr{A}$ with $b = \Theta \Rightarrow \neg(\varphi,\psi)$. So b attacks a and by the stability of $\mathscr{A}$, $a \notin \mathscr{A}$.

We now show that $\mathscr{N}' \in \mathsf{maxfam}_{\mathscr{R},\mathsf{L}}(\mathscr{K})$. Assume for a contradiction that $\mathscr{N}'$ is inconsistent with $\mathscr{C}$. So, there is a $\theta \in \mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\Delta,\Theta)$ for some $\Delta \subseteq \mathscr{F}$ and $\Theta \subseteq \mathscr{N}'$ for which $\mathscr{C} \vdash \neg\theta$. By Theorem 1, $a = \Delta^f, \Theta \Rightarrow \theta^o \in \mathscr{A}$. Assume first that $\Theta = \emptyset$.

If $\mathbf{TP} \notin \mathscr{S}$, by Lemma 3.3, $\vdash \theta$ and thus $\mathscr{C}$ is inconsistent which is a contradiction. Thus, $\Theta \neq \emptyset$. If $\mathbf{TP} \in \mathscr{S}$ then, by Lemma 3.2, $\Delta \vdash \theta$. But then $\mathscr{F} \cup \mathscr{C}$ is inconsistent, a contradiction and so $\Theta \neq \emptyset$. So, in both cases $\Theta \neq \emptyset$.

Let $(\varphi,\psi) \in \Theta$. By **R-N** and **R-C**, $\vdash_{\mathscr{S}} \Delta, \Theta \setminus \{(\varphi,\psi)\}, (\neg\theta)^c \Rightarrow \neg(\varphi,\psi)$. By Lemma 1, there is a $\Gamma \subseteq \mathscr{C}$ for which $b = \Delta, \Theta \setminus \{(\varphi,\psi)\}, \Gamma^c \Rightarrow \neg(\varphi,\psi) \in \mathscr{A}$. Since b attacks a, this contradicts conflict-freeness of $\mathscr{A}$, which shows $\mathscr{N}'$ is consistent with $\mathscr{C}$.

Assume for a contradiction that there is a $(\varphi,\psi) \in \mathscr{N} \setminus \mathscr{N}'$ such that $\mathscr{N}' \cup \{(\varphi,\psi)\}$ is consistent with $\mathscr{C}$ (i.e., $\mathscr{N}'$ is not maximal). By the definition of $\mathscr{N}'$, there is a $b = \Delta^f, \Theta, \Gamma^c \Rightarrow \neg(\varphi,\psi) \in \mathscr{A}$. By Lemma 2, $\vdash_{\mathscr{S}} \Delta^f, \Theta, (\varphi,\psi), \Gamma^c \Rightarrow$. By Lemma 5, $\vdash_{\mathscr{S}} \Delta^f, \Theta, (\varphi,\psi) \Rightarrow \sigma^o$ such that $\sigma \vdash \neg\bigwedge\Gamma$. By Theorem 1, $\sigma \in \mathsf{deriv}_{\mathscr{R},\mathsf{L}}(\Delta, \Theta \cup \{(\varphi,\psi)\})$ which shows that $\mathscr{N}' \cup \{(\varphi,\psi)\}$ is inconsistent with $\mathscr{C}$ (note that $\Gamma \subseteq \mathscr{C}$). This completes our proof since it shows that $\mathscr{N}' \in \mathsf{maxfam}_{\mathscr{R},\mathsf{L}}(\mathscr{K})$. □

Corollary 1. *Let $\mathscr{K}$ be a knowledge base, $\mathscr{R}$ a set of* deriv*-rules, and $\mathscr{S}$ a set of corresponding* DXC*-rules (Table 2). For $i \in \{s,c\}$, $AF_{\mathscr{S}}(\mathscr{K}) \mid\!\sim^{i}_{stable} \varphi$ iff $\mathscr{K} \mid\!\sim^{i}_{\mathscr{R},\mathsf{L}} \varphi$.*

7. Related Work and Conclusion

In [14], a sequent-style system for monotonic I/O logics without constraints is presented. It utilizes a correspondence between I/O and conditional logics. In [15] proof systems for constrained I/O logic are developed, where modalities for 'input' and 'output' allow for meta-reasoning in the object language. DAC uses labels instead of modalities and additionally allows for meta-reasoning about the (in)applicability of norms. In [9], sequent argumentation is used for defeasible reasoning with deontic logic. Norms are modelled with material implications which allows for less fine-tuning of norms than in DAC.

In [5,6,18] argumentative characterizations of normative systems employing priority orderings are studied. Their language is restricted to literals only, whereas our approach adopts a full propositional language. In [6,18] arguments consist only of (sets of) norms. In future work, we aim to incorporate priority and preference reasoning in the more transparent context of DAC. Moreover, the I/O formalism has other applications including reasoning with consistency checks, permissions, and constitutive norms [8,17]. In particular, we aim to exploit the internalization of meta-reasoning in DAC to characterize various types of permission [17], for instance, negative permissions as defined in terms of the absence of applicable norms to the contrary.

An alternative approach to model reasoning with norms is to instantiate ASPIC$^+$ [2] with conditionals representing norms and a defeasible modus ponens rule. This approach leads to a "greedier" style of reasoning than our approach. Consider $\mathscr{F} = \emptyset$ and $\mathscr{N} = \{(\top,p),(p,q),(\top,\neg q)\}$. An ASPIC$^+$-based approach yields the obligation to p

with stable semantics since the argument for p from $(\top, p)$ is unchallenged. In contrast, our approach generates the argument $(\top, \neg q), (p, q) \Rightarrow \neg(\top, p)$ concluding the inapplicability of $(\top, p)$. The latter is in line with the I/O approach to normative reasoning.

We illustrated our approach with the notion of related admissibility [16]. For future work, we will investigate other argumentative approaches to explanation and how these can be used in the context of DAC, e.g., explicit reasoning about the inapplicability of norms in DAC can be harnessed to explain the non-acceptability of arguments [19].

Last, explanations typically occur in dialogues, through an exchange of reasons, questions, and arguments [20]. Consequently, explanations are often tailored to the background of the explainee. We will adopt our approach to dialogue models in future work.

In conclusion, in normative reasoning contexts one is not just interested in whether a specific obligation holds, but also in why it holds despite other norms to the contrary. To address this challenge, we developed Deontic Argument Calculi (DAC) which are rule-based proof systems that use labels to facilitate transparency and incorporate metanormative reasoning with norms into the object language.

References

[1] Gabbay D, Horty J, Parent X, van der Meyden R, van der Torre L. Handbook of Deontic Logic and Normative Systems. vol. 1. United Kingdom: College Publications; 2013.

[2] Gabbay D, Giacomin M, Simari GR, Thimm M. Handbook of Formal Argumentation. vol. 2. United Kingdom: College Publications; 2021.

[3] Makinson D, van der Torre L. Constraints for Input/Output Logics. Journal of Philosophical Logic. 2001;30(2):155-85.

[4] Beirlaen M, Straßer C, Heyninck J. Structured argumentation with prioritized conditional obligations and permissions. Journal of Logic and Computation. 2018;29(2):187-214.

[5] Governatori G, Rotolo A, Riveret R. A deontic argumentation framework based on deontic defeasible logic. In: Poceedings of PRIMA 2018. Springer; 2018. p. 484-92.

[6] Liao B, Oren N, van der Torre L, Villata S. Prioritized norms in formal argumentation. Journal of Logic and Computation. 2018;29(2):215-40.

[7] Peirera C, Tettamanzi AG, Villata S, Liao B, Malerba A, Rotolo A, et al. Handling norms in multi-agent system by means of formal argumentation. IfCoLog. 2017;4(9):1-35.

[8] Pigozzi G, van der Torre L. Arguing about constitutive and regulative norms. Journal of Applied Non-Classical Logics. 2018;28(2-3):189-217.

[9] Straßer C, Arieli O. Normative reasoning by sequent-based argumentation. Journal of Logic and Computation. 2015 07;29(3):387-415.

[10] Parent X, van der Torre L. Introduction to deontic logic and normative systems. College Publications; 2018.

[11] Brunero J. Reasons, Evidence, and Explanations. Oxford Handbooks Online. 2018 Jul.

[12] Nair S, Horty J. The Logic of Reasons. Oxford Handbooks Online. 2018 Jul.

[13] Horty JF. Reasons as defaults. Oxford University Press, USA; 2012.

[14] Lellmann B. From Input/Output Logics to Conditional Logics via Sequents – with Provers. In: Automated Reasoning with Analytic Tableaux and Related Methods. Cham: Springer; 2021. p. 147-64.

[15] Straßer C, Beirlaen M, Van De Putte F. Adaptive logic characterizations of input/output logic. Studia Logica. 2016;104(5):869-916.

[16] Fan X, Toni F. On computing explanations in argumentation. In: AAAI; 2015. p. 1496-502.

[17] Tosatto SC, Boella G, van der Torre L, Villata S. Abstract normative systems: Semantics and proof theory. In: Proceedings of KR12; 2012. .

[18] Straßer C, Pardo P. Prioritized Defaults and Formal Argumentation. In: Proceedings of 14th International Conference of Deontic Logic and Normative Systems. College Publications; 2021. p. 427-46.

[19] Borg A, Bex F. A Basic Framework for Explanations in Argumentation. IEEE Int Systems. 2021:25-35.

[20] Walton D. A dialogue system specification for explanation. Synthese. 2010 Apr;182(3):349–374.

This page intentionally left blank

Demo Papers

This page intentionally left blank

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220166

An Argumentative Explanation of Machine Learning Outcomes[1]

Stefano BISTARELLI [a,2], Alessio MANCINELLI [a], Francesco SANTINI [a,2], and Carlo TATICCHI [a,2,3]

[a] *Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Italy*

Keywords. Computational argumentation, explainability, machine learning.

The black box model used in Machine Learning is considered one of the major problems in the application of Artificial Intelligence techniques [1] as it makes machine decisions non-transparent and often incomprehensible even to experts or developers themselves. In this paper, we provide an argumentative interpretation of both the training process and the results predicted. The goal is to build a Bipolar Argumentation Framework (BAF) [2] showing the dialectical reasoning behind the assignment of a certain class to a given record. Since we make assumptions neither on the dataset nor on the algorithm used, the presented procedure can be applied to existing models without the need for further adjustments. To illustrate our proposal, we use the *Titanic* dataset from `www.kaggle.com`, which contains records relating to people involved in the Titanic disaster. We consider three categorical features, namely *Survived* (the class to predict, with value 1 if the person survived or 0, otherwise), *Pclass* (ticket class among 1, 2 and 3) and *sex* (0 for woman and 1 for man), and two numerical features: *Age* (passenger age, ranging from 0.17 to 76) and *Fare* (passenger fare with values from 0 to 512). In the following, we describe the step our procedure goes through in order to find an explanation for the class *Survived=1*.

Dataset Clustering. In the first step, starting from the input dataset, we create a new clustered dataset in which numerical features are split into categories that group ranges of values to obtain a more appropriate and concise explanation.

BAF Generation. Then we build a BAF based on the correlation matrix computed among the features. By construction, the obtained BAF only has symmetric relations.

Breaking Complete Symmetry. Given the correlation matrix, we apply a procedure that removes symmetric edges from the BAF to establish a causal relationship between features. In particular, we use the conditional probability [3] computed for arguments which attack/support each other. We choose the minimum values possible that keep the graph connected.

Computing Extensions. To identify the set of arguments which are more likely to be accepted, we compute the semi-stable extensions [4] of the previously obtained

[1] This work has been partially supported by: GNCS-INdAM, CUP E55F22000270001; Project RACRA - funded by Ricerca di Base 2018-2019, Univeristy of Perugia; Project BLOCKCHAIN4FOODCHAIN: funded by Ricerca di Base 2020, Univeristy of Perugia; Project DopUP - REGIONE UMBRIA PSR 2014-2020.

[2] The author is a member of the INdAM Research group GNCS and of Consorzio CINI.

[3] Corresponding Author: Carlo Taticchi, Università degli Studi di Perugia; E-mail: carlo.taticchi@unipg.it.

framework and then we use the tool described in [5] to find, for each of them, its probability of being admissible. In our example, we obtain the following extension, which is semi-stable and also admissible with probability 1 (the highest possible).

$$Age<0.96,\ Fare\geq 10.48,\ Pclass=1,\ Sex=0,\ \mathbf{Survived=1}$$

Building the Explanation Tree. Finally, starting from the arguments of the selected extension, we produce the explanation tree of Figure 1, where accepted arguments are highlighted in green and rejected ones in red.

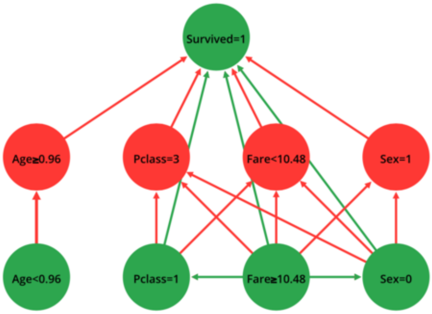


Figure 1. An explanation tree for the class *Survived=1* of the Titanic dataset.

Looking at the obtained explanation we can conclude, for instance, that the person in question survived because "she is a woman (*Sex=0*), with a paid ticket (*Fare≥10.48*) and travelling first class (*Pclass=1*)". Indeed, arguments representing those features in Figure 1 attack other arguments that are against the assignment of the class *Survived=1*, standing in turn for being male (*Sex=1*) and having a third-class ticket (*Pclass=3*) with a low fare (*Fare<10.48*).

In future work, alternative techniques could be applied to break the symmetry of the graph to obtain a causal relationship between arguments. Furthermore, particular attention could be paid to simplifying the explanation provided, including notions of symmetry and interchangeability between arguments, as well as applying Natural Language Processing to provide a further textual explanation.

References

[1] von Eschenbach WJ. Transparency and the Black Box Problem: Why We Do Not Trust AI. Philosophy & Technology. 2021 Dec;34(4):1607-22.

[2] Amgoud L, Cayrol C, Lagasquie-Schiex M, Livet P. On bipolarity in argumentation frameworks. Int J Intell Syst. 2008;23(10):1062-93.

[3] Casella G, Berger RL. Statistical inference. vol. 2. Duxbury Pacific Grove, CA; 2002.

[4] Baroni P, Caminada M, Giacomin M. An introduction to argumentation semantics. Knowl Eng Rev. 2011;26(4):365-410.

[5] Bistarelli S, Mantadelis T, Santini F, Taticchi C. Using MetaProbLog and ConArg to compute Probabilistic Argumentation Frameworks. In: Proceedings of the 2nd Workshop on Advances In Argumentation In Artificial Intelligence. vol. 2296 of CEUR Workshop Proceedings. CEUR-WS.org; 2018. p. 6-10.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220167

PyArg for Solving and Explaining Argumentation in Python: Demonstration

AnneMarie BORG [a], Daphne ODEKERKEN [a,b,1]
[a] *Department of Information and Computing Sciences, Utrecht University*
[b] *National Police Lab AI, Netherlands Police*
ORCiD ID: AnneMarie Borg https://orcid.org/0000-0002-7204-6046 , Daphne Odekerken https://orcid.org/0000-0003-0285-0706

Abstract. We introduce PyArg, a Python-based solver and explainer for both abstract argumentation and ASPIC$^+$. A large variety of extension-based semantics allows for flexible evaluation and several explanation functions are available.

Keywords. Abstract Argumentation, Structured Argumentation, Explainable Artificial Intelligence, Python

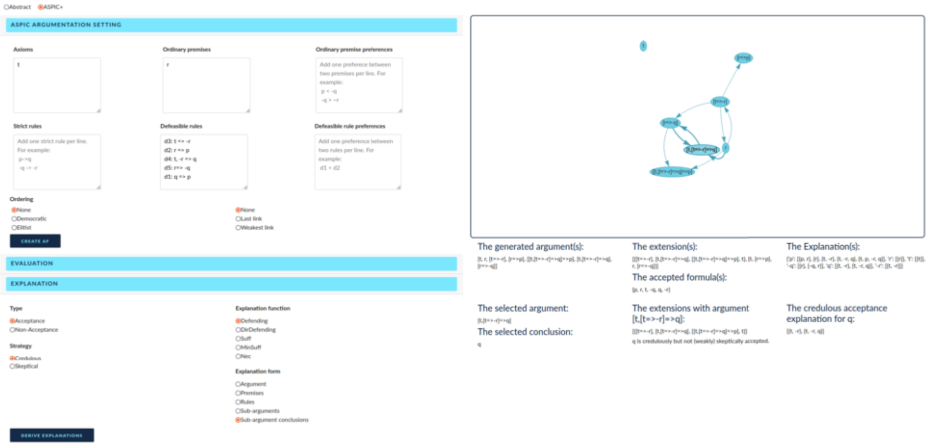
Introduction. Deriving extensions and conclusions from argumentation settings is an essential part of computational argumentation. Moreover, in recent years the interest in argumentation-based explainable artificial intelligence has increased considerably [1]. Since the derivation of conclusions and explanations tends to become intractable when the number of arguments and attacks (in the abstract setting [2]) or the size of the knowledge base and the set of rules (in the structured setting, e.g., [3]) increases, it is useful to have a computational tool that does this for us. To this end, we introduce PyArg, which, in addition to being a solver, can also derive explanations.

The Demonstration. We introduce PyArg [4], a solver designed for researchers and students who are used to work with Python. The package provides various implementations of formalisms and algorithms in both abstract argumentation and ASPIC$^+$ and comes equipped with an integrated, interactive visualization.

- Selection between abstract argumentation [2] and ASPIC$^+$ [3]. In the abstract setting, users can provide arguments and the attacks between them, in the ASPIC$^+$ setting users can provide axioms, ordinary premises with their preferences, strict rules, defeasible rules with their preferences and a choice in how to derive an ordering from these preferences.
- Evaluation based on a large variety of extension-based semantics [5]. The admissible, complete, grounded, preferred, ideal, stable, semi-stable and eager semantics are available as well as a credulous and skeptical strategy.
- Explanations for (non-)accepted arguments and formulas (in the case of an ASPIC$^+$-setting), based on the explanations from [6,7]. There are functions based on the notion of defense as well as based on necessity and sufficiency.

[1] Author order alphabetical.

PyArg, the code and a link to the browser app, is available open source on https://git.science.uu.nl/D.Odekerken/py_arg. We hope that it will turn into a community project where PyArg becomes a complete solver for many argumentation formalisms, which can be used for teaching and research purposes and that is easily extendable for anyone interested to implement their own ideas.



○Abstract ●ASPIC+

ASPIC ARGUMENTATION SETTING

Axioms: t

Ordinary premises: r

Ordinary premise preferences: Add one preferece between two premises per line. For example: p < -q -q > ~r

Strict rules: Add one strict rule per line. For example: p->q -q -> -r

Defeasible rules:
d3: t => -r
d2: r => p
d4: t, -r => q
d5: r=> -q
d1: q => p

Defeasible rule preferences: Add one preference between two rules per line. For example: d1 < d2

Ordering
●None
○Democratic
○Elitist

●None
○Last link
○Weakest link

EVALUATION

EXPLANATION

Type
●Acceptance
○Non-Acceptance

Strategy
●Credulous
○Skeptical

Explanation function
●Defending
○DirDefending
○Suff
○MinSuff
○Nec

Explanation form
○Argument
○Premises
○Rules
○Sub-arguments
●Sub-argument conclusions

The generated argument(s): [t, r, [t=>-r], [r=>p], [[t,[t=>-r]=>q]=>p], [t,[t=>-r]=>q], [r=>-q]]

The extension(s): [[[t=>-r], [t,[t=>-r]=>q], [[t,[t=>-r]=>q]=>p], t], [t, [r=>p], r, [r=>-q]]]

The accepted formula(s): [p, r, t, -q, q, -r]

The Explanation(s): {'p': [[p, r], [r], [t, -r], [t, -r, q], [t, p, -r, q]], 'r': [[r]], 't': [[t]], '-q': [[r], [-q, r]], 'q': [[t, -r], [t, -r, q]], '-r': [[t, -r]]}

The selected argument: [t,[t=>-r]=>q]

The selected conclusion: q

The extensions with argument [t,[t=>-r]=>q]: [[[t=>-r], [t,[t=>-r]=>q], [[t,[t=>-r]=>q]=>p], t]]

q is credulously but not (weakly) skeptically accepted.

The credulous acceptance explanation for q: [[t, -r], [t, -r, q]]

Figure 1. Screenshot of PyArg, in the ASPIC$^+$ setting, based on [6, Example 3].

Future Work. We intend to extend PyArg by implementing additional argumentation formalisms, semantics and explanation functions as well as by introducing dynamic settings. In particular, we will implement algorithms for stability and relevance for incomplete argumentation frameworks in an upcoming release [8].

References

[1] Čyras K, Rago A, Albini E, Baroni P, Toni F. Argumentative XAI: A Survey. In: Proceedings of IJCAI'21. ijcai.org; 2021. p. 4392-9. doi:10.24963/ijcai.2021/600.

[2] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77(2):321-57. doi:10.1016/0004-3702(94)00041-X.

[3] Prakken H. An abstract framework for argumentation with structured arguments. Argument & Computation. 2010;1(2):93-124. doi:10.1080/19462160903564592.

[4] Odekerken D, Borg A. PyArg; 2022. Available from: https://git.science.uu.nl/D.Odekerken/py_arg.

[5] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: Handbook of Formal Argumentation. College Publications; 2018. p. 159-236.

[6] Borg A, Bex F. A Basic Framework for Explanations in Argumentation. IEEE Intelligent Systems. 2021;36(2):25-35. doi:doi: 10.1109/MIS.2021.3053102.

[7] Borg A, Bex F. Necessary and Sufficient Explanations for Argumentation-Based Conclusions. In: Proceedings of ECSQARU'21. Springer; 2021. p. 45-58. doi:10.1007/978-3-030-86772-0_4.

[8] Odekerken D, Borg A, Bex F. Stability and Relevance in Incomplete Argumentation Frameworks. In: Proceedings of COMMA'22; 2022. Forthcoming.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220168

Providing Explanations via the EQR Argument Scheme

Federico CASTAGNA [a,1], Simon PARSONS [a], Isabel SASSOON [b] and Elizabeth I. SKLAR [c]

[a] *School of Computer Science, University of Lincoln*
[b] *Department of Computer Science, Brunel University London*
[c] *Lincoln Institute for Agri-Food Technology, University of Lincoln*

ORCiD ID: Federico Castagna https://orcid.org/0000-0002-5142-4386, Simon Parsons https://orcid.org/0000-0002-8425-9065, Isabel Sassoon https://orcid.org/0000-0002-8685-1054, Elizabeth I. Sklar https://orcid.org/0000-0002-6383-9407

Abstract. This demo paper outlines the EQR argument scheme (AS) structure and deploys its instantiations to convey explanations using a chatbot.

Keywords. argument schemes, chatbot, explanations, decision-support systems

Devised as a pattern of **E**xplanation-**Q**uestion-**R**esponse interactions between agents, the EQR scheme draws from the *AS for Practical Reasoning* [1] and the *Expert Opinion* [2] schemes in order to formalise the consequences entailed by following the assertion of an expert opinion. A reference to such authority provides the rationale that justifies the conclusion of the argument, also leaving chances of inquiry for more detailed explanations.

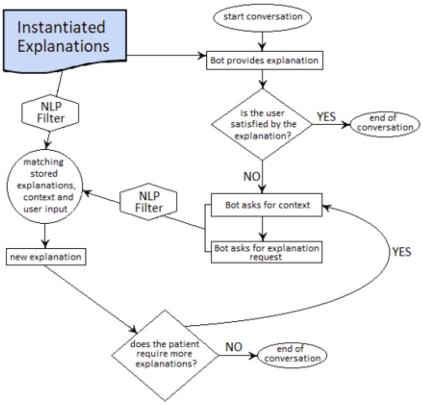
EQR Scheme

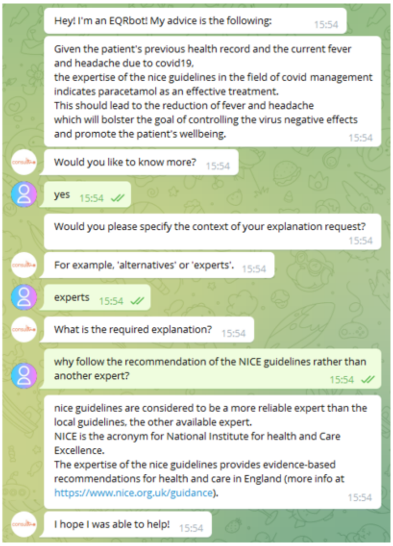
Premise : In the current state R
Premise : asserting α (from an expert E in a field F)
Premise : will result in a new state S
Premise : which will make proposition A true (alternatively, false)
Premise : which will promote some value v

Conclusion : Following the opinion α should make proposition A true (false)

CONSULT[2] is a novel data-driven mobile decision support system (DSS) designed to help patients with chronic conditions self-manage their treatment plans [3]. Such a DSS can deliver to the user more exhaustive information and more detailed answers to follow-on questions by employing the EQR scheme through a chatbot.

[1] Corresponding Author: Federico Castagna, *fcastagna@lincoln.ac.uk*.
[2] https : //consultproject.co.uk


Instantiated Explanations
start conversation
Bot provides explanation
NLP Filter
Is the user satisfied by the explanation?
YES
end of conversation
NO
matching stored explanations, context and user input
NLP Filter
Bot asks for context
Bot asks for explanation request
YES
new explanation
does the patient require more explanations?
NO
end of conversation


Hey! I'm an EQRbot! My advice is the following: 15:54

Given the patient's previous health record and the current fever and headache due to covid19,
the expertise of the nice guidelines in the field of covid management indicates paracetamol as an effective treatment.
This should lead to the reduction of fever and headache
which will bolster the goal of controlling the virus negative effects
and promote the patient's wellbeing. 15:54

Would you like to know more? 15:54

yes 15:54

Would you please specify the context of your explanation request? 15:54

For example, 'alternatives' or 'experts'. 15:54

experts 15:54

What is the required explanation? 15:54

why follow the recommendation of the NICE guidelines rather than another expert? 15:54

nice guidelines are considered to be a more reliable expert than the local guidelines, the other available expert.
NICE is the acronym for National Institute for health and Care Excellence.
The expertise of the nice guidelines provides evidence-based recommendations for health and care in England (more info at https://www.nice.org.uk/guidance). 15:54

I hope I was able to help! 15:54

Figure 1. High-level operations (left), and example of explanations performed by the chatbot (right).

EQRbot. The interaction with the patient will be handled by the chatbot[3] which, after providing the initial explanation (i.e., an instantiation of the EQR scheme through the data collected by CONSULT), will ask the patient for feedback. If the user is satisfied, then the conversation will immediately end. Alternatively, the chatbot will demand a brief context along with the actual patient's request. By matching stored explanations, context and user input, the bot will output the additional solicited information (Figure 1, left). Observe that the double query prompted by the bot, along with a general NLP filter, ensures a significant reduction of misunderstandings when providing answers.

Example. Consider a patient suffering from fever and headache due to the Covid-19 virus. These facts, and the treatment recommended by the clinical guidelines of NICE-NG191[4], will be registered and encoded by the CONSULT system, eliciting the instantiation of the EQR scheme (the initial explanation) and of potential additional information (subsequent explanations) that will be conveyed by the EQRbot (Figure 1, right).

Acknowledgements. This research was partially funded by the UK Engineering & Physical Sciences Research Council (EPSRC) under grant #EP/P010105/1.

References

[1] Atkinson K, Bench-Capon T. Practical reasoning as presumptive argumentation using action based alternating transition systems. Artificial Intelligence. 2007;171(10-15):855-74.

[2] Walton D. Appeal to Expert Opinion: Arguments from Authority. Pennsylvania State University Press, University Park, PA, USA. 1997.

[3] Kökciyan N, Chapman M, Balatsoukas P, Sassoon I, Essers K, Ashworth M, et al. A Collaborative Decision Support Tool for Managing Chronic Conditions. Studies in health technology and informatics. 2019;264:644-8.

[3] https://github.com/FCast07/EQRbot

[4] https://www.nice.org.uk/nice-guidance

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220169

EGNN: A Deep Reinforcement Learning Architecture for Enforcement Heuristics

Dennis CRAANDIJK [a,b], Floris BEX [b,c]
[a] *National Police Lab AI, Netherlands Police*
[b] *Department of Information and Computing Sciences, Utrecht University*
[c] *Institute for Law, Technology and Society, Tilburg University*

1. Introduction

An increasing amount of research is being directed towards neuro-symbolic computing, combining learning in neural networks with reasoning and explainability via symbolic representations [4]. One subfield of AI where neuro-symbolic methods are a promising alternative for existing symbolic methods is computational argumentation. Much of the theory of computational argumentation is based on the seminal work by Dung [6], in which he introduces abstract argumentation frameworks (AFs) of arguments and attacks, and several acceptability semantics that define which sets of arguments (*extensions*) can be reasonably accepted. Core computational problems in abstract argumentation are typically solved with handcrafted symbolic methods [1]. However, recently we demonstrated the potential of a deep learning approach by showing that a graph neural network is able to learn to determine almost perfectly which arguments are (part of) an extension [2].

When considering dynamic argumentation - a growing research area where the knowledge about attacks between arguments can be incomplete or evolving - other types of computational problems arise where neuro-sybmolic methods are still unexplored. In [3] we propose our enforcement graph neural network (EGNN), a learning-based approach to the dynamic argumentation problem of enforcement: given sets of arguments that we (do not) want to accept, how to modify the argumentation framework in such a way that these arguments are (not) accepted, while minimizing the number of changes [5]. Here we demonstrate our implementation of an EGNN.

2. Demonstration

When confronted with some problems with a high computational complexity, existing symbolic enforcement solvers exhibit quite a significant drop in runtime performance, limiting their practical applicability. While there is a need for efficient heuristics to address this problem, designing such heuristics takes considerable expert effort and domain knowledge. EGNN is a single architecture that can be trained through deep reinforcement learning to learn enforcement heuristics for all common semantics and enforcement problems, without supervision of an existing solver. EGNN learns a *message passing*

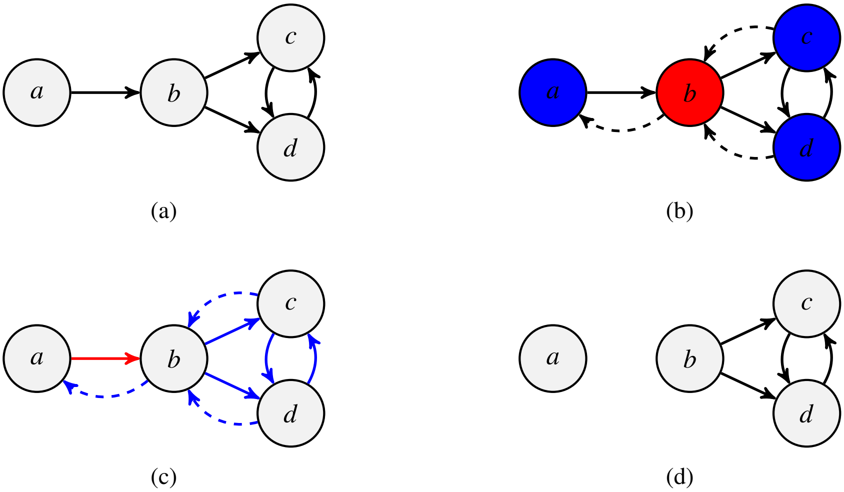


Figure 1. Consider enforcing argument b in the AF $F = (\{a,b,c,d\},\{(a,b),(b,c)(b,d),(c,d),(d,c)\})$. EGNN takes the AF (a), maps it it to a fully connected graph where nodes have a vectorial representation denoting which arguments should be enforced (b). Node vectors are updated through *message passing* and are mapped to an output per edge (c) indicating which edge should be *flipped* (d).

algorithm that predicts which attack relations between arguments should be *flipped* (i.e. added or deleted) in order to enforce the acceptability of (a set of) arguments. Experimental results demonstrate that EGNN can learn near-optimal heuristics for all *extension* and *status* enforcement problems under the most common semantics, and outperforms symbolic solvers with respect to efficiency on enforcement problems that are higher in the complexity hierarchy.

We demonstrate our Python implementation of an EGNN and show: the input, message passing and output steps of the model; the learned heuristics for enforcement problems; how the learned heuristic differs from symbolic algorithms. We do so by graphically demonstrating EGNN's behaviour on an AF (cf. Figure 1).

References

[1] Günther Charwat, Wolfgang Dvorák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation - A survey. *Artificial Intelligence*, 220:28–63, 2015.

[2] Dennis Craandijk and Floris Bex. Deep learning for abstract argumentation semantics. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1667–1673, 2020.

[3] Dennis Craandijk and Floris Bex. Enforcement heuristics for argumentation with deep reinforcement learning. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pages 5573–5581, 2022.

[4] Artur S. d'Avila Garcez, Marco Gori, Luís C. Lamb, Luciano Serafini, Michael Spranger, and Son N. Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *FLAP*, 6(4):611–632, 2019.

[5] Sylvie Doutre and Jean-Guy Mailly. Constraints and changes: A survey of abstract argumentation dynamics. *Argument & Computation*, 9(3):223–248, 2018.

[6] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220170

ADF-BDD: An ADF Solver Based on Binary Decision Diagrams[1]

Stefan ELLMAUTHALER [a], Sarah A. GAGGL [b], Dominik RUSOVAC [b], and Johannes P. WALLNER [c]

[a] *Knowledge-Based Systems Group, cfaed, TU Dresden, Germany*
[b] *Logic Programming and Argumentation Group, TU Dresden, Germany*
[c] *Institute of Software Technology, Graz University of Technology, Austria*

Keywords. abstract dialectical frameworks, binary decision diagrams, rust-lang

Abstract Dialectical Frameworks [1] (ADF) are a generalisation of Dung's Argumentation frameworks [2]. Multiple approaches for reasoning under various semantics have been proposed over the last decade [3,4,5,6]. We present *"**A**bstract **D**ialectical **F**rameworks solved by **B**inary **D**ecision **D**iagrams, developed in **D**res**d**en"* (ADF-BDD)[2], a novel approach that relies on the translation of the acceptance conditions of a given ADF into reduced ordered binary decision diagrams (roBDD) [7]. Our system is based on the consideration that many otherwise hard to decide problems in ADF semantics (e. g., answering SAT-questions) can be solved in polynomial time on roBDDs (see [8] for an in-depth analysis). Our novel approach differs to the currently used systems, like the SAT-based approach K++ADF [5] or the wide spectrum of answer set programming (ASP) focused approaches like the DIAMOND family (e. g., DIAMOND [3] or GODIAMOND [4]) and YADF [6]. ADF-BDD is written in RUST [9] to provide good performance while enforcing a high amount of memory- and type-safety. In addition the rust-compiler produces highly optimised machine code, while keeping the whole tech stack simple.

ADF-BDD accepts the established input format, introduced first in [10]. There statements are unary predicates `s`, defining the labels and the acceptance conditions are binary predicates `ac`, relating the label to a formula. It allows to enumerate the grounded and complete interpretations, and stable models of the given input instance. The set of statements is the shared signature of all acceptance conditions, hence our implementation uses a single structure to store the nodes of all the roBDDs, which represent each acceptance condition. This allows for efficient caching of nodes and to eliminate duplicate node candidates. Another side-effect is that shared sub-BDDs are computed only once. ADF-BDD provides the explained implementation of roBDDs as the representation of the acceptance conditions. As the instantiation of roBDDs is a computational hard task, it is possible to utilise another state-of-the art competitive library called Biodivine/LibBDD[3]. It is part of the Biodivine software in the AEON project [11]. While LibBDD is faster in

[1] This work is partly supported by the BMBF, Grant 01IS20056_NAVAS, by the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), and by the DFG through the Collaborative Research Center, Grant TRR 248 project ID 389792660.

[2] `https://github.com/ellmau/adf-obdd`, version 0.2.4, `https://crates.io/crates/adf_bdd`

[3] `https://crates.io/crates/biodivine-lib-bdd`

the instantiation, it is unfortunately not providing all the used features for the efficient application and backtracking of operations on roBDDs of ADF-BDDs implementation. The user can either use one of the two libraries to handle the representation of roBDDs or a hybrid approach, combining the fast instantiation and the efficient operations.

The grounded interpretation is computed via a deterministic approach computing the least fixed point of the approximate operator for ADFs. For the complete interpretations we have chosen to implement a naive approach which lazily checks all possible three valued interpretations. The stable model computation supports this naive approach of lazily checking all possible two-valued interpretations. Furthermore a simple heuristics-based approach is implemented. It allows to incorporate various easily accessible information about the acceptance condition, provided by the roBDD representation. The heuristics then approximate which of the two truth values one statement can have is less costly and computes its influence to the other statements. For the other value we use a no-good like list of value assertions, to steer the further enumeration of possible two valued models.

The performance of our tool[4] is positioned in between the fastest SAT-based approach and the ASP based approaches. This is achieved although the complete semantics are computed in a naive manner. The use of the heuristics based approach for stable models runs faster and more reliable than the naive implementation. This shows that the representation with roBDDs is a promising approach. Future optimisation, more sophisticated learning algorithms, and better heuristics will reduce the gap to K++ADF further.

We present a library (*"adf_bdd"*) for an easy use of the functionality in other software-products and provide an executable (*"adf-bdd"*) to use the library as a straightforward and simple to use solver.

References

[1] Brewka G, Ellmauthaler S, Strass H, Wallner JP, Woltran S. Abstract Dialectical Frameworks. In: Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018. p. 237-85.

[2] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artif Intell. 1995;77(2):321-58.

[3] Ellmauthaler S, Strass H. The DIAMOND System for Computing with Abstract Dialectical Frameworks. In: Proc. COMMA. vol. 266 of FAIA. IOS Press; 2014. p. 233-40.

[4] Strass H, Ellmauthaler S. GoDIAMOND 0.6.6–ICCMA 2017 System Description. Second International Competition on Computational Models of Argumentation. 2017.

[5] Linsbichler T, Maratea M, Niskanen A, Wallner JP, Woltran S. Advanced algorithms for abstract dialectical frameworks based on complexity analysis of subclasses and SAT solving. Artif Intell. 2022;307:103697.

[6] Brewka G, Diller M, Heissenberger G, Linsbichler T, Woltran S. Solving Advanced Argumentation Problems with Answer Set Programming. TPLP. 2020;20(3):391-431.

[7] Bryant RE. Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams. ACM Comput Surv. 1992;24(3):293-318.

[8] Darwiche A, Marquis P. A Knowledge Compilation Map. J Artif Intell Res. 2002;17:229-64.

[9] Matsakis ND, II FSK. The rust language. In: Proc. HILT. ACM; 2014. p. 103-4.

[10] Ellmauthaler S, Wallner JP. Evaluating Abstract Dialectical Frameworks with ASP. In: Verheij B, Szeider S, Woltran S, editors. Proc. COMMA. vol. 245. IOS Press; 2012. p. 505-6.

[11] Beneš N, Brim L, Kadlecaj J, Pastva S, Šafránek D. AEON: Attractor Bifurcation Analysis of Parametrised Boolean Networks. In: Proc. CAV. Springer; 2020. p. 569-81.

[4] Dataset and runtimes can be found at https://doi.org/10.5281/zenodo.6498235

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220171

Annotating Very Large Arguments

Kamila GORSKA [a,1], Wassiliki SISKOU [b] and Chris REED [a]
[a] *Centre for Argument Technology, University of Dundee, UK*
[b] *University of Konstanz, Germany*

Abstract. We present a method of annotating very large arguments, with the use of IMC-Tool. IMC-Tool aids in creating long-distance argument structure relations, by providing a simple annotation tool, and integration software to synthesise argument annotations at scale.

Keywords. argument annotation, inter-map correspondence, large arguments

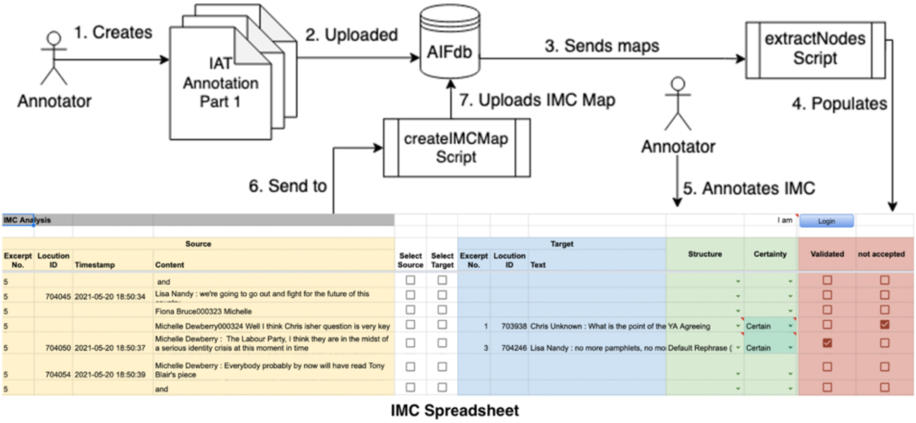
Annotating very large arguments presents particular problems, which can be naively tackled by simply dividing into sub-tasks – rather than try to analyse 5,000 words of argumentation as a whole, instead split into 20 sub-tasks of 250 words each. For environments in which arguments can be artificially constrained, (such as `kialo.com` and `debategraph.org`, for example) this solution can suffice. In general, however, the problem is that synthesising the solutions to the sub-tasks is a major challenge in itself. IMC-Tool offers a solution to this problem by providing a method to add *inter-map correspondence*, or IMC, to the process of argument analysis, and specifically to argument analysis conducted using IAT (Inference Anchoring Theory) [1].

IMC-Tool consists of multiple components; an *annotation tracker spreadsheet* which stores id numbers of annotated argument maps, an *imc spreadsheet* which is used by the annotators for the IMC argument annotation, the *extractNodes script* which populates the imc spreadsheet with node details for use for IMC annotation, and a *createIMCMap script* which turns the identified relations into an IAT map uploaded to AIFdb. Figure 1 shows a diagram demonstrating how these components work together.

Firstly, the initial annotation is split into sub-tasks and annotated using OVA+ [2]. Each map is uploaded to AIFdb [3] and a unique AIFdb map id for each map is stored in the annotation tracker spreadsheet. To begin the IMC procesure, by using the annotation tracker spreadsheet, the extractNodes script extracts all required node details for the task, by requesting each map from AIFdb. The map is parsed and the data is used to populate the imc spreadsheet with the content of each node.

The annotator carries out the IMC annotation by selecting the source and target nodes in the IMC spreadsheet. Subsequently, the target node's details are appended to the source node's row, containing the node's excerpt number, locution id and content. The analyst can specify the identified structure of the relation from a drop down list, containing relation types such as inference, conflict and rephrase, and their certainty that the identified relation is correct. The annotation is then verified by another annotator, who will either accept or reject the implementation. An example of the IMC spreadsheet containing IMC annotations is shown in Figure 1.

[1] Corresponding Author. E-mail:k.gorska@dundee.ac.uk



Source						Target						
Excerpt No.	Locution ID	Timestamp	Content	Select Source	Select Target	Excerpt No.	Locution ID	Text	Structure	Certainty	Validated	not accepted
5			and	□	□						□	□
5	704045	2021-05-20 18:50:34	Lisa Nandy : we're going to go out and fight for the future of this	□	□						□	□
5			Fiona Bruce000323 Michelle	□	□						□	□
5			Michelle Dewberry000324 Well I think Chris isher question is very key	□	□	1	703938	Chris Unknown : What is the point of the	YA Agreeing	Certain	□	☑
5	704050	2021-05-20 18:50:37	Michelle Dewberry : The Labour Party, I think they are in the midst of a serious identity crisis at this moment in time	□	□	3	704246	Lisa Nandy : no more pamphlets, no mo	Default Rephrase (	Certain	☑	□
5	704054	2021-05-20 18:50:39	Michelle Dewberry : Everybody probably by now will have read Tony Blair's piece	□	□						□	□
5			and	□	□						□	□

Figure 1. Model of the IMC process using IMC-Tool.

Once the IMC annotation is complete and all relations between parts have been implemented, the createIMCMap script is used to build the complete IAT structure of all identified relations. The source and target locutions are matched to the corresponding locution nodes in the annotated maps in AIFdb. The relation type identified by the annotator is automatically implemented using IAT structure, and once complete for all rows, a map of the complete IMC annotation is uploaded to AIFdb. When all individual maps and the map of identified relations between them are added to one corpus, AIFdb resolves all duplication of nodes, resulting in a complete IAT analysis of the full text.

The IMC procedure described was applied to a 280,000-word corpus of 30 episodes of IAT-annotated topical debate, QT30 [4]. The IMC analysis was completed in the context of near real-time analysis by 4-6 annotators in around 90-100 minutes per hour-long debate. In the QT30 corpus, IMC relations make up 14% of all argumentative relations. Out of all IMC relations, there is a predominance of rephrases (64% compared to 22% and 13% of inferences and conflicts respectively), which is also a higher proportion than the 43% found in non-IMC relations. This suggests that in order to understand non-local phenomena, we must first understand rephrase. Another area of future work is evaluating the inter-annotator agreement of the IMC procedure.

References

[1] Reed, C.A. and Budzynska, K. (2011) ”How dialogues create arguments” in van Eemeren, F.H. et al. (eds) Proceedings of the 7th Conference of the International Society for the Study of Argumentation (ISSA 2010), Sic Sat, Amsterdam

[2] Janier M, Lawrence J, Reed C. OVA+: an Argument Analysis Interface. InComputational Models of Argument 2014 (pp. 463-464). IOS Press.

[3] Lawrence J, Bex F, Reed C, Snaith M. AIFdb: Infrastructure for the argument web. InComputational Models of Argument 2012 (pp. 515-516). IOS Press.

[4] A. Hautli-Janisz, Z. Kikteva, W. Siskou, K. Gorska, R. Becker, and C. Reed. QT30: A Corpus of Argument and Conflict in Broadcast Debate. In Proceedings of the Language Resources and Evaluation Conference, Marseille, France, Jun. 2022, pp. 3291—3300.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220172

CPrAA – A Checker for Probabilistic Abstract Argumentation

Nikolai KÄFER [1]
Technische Universität Dresden, Germany

Keywords. probabilistic semantics, reasoning problems, constraint solving

1. Introduction

Classical *argumentation semantics* [1] determine which arguments of a given argumentation framework (AF) are considered to be jointly acceptable in light of the AF's attack relation. A collection of such compatible arguments is called an extension; thus semantics can be seen as functions that associate AFs with sets of extensions. Recently, abstract argumentation has been lifted into probabilistic settings to allow modelling of uncertainty, beliefs, and other quantitative aspects, giving rise to various *probabilistic argumentation semantics*. These include semantics working on the marginal probabilities of single arguments [2], notions to capture admissibility and complete semantics in the probabilistic setting [3], and direct liftings on the level of extensions of classical semantics [4].

For a given AF, a probabilistic semantics induces a family of probability distributions over the AF's extensions. Notably, while there is only a finite number of possible extensions for finite argumentation graphs, the space of distributions over extensions is infinite. As a result, reasoning problems for probabilistic semantics come with additional challenges of representing and enumerating solutions.

CPrAA[2] is a Python tool developed in the context of [3] that is capable of solving common reasoning problems in the probabilistic setting. Given an AF as input and selecting one or multiple probabilistic semantics from [2,3,4], the tool can

- compute distributions satisfying the semantics' constraints, or get an assertion that no such distribution exists,
- check credulous and skeptical acceptance of single arguments in the AF under a given threshold (see [3]), i.e., verify whether the marginal probability of the argument exceeds the threshold for at least one, or, respectively, all distributions,
- find a distribution under which the marginal probability of a given argument is maximal (or minimal),
- enumerate infinite solution spaces by finding the distributions at corners of solution polytopes,
- utilize a number of labelling schemes [5] to yield the finite set of all labellings corresponding to the (potentially infinitely many) distributions.

[1] E-mail: nikolai.kaefer@tu-dresden.de, ORCID: `https://orcid.org/0000-0002-0645-1078`
[2] `https://perspicuous-computing.science/cpraa/`

2. Tool Architecture

Based on the input AF, the selected semantics, and the chosen task, CPrAA generates a set of constraints on distributions for the AF. Subsequently, the constraints are passed to an appropriate backend solver, with two general classes of solvers available.

First, linear solvers (via CVXOPT [6]) are applicable for semantics where all induced constraints are linear, which is the case for the majority of semantics taken from the literature (see [3] for details). They allow convex optimization tasks like the minimization or maximization of marginal probabilities mentioned above. Linear constraints further imply that the solution space forms a convex polytope. By enumerating the distributions found at the corners of the polytope, one yields an explicit representation of the solution space: all distributions within this space arise as convex combination of the corner distributions.

Second, SMT solvers like Z3 [7] cover all probabilistic semantics characterized by polynomial constraints. This includes not only all semantics from [2,3,4], but also their respective complement semantics, that is, a semantics inducing exactly those distributions not induced by the original semantics. Further, they allow enforcement of additional context-specific constraints by using the standard SMT-LIB format [8], e.g., to specify the conditional probability for an argument to be accepted given that certain other arguments are not accepted. In addition to Z3, all SMT solvers available via the pySMT interface [9] can be used as backend.

The tool comes with a rich command line interface for direct interaction and can also be included as a Python library in other projects. Due to the constraint-based design, extending CPrAA with new probabilistic semantics for AFs is straightforward.

Acknowledgments. I'd like to thank Christel Baier, Martin Diller, Clemens Dubslaff, Sarah Gaggl, and Holger Hermanns for the collaboration that sparked the tool's development. This work was funded by DFG grant 389792660 as part of TRR 248 – CPEC.

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995;77(2):321-58.

[2] Hunter A, Thimm M. Probabilistic Reasoning with Abstract Argumentation Frameworks. Journal of Artificial Intelligence Research. 2017;59:565-611.

[3] Käfer N, Baier C, Diller M, Dubslaff C, Gaggl SA, Hermanns H. Admissibility in Probabilistic Argumentation. Journal of Artificial Intelligence Research. 2022;74.

[4] Thimm M, Baroni P, Giacomin M, Vicig P. Probabilities on Extensions in Abstract Argumentation. In: Proceedings of TAFA 2017. vol. 10757 of LNCS. Springer; 2017. p. 102-19.

[5] Käfer N. A General Framework for Probabilistic Abstract Argumentation; 2020. Master's thesis. Saarland University.

[6] Andersen M, Dahl J, Vandenberghe L. CVXOPT – Python Software for Convex Optimization; 2014. `https://cvxopt.org/`.

[7] de Moura L, Bjørner N. Z3: An Efficient SMT Solver. In: Tools and Algorithms for the Construction and Analysis of Systems. vol. 4963 of Lecture Notes in Computer Science. Springer; 2008. p. 337-40.

[8] Barrett C, Fontaine P, Tinelli C. The Satisfiability Modulo Theories Library (SMT-LIB); 2016. `http://smt-lib.org`.

[9] Gario M, Micheli A. PySMT: a solver-agnostic library for fast prototyping of SMT-based algorithms; 2015. SMT Workshop 2015.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220173

COGNICA: Cognitive Argumentation

Adamos KOUMI [a] and Antonis KAKAS [a] and Emmanuelle DIETZ [b]
[a] *University of Cyprus, Cyprus*
[b] *Airbus Central R&T, Germany*

Keywords. Computational Argumentation, Cognitive Modeling, Explainable AI

Cognitive Argumentation [1] is the study of synthesis of cognitive principles within formal computational frameworks of argumentation. Cognitive principles are drawn from our understanding of human reasoning as acquired across a wide range of disciplines, such as Cognitive Science, Philosophy and Linguistics. They inform and regulate the computational process of argumentation to be cognitively compatible to human argumentation and reasoning. By "humanizing" the form of machine argumentation we can facilitate an effective and naturally enhancing integration of machines with the human.

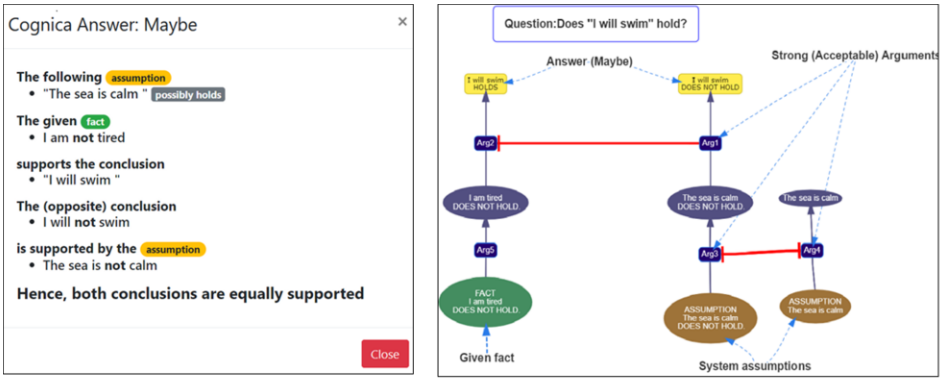
COGNICA[1] is a system that implements the framework of Cognitive Argumentation with emphasis on conditional reasoning. It is based on the particular work of Johnson-Laird and Byrne, "Conditionals: A Theory of Meaning, Pragmatics, and Inference" and the mental models theory that underlies this work [2]. Using argumentation it is possible to accommodate and extend their interpretation of the various types of conditionals used in human discourse. Importantly, these argumentation-based interpretations can be extended from individual conditionals to sets of conditionals of different types that together form a piece of knowledge on some subject of interest.

The COGNICA system has a simple interface of a Controlled Natural Language for expressing different types of conditional sentences. These are automatically translated into the GORGIAS[2] argumentation framework and executed by the GORGIAS system on top of which COGNICA is build. During this translation COGNICA automatically also forms priority arguments across the arguments that result from the different types of conditional statements in the knowledge, thus capturing the interaction between these individual conditional statements. The controlled natural language of COGNICA allows one to enter conditionals of different types as *foreground knowledge*, i.e., the particular knowledge that the system would reason about. This may need to be complemented by some relevant *background knowledge* entered in the system, alongside the foreground knowledge, using exactly the same conditional form of controlled natural language.

Example (Foreground Knowledge)**.**
If *I am not tired* ***then*** *I will swim.*
If *the sea is crowded* ***then possibly*** *I will not swim.*
Only If *if the sea is calm* ***then*** *I will swim.*

Then given a certain situation where some specific facts hold the COGNICA system will consider queries and give a reply of "Yes", "No" or "Maybe". For example, when

[1] http://cognica.cs.ucy.ac.cy/COGNICAb/login.php
[2] http://gorgiasb.tuc.gr/GorgiasCloud.html



Cognica Answer: Maybe

×

The following assumption
- "The sea is calm " possibly holds

The given fact
- I am **not** tired

supports the conclusion
- "I will swim "

The (opposite) conclusion
- I will **not** swim

is supported by the assumption
- The sea is **not** calm

Hence, both conclusions are equally supported

Close

Question:Does "I will swim" hold?
Answer (Maybe)
Strong (Acceptable) Arguments
I will swim HOLDS
I will swim DOES NOT HOLD
Arg2
Arg1
I am tired DOES NOT HOLD.
The sea is calm DOES NOT HOLD.
The sea is calm
Arg5
Arg3
Arg4
FACT I am tired DOES NOT HOLD.
ASSUMPTION The sea is calm DOES NOT HOLD.
ASSUMPTION The sea is calm
Given fact
System assumptions

Figure 1. Verbal and Visual explanations of COGNICA for Example 1.

given the facts "I am not tired" and "the sea is not crowded", COGNICA will reply "Maybe" to the query of "Will I swim?".

Importantly, COGNICA provides automatically generated explanations in verbal and graphical form for its answers. Figure 1 shows the explanations for the answer "Maybe" in the above example. Note that the graphical explanations present the argumentative reasoning by COGNICA as *"reasoning pathways"* of the "mind" of the COGNICA system. COGNICA offers the opportunity for carrying out large scale empirical studies of comparison between human and machine reasoning and to examine the nature of an argumentation-based human-machine interaction. For example, to study the effect that explanations can have on humans when reasoning or deciding what action to pursue.

A first such study was carried out where participants were asked to answer questions based on foreground information that typically included three to five conditionals from everyday life. They were then asked to reconsider these questions after they were shown the answer of the COGNICA system together with its explanations (verbal and/or visual). Initial results show that in around 50% of the cases where the conclusion of the human participants differed from the one of the machine, the participants changed their answer when they saw the explanations of the system. It is also possible to observe that this kind of interaction with the system motivates participants to "drift" to more "careful reasoning" as they progress in the experiment, in accordance with the argumentation theory of Mercier and Sperber [3]. The exercise is ongoing and open to anyone. It can be found at `http://cognica.cs.ucy.ac.cy/cognica_evaluation/index.html`. We are also currently designing new such experiments in order to investigate how this argumentation-based and explanation driven machine-human interaction varies across the population with different cognitive and personality characteristics.

References

[1] Dietz E, Kakas AC. Cognitive Argumentation and the Selection Task. In: Proceedings of the Annual Meeting of the Cognitive Science Society, 43. Cognitive Science Society; 2021. p. 1588-94.

[2] Johnson-Laird PN, Byrne RM. Conditionals: a theory of meaning, pragmatics, and inference. Psychological Review. 2002;109(4):646-78.

[3] Mercier H, Sperber D. Why do humans reason? Arguments for an argumentative theory. Behavioral and Brain Sciences. 2011;34(2):57-74.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220174

probo2: A Benchmark Framework for Argumentation Solvers

Jonas KLEIN and Matthias THIMM
Artificial Intelligence Group, University of Hagen, Germany

Abstract. We introduce probo2, an end-to-end benchmark framework for abstract argumentation solvers. It offers evaluation capabilities and analysis features for a wide range of computational problems and is easily customizable.

Keywords. abstract argumentation, solver, benchmark, evaluation

1. Introduction

Approaches to formal argumentation [1,7] are a significant part of active research in Artificial Intelligence. Especially, solving reasoning problems in Dung's [5] abstract argumentation framework is fundamental for many argumentation systems. Solving such computational complex reasoning problems requires efficient algorithms. In recent years, there has been an increased effort to develop such algorithms and solvers [2]. To assess the performance of a system, the evaluation on meaningful benchmarks is crucial and adequate tools supporting researchers and developers in this task are essential. An example of such a tool is the original probo [4] framework, which provides a general interface and allows the comparison of abstract argumentation solvers in terms of correctness and performance. However, this tool lacks functionality for extensive analysis and visualizations.

In this extended abstract, we present probo2, an end-to-end benchmark framework for abstract argumentation solvers. This framework aims at providing researchers and developers with an easy-to-use, robust, and flexible command-line tool to simplify and speed up typical tasks in benchmarking abstract argumentation solvers. Therefore, probo2 offers functionalities to (1) generate benchmarks, (2) execute solvers and collect data, (3) verify the correctness of solvers, (4) compare the performance of solvers, (5) perform statistical analysis and—of particular interest in the research context—(6) generate "publication-ready" visualizations of results. probo2 offers a standardized pipeline for evaluating argumentation solvers. Users can specify which performance metrics to use and how to visualize the results, allowing to compare different solvers easily. To reproduce results reliably, any experiment is completely described by a configuration file. In addition, probo2 supports the parallelization of experiments. During development, we set a strong focus on customizability. The modular design of probo2 allows users to modify and extend functionality to their needs. For now, the framework focuses on abstract argumentation. However, since probo2 is still being further developed, extensions to structured argumentation frameworks are foreseen for the future.

2. Implementation Overview

probo2 is written in Python. The source code and a detailed documentation are publicly available on GitHub[1]. It is compatible with any solver implementing the ICCMA interface. All computational problems and semantics of past competitions are supported, including classical problems such as deciding acceptability and enumerating extensions as well as corresponding tasks for dynamic problems and counting tasks. probo2 accepts abstract argumentation frameworks in the ASPARTIX format [6] and the Trivial Graph Format[2]. The framework also incorporates various graph generators such as the *StableGenerator* [4] or *AFBenchGen* [3]. This enables the generation of diverse and challenging argumentation graphs. A configuration file fully describes experiments, allowing for reproducing results reliably. Correctness of solutions is verified by specifying a reference solver or providing pre-calculated solutions. In order to compare the performance of different solvers, various established performance measures such as the penalized average runtime, instance coverage, and the ICCMA scorings are available. For more extensive statistical analysis, probo2 offers parametric and non-parametric significance tests, including posthoc analysis. Users can choose between various result visualizations, including scatter plots, cactus plots, and pie charts. Tabular data can be directly exported to LatTeX, HTML, or just plain text formats. As stated before, a key feature of probo2 is its customizability. All visualizations can be customized according to user preferences. Custom functionalities can be integrated into the existing pipeline with a single command.

3. Summary

In this extended abstract, we gave a short overview on the probo2 benchmark framework. probo2 is continuously being improved and we welcome any feedback or suggestions for further features.

References

[1] Pietro Baroni, Dov Gabbay, Massimilino Giacomin, and Leendert van der Torre, editors. *Handbook of Formal Argumentation*. College Publications, 2018.

[2] Federico Cerutti, Sarah A. Gaggl, Matthias Thimm, and Johannes P. Wallner. Foundations of implementations for formal argumentation. In *Handbook of Formal Argumentation*, chapter 15. College Publications, 2018.

[3] Federico Cerutti, Massimiliano Giacomin, and Mauro Vallati. Generating challenging benchmark afs. *COMMA*, 14:457–458, 2014.

[4] Federico Cerutti, Nir Oren, Hannes Strass, Matthias Thimm, and Mauro Vallati. A benchmark framework for a computational argumentation competition. In *COMMA*, pages 459–460, 2014.

[5] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–358, 1995.

[6] U. Egly, S. A. Gaggl, and S. Woltran. Answer-set programming encodings for argumentation frameworks. *Argument and Computation*, 1(2):147–177, 2010.

[7] Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors. *Handbook of Formal Argumentation*, volume 2. College Publications, August 2021.

[1] https://github.com/aig-hagen/probo2

[2] http://en.wikipedia.org/wiki/Trivial_Graph_Format

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220175

Polemicist: A Dialogical Interface for Exploring Complex Debates

John LAWRENCE [a,1], Jacky VISSER [a] and Chris REED [a]
[a] *Centre for Argument Technology, University of Dundee*

Abstract. In this paper, we present *Polemicist* [2], a dialogical interface for exploring complex debates from the BBC Radio 4 programme *The Moral Maze*. Polemicist allows the user to interact with software agents representing the participants in the original programme. The software enables the user to explore the topic as they wish, asking questions to dive deeper on the areas that interest them most.

Keywords. BBC Radio, debate, DGDL, dialogue system, Moral Maze

The Polemicist[2] application allows users to explore issues discussed in the BBC Radio 4 Moral Maze programme[3] by interacting with software agents which represent the participants from the debate and whose knowledge bases are extracted from analysis of the original episodes.

The Moral Maze is billed as *combative, provocative and engaging live debate examining the moral issues behind one of the week's news stories* and is broadcast on the UK's leading non-music radio station. Each episode features four regular panellists along with a series of 'witnesses' – experts or others knowledgeable in the field under discussion – who are questioned in turn by the panellists. Each weekly episode is 45 minutes in duration and contains a high density of argumentative content. An average analysed episode represented in the Argument Interchange Format [1] contains around 500 Information nodes (I-nodes) – the propositional contents of the arguments – and 250 Scheme nodes (S-nodes) – capturing the application of patterns of relationship between them.

Such a sizeable knowledge graph can prove extremely difficult to navigate and understand. Polemicist addresses this problem, effectively translating the navigation of the graph into a series of dialogical moves conducted according to a particular dialogue game [2]. Polemicist uses a fixed protocol, defined in the Dialogue Game Description Language (DGDL) [3], allowing the user to take on the role of the moderator of the debate: selecting topics, controlling the flow of the dialogue, and thus exploring all the angles of the rich argumentative content on offer. Playing the role of moderator allows the user to rearrange the arguments and create wholly novel virtual discussions between the contributions of participants that might not have engaged directly in the original debate, while staying true to their stated opinions.

The Polemicist dialogue interface features two main panels (see Figure 1). On the left, a list of participants can be seen along with green and red highlighting showing their

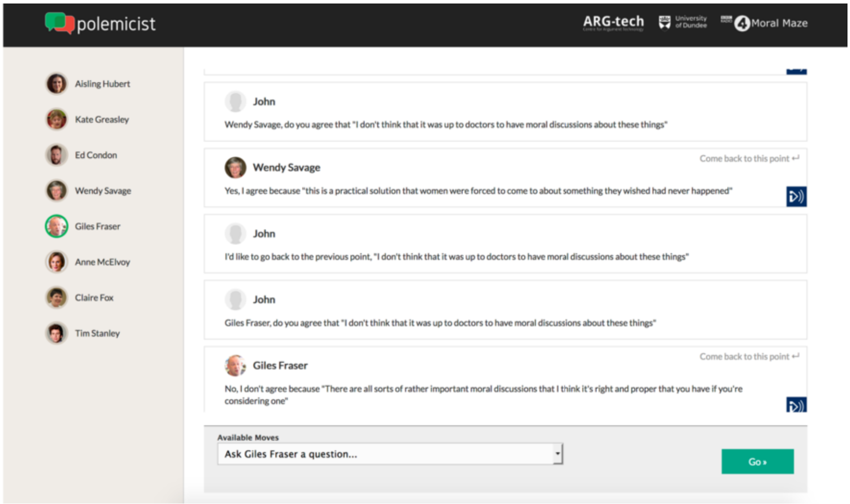
[1] Corresponding Author: School of Science and Engineering (Computing), University of Dundee, DD1 4HN, United Kingdom; E-mail: j.lawrence@dundee.ac.uk

[2] http://polemici.st

[3] https://www.bbc.co.uk/programmes/b006qk11

agreement or disagreement with the most recent point made. This highlighting allows the user to pursue a line of questioning which explores these opinions and the reasons behind them.

The right-hand side of the dialogue interface shows a panel at the bottom allowing the user to first select a participant to address a question to, and then select a question either asking for an opinion or for reasons why a participant holds that opinion. Above this is a record of the dialogue so far. This record allows the user to view the dialogue as well as return to previous points, and to listen to the original audio associated with each text segment.



John

Wendy Savage, do you agree that "I don't think that it was up to doctors to have moral discussions about these things"

Wendy Savage

Yes, I agree because "this is a practical solution that women were forced to come to about something they wished had never happened"

John

I'd like to go back to the previous point, "I don't think that it was up to doctors to have moral discussions about these things"

John

Giles Fraser, do you agree that "I don't think that it was up to doctors to have moral discussions about these things"

Giles Fraser

No, I don't agree because "There are all sorts of rather important moral discussions that I think it's right and proper that you have if you're considering one"

Ask Giles Fraser a question...

Figure 1. The Polemicist dialogue interface

Whilst Polemicist currently relies on pre-annotated material from AIFdb[4] to provide the responses of the software agents in a dialogue, it presents a valuable potential use case for automatically mined arguments. If the argumentative structure in a radio transcription can be extracted in real-time by argument mining, conversations in Polemicist could take place within moments of transmission ending. Combining the dialogue interfaces with a robust argument mining platform could enable users to discuss any issue of their choosing with any person whose opinions on that topic have been previously recorded.

Acknowledgements This research was supported by EPSRC grant EP/N014871/1, and wouldn't have been possible without the support of the *Moral Maze* production team.

References

[1] Chesñevar C, McGinnis J, Modgil S, Rahwan I, Reed C, Simari G, et al. Towards an Argument Interchange Format. The Knowledge Engineering Review. 2006;21(04):293-316.

[2] Prakken H. Formal systems for persuasion dialogue. The Knowledge Engineering Review. 2006;21(2):163–188.

[3] Bex F, Lawrence J, Reed C. Generalising argument dialogue with the Dialogue Game Execution Platform. In: Parsons S, Oren N, Reed C, Cerutti F, editors. Proc of COMMA 2014. IOS Press; 2014. p. 141-52.

[4] http://www.aifdb.org/

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220176

User-Centric Argument Mining with ArgueMapper and Arguebuf

Mirko LENZ [a,1] and Ralph BERGMANN [a,b]
[a] *Trier University, Universitätsring 15, 54296 Trier, Germany*
[b] *DFKI Branch Trier University, Behringstr. 21, 54296 Trier, Germany*

Keywords. Argument Graphs, Argument Mining, User Interfaces, Open Source

1. Introduction

Contrary to unstructured representations like natural language texts, *argument graphs* enable advanced analysis of an argument's structure which consists of linked Argumentative Discourse Units (ADUs). Since most existing works dealing with the creation of such graphs are primarily geared towards *experts* and neglect the needs of *developers* and *laymen*, we propose (i) an intuitive, stable, and scalable *tool* (ArgueMapper)[2] for creating and browsing graph-based representations of arguments by experts and laymen alike and (ii) a straightforward *format* (Arguebuf)[3] enabling developers to build related tools and exchange data more easily. Both ArgueMapper and Arguebuf are available under the permissive MIT license and are open to any kind of contribution.

2. ArgueMapper: A Tool for Manual Argument Mining

This section will highlight some features of ArgueMapper compared to existing tools like Online Visualization of Arguments (OVA) [1] and MonkeyPuzzle [2].

Intuitive Interface Our tool (see Figure 1) complies with Nielsen's usability heuristics [4] to ensure as little friction as possible for laymen. At the same time, it is similar enough to OVA to be familiar to experts as well.

Optimized for Mobile Devices ArgueMapper is fully functional on smartphones and tablets by providing finger-optimized buttons and gesture controls.

Auto-Layout We combined ideas of OVA and MonkeyPuzzle by implementing a hierarchical automatic layout algorithm that runs entirely in the user's browser.

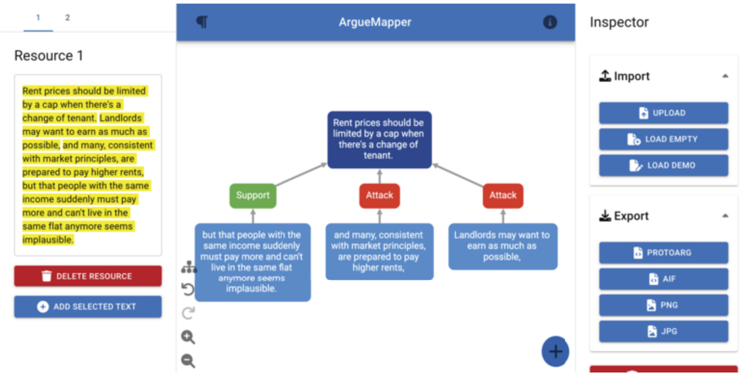
State Management To prevent loss of unsaved data, the app's state is always stored in the browser's storage. In addition, we also fully support undo/redo functionality.

Modern Development Stack To simplify contributions, we built ArgueMapper using modern tooling like TypeScript and React. It has a modular architecture and thus may be embedded into other systems as well.

[1] Corresponding Author. Mail: `info@mirko-lenz.de`

[2] `https://github.com/recap-utr/arguemapper`, Demo at `https://arguemapper.uni-trier.de`

[3] `https://github.com/recap-utr/arguebuf`



Resource 1

Rent prices should be limited by a cap when there's a change of tenant. Landlords may want to earn as much as possible, and many, consistent with market principles, are prepared to pay higher rents, but that people with the same income suddenly must pay more and can't live in the same flat anymore seems implausible.

Rent prices should be limited by a cap when there's a change of tenant.

but that people with the same income suddenly must pay more and can't live in the same flat anymore seems implausible.

and many, consistent with market principles, are prepared to pay higher rents,

Landlords may want to earn as much as possible,

Figure 1. Three-pane layout of ArgueMapper with source texts on the left, the graph in the middle, and additional functions in the right sidebar. Example taken from the Microtexts corpus [3].

3. Arguebuf: A Format for Argument Graphs

In conjunction with ArgueMapper, we developed the format Arguebuf to address limitations of existing ones like Argument Interchange Format (AIF) [5] and SADFace [6].

Simple Specification Arguebuf is specified using the concise and intuitive language Protocol Buffers (Protobuf), meaning that it is easily expandable. Graphs may be serialized to JSON or a more efficient binary format for use with gRPC.

Superset of AIF and SADFace It is possible to transform every AIF graph or SADFace document into our new format without any information loss.

Code Generation Protobuf automatically creates native code for most programming languages. Among others, this enables code completion and type checks in IDEs.

Supercharged Python Implementation We provide an optimized Python client with advanced analysis features—for instance, importing legacy formats, converting from/to AIF, and integrating with Graphviz, NetworkX, and spaCy.

Acknowledgements

This work has been funded the by the DFG within the project *ReCAP-II* (No. 375342983) as part of the priority program RATIO (SPP-1999) as well as the *Studienstiftung*.

References

[1] Bex F, Lawrence J, Snaith M, Reed C. Implementing the argument web. Communications of the ACM. 2013 Oct;56(10):66-73.

[2] Douglas J, Wells S. Monkeypuzzle - Towards Next Generation, Free & Open-Source, Argument Analysis Tools. In: CMNA@ICAIL. CEUR; 2017. .

[3] Peldszus A, Stede M. An annotated corpus of argumentative microtexts. In: Argumentation and Reasoned Action. vol. 2. Lisbon, Portugal: College Publications; 2015. p. 801-15.

[4] Nielsen J. Usability Engineering. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1994.

[5] Chesñevar C, Mcginnis, Modgil S, Rahwan I, Reed C, Simari G, et al. Towards an argument interchange format. Knowledge Engineering Review. 2006 Dec;21(4):293-316.

[6] Wells S. Datastores for Argumentation Data. In: Proceedings of the 20th Workshop on Computational Models of Natural Argument. CEUR Workshop Proceedings. CEUR; 2020. p. 31-40.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220177

Attractor - A Java Library for Gradual Bipolar Argumentation

Nico POTYKA [a]

[a] *Imperial College London, United Kingdom*

Keywords. Gradual Argumentation, Algorithms, Implementations

Gradual argumentation frameworks (GAFs) are abstract argumentation frameworks that interpret arguments numerically [1]. Figure 1 shows a simple GAF on the left. Nodes represent abstract arguments. *Buy* and *Sell* represent decisions (buy or sell stocks of a company) and *A1*, *A2*, *A3* represent arguments given by experts. Solid edges denote attack and dashed edges support relations. Every argument has an initial weight shown in the node. Intuitively, this weight is an apriori belief in the strength of the argument when ignoring the others.

Semantically, attackers should decrease the initial weight, while supporters should increase it. Various gradual semantics have been proposed, but many of them can be seen as instances of *modular semantics* [2]. Modular semantics assign strength values using an iterative procedure that initializes the strength values of arguments with their base scores and repeatedly update the values based on the strength of their attackers and supporters. To do so, an *aggregation function* aggregates the strength values of attackers and supporters and an *influence function* adapts the base score based on the aggregate. While this process may start oscillating in cyclic graphs [2], it usually converges quickly in practice [3]. Figure 1 illustrates this procedure for the DF-QuAD semantics [4] on the right.

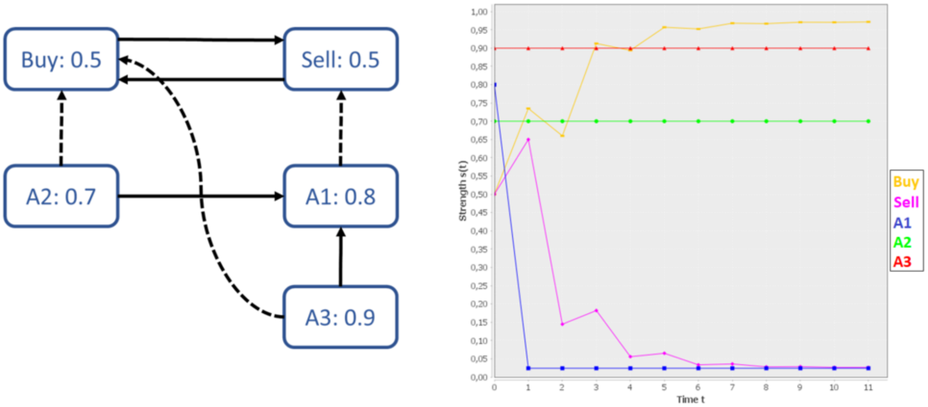


Figure 1. Example of a GAF and illustration of strength computation for DF-QuAD.

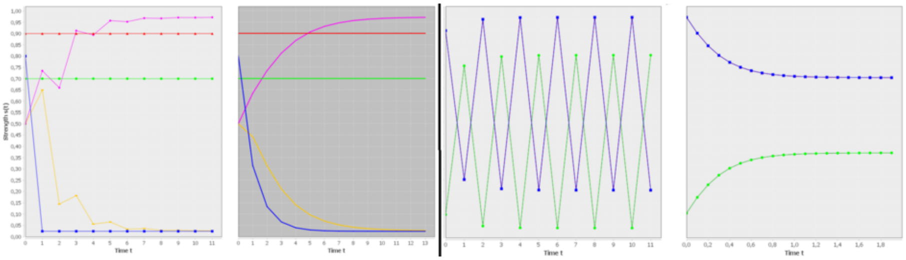


Figure 2. Discrete vs. continuized semantics.

Attractor[1] allows implementing and evaluating gradual argumentation frameworks in Java in a straightforward way. Implementation of several semantics, including Df-QuAD [4], Euler-based [5], Quadratic Energy [3] and MLP-based semantics [6] can be used out of the box. Other modular semantics can be easily implemented by combining pre-implemented aggregation and influence functions. New aggregation and influence functions can be added to implement novel modular semantics. Non-modular semantics can be integrated as well if they maintain a simple interface. Attractor also provides auxiliary functions to plot the evolution of strength values like in Figure 1 and to evaluate and plot the computational performance of different semantics and algorithms on GAFs of increasing size.

Attractor implements two reasoning algorithms that are based on the observation that gradual semantics can be seen as dynamical systems [3]. This view allows continuizing the iterative computation of strength values described above. Continuization can improve the convergence guarantees of modular semantics in cyclic GAFs without changing their semantics in convergent cases [7]. Figure 2 illustrates on the left, how the continuized semantics converges to the same strength values when the strength values under the discrete semantics converge. On the right, it shows an example from [2] where the strength values under the discrete semantics start oscillating, while its continuization finds a reasonable compromise.

References

[1] Baroni P, Rago A, Toni F. How many properties do we need for gradual argumentation? In: AAAI Proceedings. AAAI; 2018. p. 1736-43.

[2] Mossakowski T, Neuhaus F. Modular Semantics and Characteristics for Bipolar Weighted Argumentation Graphs. arXiv preprint arXiv:180706685. 2018.

[3] Potyka N. Continuous Dynamical Systems for Weighted Bipolar Argumentation. In: KR Proceedings; 2018. p. 148-57.

[4] Rago A, Toni F, Aurisicchio M, Baroni P. Discontinuity-Free Decision Support with Quantitative Argumentation Debates. In: KR Proceedings; 2016. p. 63-73.

[5] Amgoud L, Ben-Naim J. Evaluation of arguments in weighted bipolar graphs. In: ECSQARU Proceedings. Springer; 2017. p. 25-35.

[6] Potyka N. Interpreting Neural Networks as Gradual Argumentation Frameworks. In: AAAI Proceedings; 2021. p. 6463-70.

[7] Potyka N. Extending Modular Semantics for Bipolar Weighted Argumentation. In: AAMAS Proceedings; 2019. p. 1722-30.

[1] https://sourceforge.net/projects/attractorproject/

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220178

Gorgias Cloud: On-line Explainable Argumentation

Nikolaos I. SPANOUDAKIS [a,1], Georgios GLIGORIS [a] Antonis C. KAKAS [b] and Adamos KOUMI [b]
[a] *Technical University of Crete, Greece*
[b] *University of Cyprus, Cyprus*

Keywords. Computational Argumentation, SaaS, IDE, xAI

Gorgias Cloud offers argumentation-based decision making as a service. The service includes an integrated development environment for the theories, testing and execution based on user scenarios, and, finally, an API for use by user applications.

Gorgias is a structured argumentation framework where arguments are constructed using a basic scheme of *argument rules*. Two types of arguments are constructed within a Gorgias argumentation theory: *object-level* arguments and *priority arguments* expressing a preference between other arguments. Admissible *composite arguments* supporting a claim typically include both types of arguments. The Gorgias framework was introduced in [1], extended in [2] and applied to a variety of real-life application problems in [3].

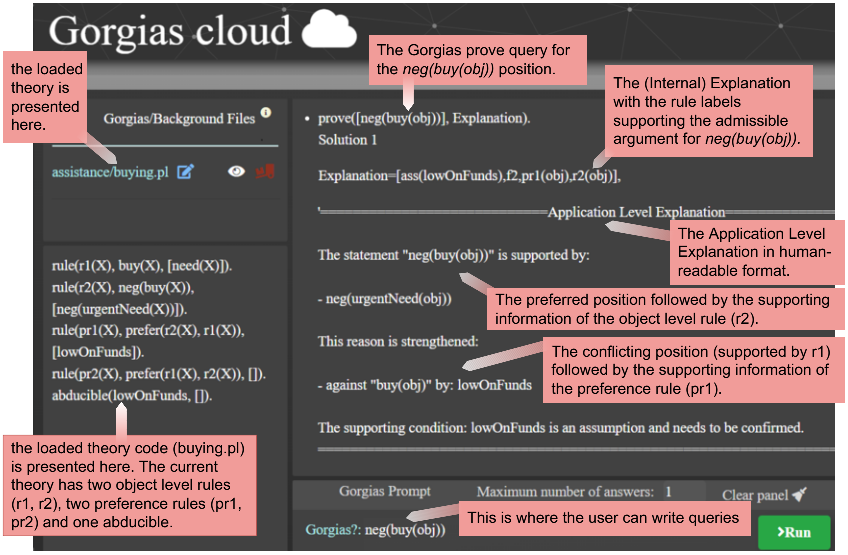
The Gorgias system allows us to code argumentation theories of the form described above and subsequently query the system to find out if there is an admissible (composite) argument that supports a desired Claim. The system of Gorgias has been publicly available since 2003 and has been used by several research groups to develop prototype real-life applications of argumentation in a variety of application domains. Today, it is available as a service over the internet with Gorgias Cloud, which provides an integrated environment for developing applications of argumentation with three novel features:

1. Assistance for editing argumentation theories in the internal code language of Gorgias using templates for object level or priority arguments and abducibles
2. Ability to store multiple scenarios to test the behaviour of developed theories
3. REST-compliant web API so that Gorgias queries can be executed from any other programming environments (e.g. Java, Python) used for developing applications.

The code shown on the left hand side of the Gorgias Cloud Execution Panel in Figure 1, shows a simple example of an argumentation theory. Gorgias rules are in the form $rule(label, conclusion, supporting\ information)$. Those with labels $r1(X)$ and $r2(X)$ are for and against buying an object. The priority argument rules $pr1(X), pr2(X)$ support one or the other of the object-level rules, depending on whether we are low on funds.

Let us assume that we have an object, obj, for which $need(obj)$ and $neg(urgent\_need(obj))$ hold. The query for asking if there is an admissible composite argument supporting the conclusion to not buy the obj, i.e. $neg(buy(obj))$, is posed at

[1] Corresponding Author: Nikolaos I. Spanoudakis, Applied Mathematics and Computers Laboratory, Technical University of Crete, University Campus, 73100 Chania, Greece; E-mail: nikos@amcl.tuc.gr.



the loaded theory is presented here.

The Gorgias prove query for the *neg(buy(obj))* position.

The (Internal) Explanation with the rule labels supporting the admissible argument for *neg(buy(obj))*.

The Application Level Explanation in human-readable format.

The preferred position followed by the supporting information of the object level rule (r2).

The conflicting position (supported by r1) followed by the supporting information of the preference rule (pr1).

the loaded theory code (buying.pl) is presented here. The current theory has two object level rules (r1, r2), two preference rules (pr1, pr2) and one abducible.

This is where the user can write queries

```
rule(r1(X), buy(X), [need(X)]).
rule(r2(X), neg(buy(X)),
[neg(urgentNeed(X))]).
rule(pr1(X), prefer(r2(X), r1(X)),
[lowOnFunds]).
rule(pr2(X), prefer(r1(X), r2(X)), []).
abducible(lowOnFunds, []).
```

```
• prove([neg(buy(obj))], Explanation).
Solution 1

Explanation=[ass(lowOnFunds),f2,pr1(obj),r2(obj)],

'==========================Application Level Explanation==========

The statement "neg(buy(obj))" is supported by:

- neg(urgentNeed(obj))

This reason is strengthened:

- against "buy(obj)" by: lowOnFunds

The supporting condition: lowOnFunds is an assumption and needs to be confirmed.
```

Gorgias?: neg(buy(obj))

Figure 1. Execution Panel of Cloud Gorgias, the theory on the left and the results of the query in the center.

the prompt at the bottom of the Execution Panel (see Figure 1). In the Execution Panel we see both a) the *Internal Explanation* (IE) of the composite argument E, $Explanation = [ass(lowOnFunds), f1, pr1(obj), r2(obj)]$, where $f1$ is the label of the belief $neg(urgent\_need(obj))$, as well as b) the *Application Level Explanation* (ALE) in a human-friendly format. ALE starts by presenting the Claim, followed by the supporting information of the object level rule that supports it. Then, if this Claim has been strengthened by a preference rule over another position, then this conflicting position is printed along with the supporting information of the preference rule. Finally, the user is informed that any employed assumptions should be confirmed.

The generation of the ALE from the IE is an important feature of the Gorgias Cloud as it exhibits the desired characteristics of being *attributive, contrastive and actionable*:

- **Attributive:** Extracted from the object-level argument rules in E.
- **Contrastive:** Extracted from the priority argument rules in E.
- **Actionable:** Extracted from the hypothetical or abducible arguments in E.

Developers and users can use the ALEs to evaluate the behaviour of their system with respect to its specifications. This is made easy as these explanations are at the same high cognitive and language level as that of the application domain of the system.

References

[1] Kakas AC, Mancarella P, Dung PM. The Acceptability Semantics for Logic Programs. In: The 11th International Joint Conference on Logic Programming; 1994. p. 504-19.

[2] Kakas AC, Moraitis P. Argumentation based decision making for autonomous agents. In: The Second International Joint Conference on Autonomous Agents & Multiagent Systems, (AAMAS 2003), July 14-18, Melbourne, Victoria, Australia. ACM; 2003. p. 883-90.

[3] Kakas AC, Moraitis P, Spanoudakis NI. *GORGIAS*: Applying argumentation. Argument & Computation. 2019;10(1):55-81.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220179

Interpretable Machine Learning with Gradual Argumentation Frameworks

Jonathan SPIELER[a],Nico POTYKA[b], Steffen STAAB[a]
[a] *University of Stuttgart, Germany*
[b] *Imperial College London, United Kingdom*

Keywords. Gradual Argumentation, Explainable AI

As black-box neural networks are increasingly applied in intelligent systems, questions about their fairness, reliability and safety become louder. Recent work tried making them human-understandable by trying to learn parameters that can be well approximated by decision trees [1]. However, the tree remains just an approximation, which leaves the question how faithful it really captures the actual mechanics of the neural network. As it turns out, gradual argumentation frameworks (GAFs) [2] are closely related to multilayer perceptrons (MLPs), one of the main classes of neural networks. More precisely, every MLP corresponds to a GAF under the *MLP-based semantics*, and conversely, every acyclic GAF under this semantics corresponds to an MLP [3].

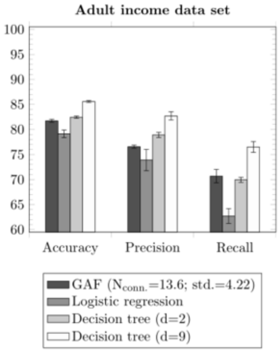
However, since a GAF with millions of attacks and supports between arguments is not easier to interpret than an MLP with millions of connections between neurons, we have to make sure that the original neural network is sparse. While learning sparse neural networks has become a more active area in recent years, current work does not focus on learning an interpretable network, but on decreasing the risk for overfitting, the memory and runtime complexity and the associated power consumption [4]. Even though the learnt networks are significantly sparser than dense networks, they are still too dense to be interpretable. Furthermore, while numerical inputs can be seen arguments with a numerical weight, MLPs result in more intuitive GAFs when all inputs are discrete. This is the opposite of what is usually done in the literature on learning neural networks, where even discrete features are often continuized (e.g., using word embeddings) to improve learning performance (while sacrificing interpretability).

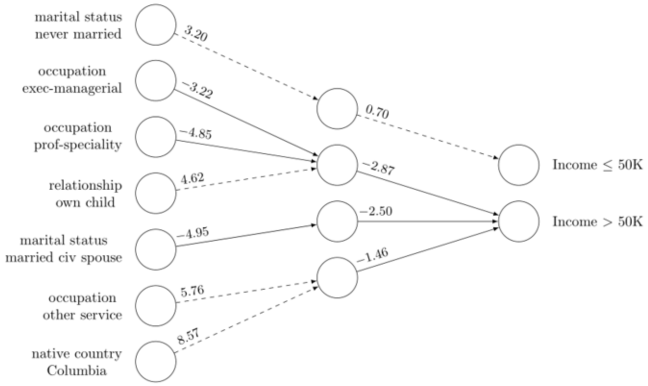
To learn a discrete sparse neural network, we apply structure learning ideas. Our search space consist of the space of all MLP structures that satisfy structural constraints. Examples for such constraints are the maximum depth (number of layers), the maximum width (number of arguments per layer), the maximum outdegree (number of outgoing edges per argument) and possible discretizations of continuous features like bins (the value falls in a particular interval) or fuzzy arguments (e.g., the value is *small*, *average*, *large*). In order to compare candidate structures, we assign a score to every candidate structure C as follows: we train C on the training set using the usual backpropagation procedure for MLPs and compute its accuracy. The score of C is then defined as

$$s_\lambda(C) = (1-\lambda)\cdot \text{Accuracy}(C, \mathcal{D}_{\text{train}}) + \lambda \cdot \frac{n_{max} - n_C}{n_{max}}.$$

The score consists of two terms that are weighted by a hyperparameter $\lambda \in [0, 1]$. The first term evaluates the accuracy, the second one the sparsity. In the second term, n_c is the number of edges in C and n_{max} the number of edges in the fully connected GAF corresponding to C .

As the search space is exponentially large, we aim at finding a good structure, rather than the best one. To do so, we implemented a genetic algorithm. Let us emphasize that the genetic algorithm is responsible for finding a good structure, not for learning the parameters of the structure (the latter is done by backpropagation as usual). A detailled description of the algorithm and an evaluation can be found in the technical report [5]. As an example, we show a GAF (solid edges denote attacks, dashed edges supports) found for the Adult income dataset from the UCI machine learning repository and a performance comparison to Logistic Regression and Decision Trees of varying depth.





Overall, the performance of GAFs is usually better than logistic regression (which can only learn linearly separable functions) and comparable to decision trees. However, flat GAFs can sometimes obtain better performance than flat decision trees [5]. They can also be easier to comprehend as they are based on gradual influences rather than on long case differentiations. We are planning to improve the results by adding fuzzy arguments and joint attacks/supports to capture joint effects of inputs without increasing the depth of the network.

References

[1] Wu M, Parbhoo S, Hughes MC, Roth V, Doshi-Velez F. Optimizing for interpretability in deep neural networks with tree regularization. JAIR. 2021;72:1-37.

[2] Baroni P, Rago A, Toni F. How many properties do we need for gradual argumentation? In: AAAI Proceedings. AAAI; 2018. p. 1736-43.

[3] Potyka N. Interpreting Neural Networks as Gradual Argumentation Frameworks. In: AAAI Proceedings; 2021. p. 6463-70.

[4] Ma R, Niu L. A survey of sparse-learning methods for deep neural networks. In: WI Proceedings. IEEE; 2018. p. 647-50.

[5] Spieler J, Potyka N, Staab S. Learning Gradual Argumentation Frameworks using Genetic Algorithms. arXiv preprint arXiv:210613585. 2021.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220180

The *Skeptic* Web Service: Utilising Argument Technologies for Reason-Checking

Jacky VISSER [a,1] and John LAWRENCE [a]
[a] *Centre for Argument Technology, University of Dundee*

Abstract. *Skeptic* is a web service aimed at automatically providing pointers for the critical assessment of a persuasive text. That is, with a natural language text as input, the web service returns a ranked list of questions designed to help readers reason-check fake news and other contentious texts. Internally, *Skeptic* maps argumentative features of the text to methods for critical assessment, such as the critical questions of argument schemes, ways of evaluating different types of propositions, and signs of possible biased reasoning. The argumentative features are retrieved by utilising extant techniques for argument mining and classification.

Keywords. argument mining, critical literacy, fact-checking, fake news, reason-checking

While deliberate misinformation, disinformation, and deception are by no means new societal phenomena, the recent rise of fake news [1] and information silos [2] has become a growing international concern, with politicians, governments and media organisations regularly lamenting the issue. Efforts to combat such disinformation dressed up as genuine news focus too often exclusively on the factual correctness of the claims made. Whilst the truth of purported facts is clearly of crucial importance, there are other, often overlooked, aspects to consider here. It is, after all, very possible to argue from true factual statements to blatantly false or misleading implications by applying skewed, biased, or otherwise defective reasoning. Furthermore, the categorical corrections on factual impropriety delivered by fact-checkers can both alienate readers who believe they are being told what to think and raise questions around the impartiality of the fact-checkers themselves [3]. For these reasons, attention is increasingly turning to the extension of fact-checking to the broader concept of reason-checking: checking not just factual statements, but the full reasoning underpinning the persuasive text [4].

Skeptic is aimed at addressing these concerns by automatically providing pointers for the critical assessment of a persuasive text beyond checking the veracity of factual statements. The software tool is implemented as a web service[2] that takes an input natural language text and returns a ranked list of questions designed to help readers reason-check the argumentation. The questions are meant to be used as pointers, empowering the readers' critical literacy skills, helping them to draw their own conclusions as to

[1] Corresponding Author: School of Science and Engineering (Computing), University of Dundee, Nethergate, Dundee, DD1 4HN, United Kingdom; E-mail: j.visser@dundee.ac.uk.

[2] `http://skeptic.arg.tech`

whether or not they should accept what they are reading. Actively involving the reader in the reason-checking process should help avoid the instinctive enmity engendered by authoritative fact-checks, while simultaneously broadening the critical spectrum.

The web service maps argumentative features of the persuasive text to methods for critical assessment, such as the critical questions of argument schemes, ways of evaluating different types of propositions, and signs of possible biased reasoning. We employ a pipeline of extant argument technologies [5], all developed to work with the AIF ontology [6], using JSON as a common file type to facilitate handover between the different pipeline components. The combined argument mining and classification techniques provide a reconstruction of the argumentative features of the text, such as the structure of the argumentation, the proposition types of premises and conclusions, and the argument schemes instantiated in the text. These features are then mapped to potential areas of concern, which the *Skeptic* web service returns as a ranked list of prompts for readers to investigate further.

Looking at the overall argumentation structure allows us to identify potential areas of bias where only one side of an argument is being exposed. The argumentation structure also allows us to identify the most central propositions in an argument. These are then classified into one of three proposition types: statements of fact, value, or policy [7]. This classification results in a powerful expansion upon mere fact-checking by broadening the range of proposition types to be checked. Where factual statements can be checked for veracity, policy statements could be checked for consistency or appropriateness, while value statements could be checked for, e.g., popularity. Finally, identified instances of argument schemes are mapped to their associated critical questions [8].

By combining the identification of argumentative features and mapping these to potential flaws in the reasoning, the software allows the user to enter a piece of text and receive a ranked list of questions that they may wish to consider further. The developed software offers a range of potential applications in, for instance, critical literacy education, tools to improve persuasive writing, and the identification of misinformation and fake news.

References

[1] Lazer DMJ, Baum MA, Benkler Y, Berinsky AJ, Greenhill KM, Menczer F, et al. The science of fake news. Science. 2018;359(6380):1094-6. Available from: http://science.sciencemag.org/content/359/6380/1094.

[2] Flaxman S, Goel S, Rao JM. Filter Bubbles, Echo Chambers, and Online News Consumption. Public Opinion Quarterly. 2016;80:298-320.

[3] Dotson T. Fact-checking may be important, but it won't help Americans learn to disagree better; 2022. Accessed: 2-5-2022. https://bit.ly/38OhsVu.

[4] Visser J, Lawrence J, Reed C. Reason-Checking Fake News. Communications of the ACM. 2020;63(11):38-40.

[5] Snaith M, Devereux J, Lawrence J, Reed C. Pipelining Argumentation Technologies. In: Baroni P, Cerutti F, Giacomin M, Simari G, editors. Proceedings of the 3rd International Conference on Computational Models of Argument (COMMA 2010). IOS Press; 2010. p. 447-54.

[6] Chesñevar C, McGinnis J, Modgil S, Rahwan I, Reed C, Simari G, et al. Towards an argument interchange format. The Knowledge Engineering Review. 2006;21(04):293-316.

[7] Visser J, Lawrence J, Reed C, Wagemans J, Walton D. Annotating argument schemes. Argumentation. 2020;35:101–139. Available from: https://doi.org/10.1007/s10503-020-09519-x.

[8] Walton D, Reed C, Macagno F. Argumentation Schemes. Cambridge University Press; 2008.

Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220181

ACH-Nav: Argument Navigation Using Techniques for Intelligence Analysis

Dimitra ZOGRAFISTOU [a,1], Jacky VISSER [a], John LAWRENCE [a] and Chris REED [a]
[a] *Centre for Argument Technology, University of Dundee, UK*

Keywords. Argumentation, ACH, AIF, Hypotheses, Argument Mapping, Navigation

Structured analytic techniques have been established as a powerful weapon in the arsenal of Intelligence Analysis that helps mitigating confirmation bias - arguably one of the most well-known yet still most pernicious cognitive biases [1], by offering systematic processes to conduct the analyses. One of the most widely implemented techniques is the Analysis of Competing Hypotheses (ACH) [2]. As with all structured analytical techniques, ACH is simple in its core idea: first, generate hypotheses that explain some particular event or series of events (there are other structured techniques for such generation); tabulate the working hypotheses; then cross those with the different items of evidence and assumption; finally for each cell in the table – each combination of hypothesis and evidence – analyse the extent to which the evidence is consistent with the hypothesis.

Argumentation is another approach that has increasingly gained traction in the Intelligence Community. Argument mapping, for instance, is a common Structured Analytical Technique [3], and abstract argumentation frameworks [4] have recently been explored as a way of identifying the most valuable or critical items in intelligence analyses [5]. An argument graph of a full-blown intelligence case can, however, very rapidly expand to hundreds of nodes, making it difficult for an analyst to track or make sense of.

To address this challenge, we develop a software tool, called *ACH-Nav* [2], which is an argument visualisation and navigation tool designed specifically to support decision-making and sense-making into large volumes of data in the domain of Intelligence Analysis. The tool offers a navigation framework which is built around the concepts and reasoning of ACH, making the navigation process directly understandable to intelligence analysts, by virtue of their familiarity with the method. It is based on the ArgNav tool[3], the goal of which was to provide the capability to navigate within large volumes of argument structures but in the general domain [6].

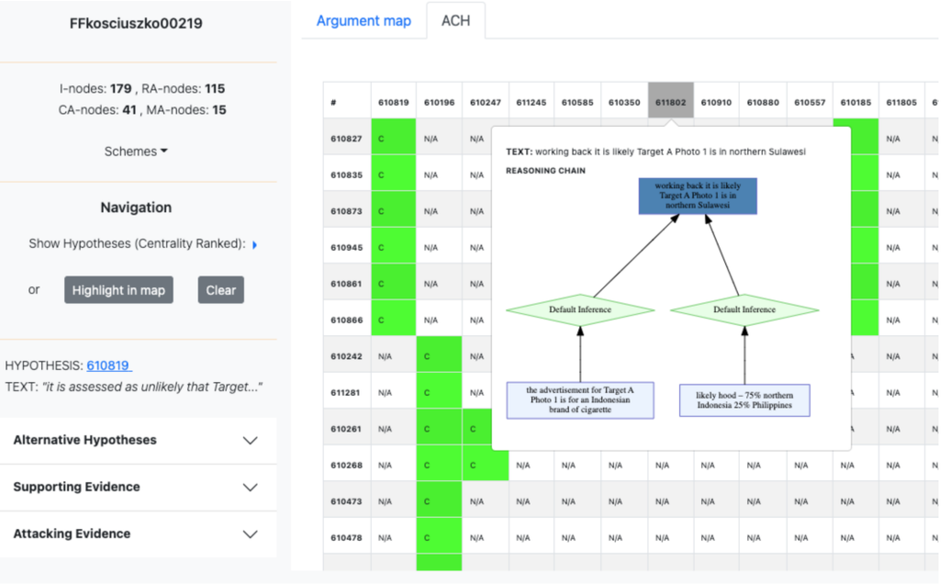
For an ACH-driven argument navigation, central concept is *hypothesis*. In the argument graph, the user can easily identify which propositional nodes are hypotheses and is provided with options to unfold related information, including consistent and inconsistent evidence or alternative hypotheses, by identifying the corresponding structural patterns around them. From these structures, the equivalent ACH matrix can be reconstructed. Thus, the main interface of ACH-Nav consists of two main views of the data,

[1] Corresponding Author: Dimitra Zografistou, Centre for Argument Technology, University of Dundee, UK; E-mail: dzografistou001@dundee.ac.uk

[2] Website: http://achnav.arg.tech/ ; Github repository: https://github.com/arg-tech/ACH-ArgNav

[3] https://argnav.arg.tech/

the *Argument map view* and the *ACH view* and gives the options to switch between them while maintaining focus on specific nodes. Additionally, from the ACH view, the tool allows to instantly uncover the reasoning chain that leads to a hypothesis, by hovering over the corresponding cell that includes this hypothesis. All this functionality is built on the mapping between ACH and argumentation expressed in the Argument Interchange Format (AIF) [7]. Figure 1 gives a screenshot of the ACH view.



FFkosciuszko00219

I-nodes: **179** , RA-nodes: **115**
CA-nodes: **41** , MA-nodes: **15**

HYPOTHESIS: 610819
TEXT: *"it is assessed as unlikely that Target..."*

#	610819	610196	610247	611245	610585	610350	611802	610910	610880	610557	610185	611805	6
610827	C	N/A	N/A									N/A	N
610835	C	N/A	N/A									N/A	N
610873	C	N/A	N/A									N/A	N
610945	C	N/A	N/A									N/A	N
610861	C	N/A	N/A									N/A	N
610866	C	N/A	N/A									N/A	N
610242	N/A	C	N/A									N/A	N
611281	N/A	C	N/A									N/A	N
610261	N/A	C	C									N/A	N
610268	N/A	C	C	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N
610473	N/A	C	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N
610478	N/A	C	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N

TEXT: working back it is likely Target A Photo 1 is in northern Sulawesi

REASONING CHAIN

working back it is likely Target A Photo 1 is in northern Sulawesi
Default Inference
Default Inference
the advertisement for Target A Photo 1 is for an Indonesian brand of cigarette
likely hood – 75% northern Indonesia 25% Philippines

Figure 1. ACH-Nav: ACH view of the data

Acknowledgements

The work described in this paper was supported in part by Dstl under the Serapis framework (SUBCT/FNC-UOD/SERAPIS/58011/EA).

References

[1] Nickerson RS. Confirmation bias: A ubiquitous phenomenon in many guises. Review of General Psychology. 1998;2:175–220.

[2] Heuer R, Good L, Shrager J, Stefik M, Pirolli P, Card S. ACH: A Tool for Analyzing Competing Hypotheses. Palo Alto Research Center (PARC); 2005.

[3] Pherson RH, Heuer Jr RJ. Structured analytic techniques for intelligence analysis. Cq Press; 2020.

[4] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77(2):321-57.

[5] Robinson T. Value of Information for Argumentation based Intelligence Analysis. arXiv; 2021. Available from: `https://arxiv.org/abs/2102.08180`.

[6] Duthie R, Lawrence J, Reed C, Visser J, Zografistou D. Navigating arguments and hypotheses at scale. In: 8th International Conference on Computational Models of Argument, COMMA 2020. IOS Press; 2020. p. 459-60.

[7] Chesñevar C, McGinnis J, Modgil S, Rahwan I, Reed C, Simari G, et al. Towards an Argument Interchange Format. The Knowledge Engineering Review. 2006;21(04):293-316.

Computational Models of Argument
F. Toni et al. (Eds.)

Subject Index

Computational Models of Argument
F. Toni et al. (Eds.)

Author Index