Computational Models of Argument
P. Baroni et al. (Eds.)

doi:10.3233/978-1-61499-686-6-127

Abstract Dialectical Argumentation Among Close Relatives

ALEXANDER BOCHMAN
Computer Science Department,
Holon Institute of Technology, Israel

Abstract. We establish a uniform modular translation of Abstract Dialectical Frameworks into the formalism of the causal calculus, and discuss the correspondences this translation creates between a number of semantics suggested for ADFs and their causal counterparts.

Keywords. formal argumentation, abstract dialectical frameworks, causal reasoning

1. Introduction

Abstract Dialectical Frameworks (ADFs) have been introduced in [8,7] as an abstract argumentation formalism purported to capture more general forms of argument interaction than just attacks among arguments, which form the basis of the original, Dung's argumentation frameworks. To achieve this, each argument in an ADF is associated with an *acceptance condition*, which is some propositional function determined by arguments that are linked to it. Using such acceptance conditions, ADFs allow to express that arguments may jointly support another argument, or that two arguments may jointly attack a third one, and so on. Dung's argumentation frameworks are recovered in this setting by acceptance condition saying that an argument is accepted if none of its parents is.

The authors of ADFs have repeatedly stressed that they primarily see their formalism not as a knowledge representation tool, but rather as a convenient and conceptually neutral abstraction tool ('argumentation middleware') that is intended to encompass a broad range of more specific argumentation and other nonmonotonic formalisms. On the other hand, [16] has considered ADFs as a particular knowledge representation formalism. In our study also, we will view ADFs as a specific knowledge representation formalism and show its close conceptual connections with the formalism of causal reasoning. This will also help us to single out some of the basic principles behind the construction of ADFs and their semantics, as well as to situate the latter in the range of closely related KR formalisms.

The plan of the paper is as follows. We present first a brief description of the formalism of ADF and the relevant parts of the causal calculus. Then we will establish a simple modular translation of ADFs into the causal calculus, and explore the counterparts of the main semantics introduced for ADFs under this translation. It will be shown, in particular, that the basic operator Γ of ADFs can be significantly enhanced by taking into account disjunctive information. This translation will also suggest a natural generalization of ADFs to a general rule-based formalism that will already subsume Logic Programming.

2. Preliminaries I: Abstract Dialectical Frameworks

An abstract dialectical framework (ADF) is a directed graph whose nodes represent statements or positions which can be accepted or not. The links represent dependencies: the status of a node s only depends on the status of its parents (denoted $par(s)$), that is, the nodes with a direct link to s. In addition, each node s has an associated acceptance condition C_s specifying the exact conditions under which s is accepted. C_s is a function assigning to each subset of $par(s)$ one of the truth values $\mathbf{t}, \mathbf{f}$. Intuitively, if for some $R \subseteq par(s)$ we have $C_s(R) = \mathbf{t}$, then s will be accepted provided the nodes in R are accepted and those in $par(s) \setminus R$ are not accepted.

Definition 1. An abstract dialectical framework is a tuple $D = (S, L, C)$ where

- S is a set of statements (positions, nodes),
- $L \subseteq S \times S$ is a set of links,
- $C = \{C_s\}_{s \in S}$ is a set of total functions $C_s : 2^{par(s)} \to \{\mathbf{t}, \mathbf{f}\}$, one for each statement s. C_s is called acceptance condition of s.

A more ‘logical’ representation of ADFs can be obtained simply by assigning each node s a *classical* propositional formula corresponding to its acceptance condition C_s (see [11]). In this case we can tacitly assume that the acceptance formulas implicitly specify the parents a node depends on. It is then not necessary to give the links L, so an ADF D amounts to a tuple (S, C) where S is a set of statements, and C is a set of propositional formulas, one for each statement from S. The notation $s[C_s]$ is used by the authors to denote the fact that C_s is the acceptance condition of s.

A two-valued interpretation v is a (two-valued) *model* of an ADF (S, C) whenever for all statements $s \in S$ we have $v(s) = v(C_s)$, that is, v maps exactly those statements to true whose acceptance conditions are satisfied under v. This notion of a model provides a natural semantics for ADFs. In addition to this semantics, however, the authors define appropriate generalizations for all the major semantics of Dung’s argumentation frameworks. In [7], all these semantics are defined by generalizing the two-valued interpretations to three-valued ones. All of them are formulated using the basic operator Γ_D over three-valued interpretations that was introduced, in effect, already in [8]. In the formulation of [7], for an ADF D and a three-valued interpretation v, the interpretation $\Gamma_D(v)$ is given by the mapping

$$s \mapsto \prod \{w(C_s) \mid w \in [v]_2\},$$

where $\prod$ is the product operator on interpretations, while $[v]_2$ is the set of all two-valued interpretations that extend v.

For each statement s, the operator Γ_D returns the consensus truth value for its acceptance formula C_s, where the consensus takes into account all possible two-valued interpretations w that extend the input valuation v. If v is two-valued, we get $\Gamma_D(v)(s) = v(C_s)$, so v is a two-valued model for D iff $\Gamma_D(v) = v$. In other words, two-valued models of D are precisely those classical interpretations that are fixed points of Γ_D.

The *grounded model* of an ADF D can now be defined as the least fixpoint of Γ_D. This fixpoint is in general three-valued, and it always exists since the operator Γ_D is monotone in the information ordering $\leq_i$, as shown in [8]. This grounded semantics is viewed by the authors as the greatest possible consensus between all acceptable ways of interpreting the ADF at hand[1].

The operator Γ_D also provides a proper basis for defining admissible, complete and preferred semantics for arbitrary ADFs.

Definition 2. A three-valued interpretation v for an ADF D is

- admissible iff $v \leq_i \Gamma_D(v)$;
- complete iff $\Gamma_D(v) = v$;
- preferred iff it is $\leq_i$-maximal admissible.

As can be shown, the above definitions provide proper generalizations of the corresponding semantics for Dung's argumentation frameworks and, moreover, preserve much of the properties and relations of the latter. Thus, the grounded semantics is always a complete model, and each complete model is admissible. In addition, as it is the case for AFs, all preferred models are complete, the grounded model is the $\leq_i$-least complete model, and the set of all complete models forms a complete meet-semilattice with respect to the information ordering $\leq_i$.

In [8], the standard Dung semantics of stable extensions was generalized only to a restricted type of ADFs called bipolar, but [7] has suggested a new definition that avoids unintended features of the original definition, and covers arbitrary ADFs, not only bipolar ones (see also [16]). This new definition is based on the notion of a *reduct* of an ADF, similar to the Gelfond-Lifschitz transformation of logic programs. We will discuss the representation of the stable semantics in ADFs later in this study.

3. Preliminaries II: Causal Reasoning

The causal calculus has been introduced in [14] as a nonmonotonic formalism purported to serve as a logical basis for reasoning about action and change. This line of research has led to the action description language $C+$, which is based on this calculus [12]. A logical basis of the causal calculus was described in [1], and it has been argued in [2] that this calculus is not necessarily restricted to temporal domains, but has actually a vast potential and representation capabilities for serving as a general-purpose nonmonotonic formalism (see also [3,4,5]).

We will assume in this section that our underlying language is an ordinary classical propositional language with the usual connectives and constants $\{\wedge, \vee, \neg, \rightarrow, \mathbf{t}, \mathbf{f}\}$. $\vDash$ and Th will stand, respectively, for the classical entailment and the associated logical closure operator. We will reserve also the letters $p, g, r, \dots$ for denoting propositional atoms, while $A, B, C, \dots$ will denote arbitrary classical propositions of the language.

By a *causal rule* we will mean an expression of the form $A \Rightarrow B$ (*"A causes B"*), where A and B are propositional formulas. A *causal theory* is a set of causal rules. A causal rule $A \Rightarrow B$ is *determinate*, if B is a literal. A determinate causal theory is a set of determinate causal rules.

[1] We will qualify this claim in what follows.

We will begin with a general notion of production inference which is actually just a slight modification of the input-output logic from [13].

Definition 3. A *production inference relation* is a binary relation $\Rightarrow$ on the set of classical propositions satisfying the following conditions:

(Strengthening) If $A \vDash B$ and $B \Rightarrow C$, then $A \Rightarrow C$;
(Weakening) If $A \Rightarrow B$ and $B \vDash C$, then $A \Rightarrow C$;
(And) If $A \Rightarrow B$ and $A \Rightarrow C$, then $A \Rightarrow B \wedge C$;
(Truth) $\mathbf{t} \Rightarrow \mathbf{t}$;
(Falsity) $\mathbf{f} \Rightarrow \mathbf{f}$.

A characteristic property of production inference is that the reflexivity postulate $A \Rightarrow A$ does not hold for it.

We extend causal rules to rules having arbitrary sets of propositions as premises using the familiar compactness recipe: for any set u of propositions, we define

$$u \Rightarrow A \equiv \bigwedge a \Rightarrow A, \text{ for some finite } a \subseteq u$$

$\mathbb{C}(u)$ will denote the set of propositions caused by u, that is

$$\mathbb{C}(u) = \{A \mid u \Rightarrow A\}$$

As could be expected, the causal operator $\mathbb{C}$ plays much the same role as the usual derivability operator for consequence relations. Note that $\mathbb{C}(u)$ is always a deductively closed set (due to And, Weakening, and Truth). Also, it satisfies monotonicity:

Monotonicity If $u \subseteq v$, then $\mathbb{C}(u) \subseteq \mathbb{C}(v)$.

Actually, due to compactness, $\mathbb{C}$ is not only monotonic, but also a continuous operator. Still, it is not inclusive, that is, $u \subseteq \mathbb{C}(u)$ does not always hold. Also, it is not idempotent, that is, $\mathbb{C}(\mathbb{C}(u))$ can be distinct from $\mathbb{C}(u)$.

3.1. Regular, basic and causal inference

A production inference relation is *regular* if it satisfies the following well-known rule:

(Cut) If $A \Rightarrow B$ and $A \wedge B \Rightarrow C$, then $A \Rightarrow C$.

Cut is one of the basic rules for ordinary consequence relations. In the context of production inference it plays the same role, namely, allows for a reuse of produced propositions as premises in further productions[2]. It corresponds to the following characteristic condition on the causal operator:

$$\mathbb{C}(u \cup \mathbb{C}(u)) \subseteq \mathbb{C}(u).$$

Following [13], a production inference relation will be called *basic* if it satisfies

[2]Such production relations correspond to input-output logics with reusable output in [13].

(Or) If $A \Rightarrow C$ and $B \Rightarrow C$, then $A \vee B \Rightarrow C$.

For basic production inference, the set of propositions caused by a theory u coincides with the set of propositions that are caused by every world containing u:

$$\mathbb{C}(u) = \bigcap\{\mathbb{C}(\alpha) \mid u \subseteq \alpha \;\&\; \alpha \text{ is a world}\}$$

Another important fact about basic production inference is that any causal rule is reducible to a set of *clausal* rules of the form $\bigwedge l_i \Rightarrow \bigvee l_j$, where l_i, l_j are classical literals.

Finally, a production inference relation will be called *causal* if it is both basic and regular.

3.2. General nonmonotonic semantics

Production inference determines a natural nonmonotonic semantics, and provides thereby a logical basis for a particular form of nonmonotonic reasoning.

Definition 4.

- A set u of propositions is an *exact theory* of a production inference relation if it is consistent, and $u = \mathbb{C}(u)$.
- A set u of propositions is an *exact theory of a causal theory* Δ, if it is an exact theory of the least production relation $\Rightarrow_\Delta$ that includes Δ.
- A *general nonmonotonic semantics* of a causal theory is the set of all its exact theories.
- A *causal nonmonotonic semantics* of a causal theory is the set of its exact theories that are worlds (complete deductively closed sets).

An exact theory describes an information state in which every proposition is caused, or *explained*, by other propositions accepted in this state. Accordingly, restricting our universe of discourse to exact theories amounts to imposing a kind of an *explanatory closure assumption*. Namely, it amounts to requiring that any accepted proposition should also have an explanation, or justification, for its acceptance.

The general nonmonotonic semantics is indeed nonmonotonic in the sense that adding new causal rules to a causal theory may lead to a nonmonotonic change of the associated semantics, and thereby to a nonmonotonic change in the derived information. This happens even though the causal rules themselves are monotonic, since they satisfy Strengthening (the Antecedent).

Exact theories are consistent fixed points of the operator $\mathbb{C}$. Since the latter operator is monotonic and continuous, exact theories (and hence the nonmonotonic semantics) always exist. Moreover, there always exists a least exact theory. In addition, the union of any chain of exact theories (with respect to set inclusion) is an exact theory, so any exact theory is included in a maximal such theory.

It has been shown in [2] (using an appropriate strong equivalence theorem) that regular production inference provides an adequate and maximal logical framework for reasoning with general exact theories.

As an interesting application of this result for our present study, it can be shown that the least exact theory of a regular inference relation coincides with the set of propositions that are caused by truth **t**. Thus, we obtain the following

Lemma 1. *The least exact theory of a causal theory Δ coincides with the set of propositions that are provable from Δ using the postulates of regular production inference.*

Finally, it has been shown that the *causal* nonmonotonic semantics, as defined above, is equivalent to the original semantics described in [14]. In addition, as a consequence of the corresponding strong equivalence theorem, it has been shown that the full system of causal inference relations (that is both regular and basic) constitutes an adequate logical basis for reasoning with respect to this semantics.

4. The Causal Representation of ADFs

Now we are going to provide a uniform and modular translation of ADFs into the causal calculus. An essential precondition of this causal representation, however, will consist in transforming the underlying semantic interpretations of ADFs in terms of three-valued models (used, e.g., in [7]) into ordinary classical logical descriptions. This latter transformation will also allow us to clarify to what extent the various semantics suggested for ADFs admit a classical logical reading. In fact, the very possibility of such a classical reformulation stems from the crucial fact that the basic operator Γ of an ADF, described earlier, is defined, ultimately, in terms of ordinary classical interpretations extending a given three-valued one. Nevertheless, our reformulation will also reveal a significant discrepancy between these semantics and their immediate causal counterparts.

4.1. Three-valued interpretations versus classical theories

To begin with, any three-valued interpretation v on the set of statements S can be faithfully encoded using an associated set of literals $[v] = S_0 \cup \neg S_1$ such that $S_0 = \{p \in S \mid v(p) = \mathbf{t}\}$ and $S_1 = \{p \in S \mid v(p) = \mathbf{f}\}$. Moreover, this set of literals generates a unique deductively closed theory $\mathrm{Th}([v])$ that corresponds in this sense to the source three-valued interpretation v. Conversely, let us say that a deductively closed set u is a *literal theory*, if it is a deductive (classical) closure of some set of literals. Then the latter set of literals will correspond to a unique three-valued interpretation v such that $u = \mathrm{Th}([v])$. These simple facts establish a precise bi-directional correspondence between three-valued interpretations and classical literal theories. Moreover, we will see in what follows that the main operator Γ of ADFs will correspond under this reformulation to a 'literal' restriction of the causal operator $\mathbb{C}$ of basic production inference.

4.2. Acceptance conditions as causal rules

As our starting point, we note a striking similarity between the official definition of an ADF and the notion of a *causal model*, used by Judea Pearl in [15].

According to [15, Chapter 7], a causal model is a triple $M = \langle U, V, F \rangle$, where

(i) U is a set of *background* (or *exogenous*) variables.
(ii) V is a set $\{V_1, V_2, \ldots, V_n\}$ of *endogenous* variables that are determined by variables in $U \cup V$.
(iii) F is a set of functions $\{f_1, f_2, \ldots, f_n\}$ such that each f_i is a mapping from $U \cup (V \setminus V_i)$ to V_i, and the entire set, F, forms a mapping from U to V.

Symbolically, F is represented as a set of equations

$$v_i = f_i(pa_i, u_i) \quad i = 1, \ldots, n$$

where pa_i is any realization of the unique minimal set of variables PA_i in $V \backslash \{V_i\}$ (parents) sufficient for representing f_i, and similarly for $U_i \subseteq U$.

In Pearl's account, every instantiation $U = u$ of the exogenous variables determines a particular "causal world" of the causal model. Such worlds stand in one-to-one correspondence with the solutions to the above equations in the ordinary mathematical sense. However, structural equations also encode causal information in their very syntax by treating the variable on the left-hand side of $=$ as the effect and treating those on the right as causes. Accordingly, the equality signs in structural equations convey the asymmetrical relation of "is determined by".

Being restricted to the classical propositional language, Pearl's notion of a causal model can be reduced to the following notion of a Boolean causal model, used in [6]:

Definition 5. Assume that the set of propositional atoms is partitioned into a set of *background* (or *exogenous*) atoms and a finite set of *explainable* (or *endogenous*) atoms.

- A *Boolean structural equation* is an expression of the form $p = F$, where p is an endogenous atom and F is a propositional formula in which p does not appear.
- A *Boolean causal model* is a set of Boolean structural equations $p = F$, one for each endogenous atom p.

As can be seen, the above definition is much similar to the logical reformulation of ADFs, with equations $p = F$ playing essentially the same role as the acceptance conditions $p[F]$. The differences are that only endogenous atoms are determined by their associated conditions in causal models, but on the other hand, there are no restrictions on appearances of atoms on both sides in ADF's acceptance conditions. Furthermore, plain (two-valued) models of ADFs correspond precisely to causal worlds of the causal model, as defined in [6]:

Definition 6. A *solution* (or a *causal world*) of a Boolean causal model M is any propositional interpretation satisfying the equivalences $p \leftrightarrow F$ for all equations $p = F$ in M.

Now, a modular representation of Boolean causal models as causal theories of the causal calculus has been given in [6], and it can now be seamlessly transformed into the following causal representation of ADFs:

Definition 7 (*Causal representation of an ADF*)**.** For any ADF D, Δ_D is the causal theory consisting of the rules

$$F \Rightarrow p \text{ and } \neg F \Rightarrow \neg p$$

for all acceptance conditions $p[F]$ in D.

The above representation is fully modular, and it will be taken as a uniform basis for the correspondences described in this study.

To begin with, based on the correspondence results from [6], we immediately establish

Theorem 2. *The two-valued semantics of an ADF D corresponds precisely to the causal nonmonotonic semantics of* Δ_D*.*

As a consequence, the full system of causal inference provides a precise logical basis for this nonmonotonic semantics.

4.3. General correspondences

Now we are going to show that the above causal representation also survives the transition to three-valued models of ADFs. To this end, however, we will have to retreat from the system of causal inference to a weaker system of basic production inference.

A broader correspondence between various semantics of ADFs and general nonmonotonic semantics of the causal calculus arises from the fact that the operator Γ of an ADF naturally corresponds to a particular causal operator of the associated causal theory.

Let L denote the set of classical literals of the underlying language. We will denote by $\mathbb{C}^L$ the restriction of a causal operator $\mathbb{C}$ to literals, that is, $\mathbb{C}^L(u) = \mathbb{C}(u) \cap L$. As we are going to show, the operator Γ of ADFs corresponds precisely to this 'literal restriction' of the causal operator associated with a basic production inference. As before, $[v]$ will denote the set of literals corresponding to a three-valued interpretation v.

Lemma 3. *For any three-valued interpretation v,*

$$[\Gamma_D(v)] = \mathbb{C}^L_D([v]),$$

where $\mathbb{C}_D$ *is a basic causal operator corresponding to* Δ_D*.*

The above equation has immediate consequences for the broad correspondence between the semantics of ADFs that are defined in terms of the operator Γ_D and natural sets of propositions definable wrt associated causal theory. Thus, we have

Theorem 4. *Complete models of an ADF D correspond precisely to the fixed points of* $\mathbb{C}^L_D$*:*

$$v = \Gamma_D(v) \quad \textit{iff} \quad [v] = \mathbb{C}^L_D([v])$$

As a result, we immediately conclude that preferred models of an ADF correspond to maximal fixpoints of $\mathbb{C}^L_D$ (with respect to set inclusion), while the grounded model corresponds to the least fixpoint of $\mathbb{C}^L_D$.

It turns out, however, that when viewed in a classical logical setting, the restriction of the causal operator to literals inadvertently leads to an information loss. More precisely, though disjunctive formulas can appear in acceptance conditions used by Γ in an ADF, the operator itself records, in effect, only literals that are produced, and thereby disregards all other information that can be obtained from its output. The following example illustrates this.

Example 1. Let us consider the following ADF D:

$$q[p] \qquad r[\neg p] \qquad s[q \vee r]$$

The grounded model of this ADF is empty (all atoms are unknown). However, the associated causal theory Δ_D comprises the following rules:

$$p \Rightarrow q \qquad \neg p \Rightarrow r \qquad q \vee r \Rightarrow s$$
$$\neg p \Rightarrow \neg q \qquad p \Rightarrow \neg r \qquad \neg q \wedge \neg r \Rightarrow \neg s$$

In view of Lemma 1, the least exact theory of $\mathbb{C}_D$ is precisely the set of propositions that are provable from the above theory using the postulates of *causal* inference (since it is both basic and regular). Now, the first two rules imply $\mathbf{t} \Rightarrow q \vee r$ (by Or), and hence $\mathbf{t} \Rightarrow s$ by Cut. Similarly, the fourth and fifth rule imply $\mathbf{t} \Rightarrow \neg q \vee \neg r$. As result, the least exact theory of $\mathbb{C}_D$ is much more informative, namely $\mathrm{Th}(\{q \leftrightarrow \neg r, s\})$.

It can also be seen from the above example that the restriction of exact theories to literals does not necessary produce fixed points of the corresponding literal operator $\mathbb{C}^L$. Still, it can be shown that for any fixpoint of the latter (that is, for any complete model an ADF) there exists a least exact theory that contains it. The latter theory may contain, however, more information than its literal source.

5. Justification Frames, Logic Programs and Generalized ADFs

A revised definition of a stable model has been given in [7], generalized already to arbitrary ADFs. Roughly, a two-valued model v of an ADF D is a *stable model* of D if the set of statements that are true in it coincides with the grounded extension of the reduced ADF D^v obtained from D by replacement of all false statements in v by their truth value in each acceptance formula. As has been shown by the authors, this definition properly generalizes stable extensions of Dung's argumentation frameworks.

It should be noted, however, that from the 'non-abstract' knowledge representation view of ADFs that we pursue in the present study, the above definition of a stable semantics constitutes a certain departure from the original formulation of ADFs that was based on *classical* acceptance conditions. Indeed, the above definition of a stable model implicitly breaks the classical symmetry between positive and negative statements, so the acceptance conditions cannot already be viewed as classical formulas. Instead, they acquire a non-classical reading that is quite familiar from logic programming.

It is well-known that the formalism of ADFs, taken in its original sense, does not capture all the semantic distinctions that are expressible in the language of Logic Programming (see, e.g., [16]). Still, the causal representation of ADFs, described in the preceding section, can also suggest a proper generalization of ADFs that would cover Logic Programming under its various semantics, while still preserving the original classical reading of their acceptance conditions (end even their original two-valued semantics). Due to space limitations, however, we can only be brief here.

To begin with, the causal representation of ADFs, described earlier, transforms them into rule-based causal theories, while the latter constitute, in turn, a very special, 'classical' case of *justification frames*, introduced in [9]. In particular, the justification rules of the latter have the general form $x \leftarrow S$, where x is a literal, while S is a set of literals. In the case of the classical negation, such justification frames correspond precisely to determinate causal theories under basic production inference.

However, the causal rules of the causal calculus have an additional expressivity in that they allow arbitrary classical formulas not only in the bodies, but also in the heads of the rules. It turns out that this expressive capability is already sufficient for representing logic programming rules and their semantics.

A causal representation of logic programming rules under various semantics for the latter has been described in [3]. It was defined for general program rules of the form

$$\mathbf{not}\, d, c \leftarrow a, \mathbf{not}\, b \tag{*}$$

where a, b, c, d are finite sets of atoms.

A general understanding of logic programs presupposes an asymmetric treatment of negative information, which is reflected in viewing the negation **not** as denoting *negation as failure*. This understanding can be formally captured in the causal calculus by accepting the following additional postulate:

(Default Negation) $\neg p \Rightarrow \neg p$, for any propositional atom p.

The above postulate makes negations of propositional atoms self-explainable propositions (or abducibles), so it expresses, in effect, the *Closed World Assumption* (CWA).

Given this postulate, a causal representation of logic programs under the stable semantics is provided by interpreting a program rule (*) as the following causal rule:

$$d, \neg b \Rightarrow \bigwedge a \to \bigvee c$$

This interpretation provides a classical understanding for **not**, so its non-classicality amounts solely to the non-classicality of $\Rightarrow$. Nevertheless, it has been shown that the stable semantics of logic programs corresponds precisely to the causal nonmonotonic semantics of the resulting causal theories, that is, to the exact worlds of the latter. Furthermore, the same causal nonmonotonic semantics has turned out to be appropriate also for logic programs under the *supported semantics*, provided we interpret the program rule (*) differently, namely as the following causal rule:

$$a, d, \neg b \Rightarrow \bigvee c$$

The only difference with the previous stable interpretation amounts to treating positive premises in a as explanations rather than as part of what is explained. Note that a normal program rule $p \leftarrow a, \mathbf{not}\, b$ corresponds under this interpretation to the causal rule

$$a, \neg b \Rightarrow p$$

which can be directly transformed into (part of) an acceptance condition for p in ADFs.

The above considerations and results suggest a natural generalization of an ADF to acceptance conditions of the form $A[B]$, where both A and B are classical formulas. This would supply the Abstract Argumentation Frameworks with further representation capabilities, and thereby even contribute to the original aim of the authors of providing a powerful and widely applicable abstraction tool for Argumentation and Reasoning.

6. Summary, Related Work and Conclusions

It has been shown in this study that Abstract Dialectical Frameworks can be uniformly translated into the causal calculus in a way that creates a broad correspondence between the main semantics for ADFs and their causal counterparts.

Among many other things, the suggested translation can be used for determining the place of ADFs (viewed as a specific KR formalism) in the broad range of formalisms for argumentation and reasoning. Thus, it has been shown in [4] that a great number of key systems for argumentation and nonmonotonic reasoning, including the causal calculus, can be viewed as direct instantiations of the original Dung's argumentation frameworks in different logical languages. Due to the results of the present study, the Abstract Dialectical Frameworks also find their natural place in this larger picture. This topic deserves, however, a separate discussion that goes beyond the scope of the present study. Still, a couple of general comments are in order here.

The field of formal argumentation is abundant with different formalisms, which creates a fertile ground for extensive and rapid development. But there is also a lot of conceptual affinity among these argumentation formalisms, as well as between the latter and the major KR representation languages. It is this affinity that allows us to use many of them for basically the same reasoning tasks. This situation creates, however, an obvious incentive for unification, namely for constructing a general theory of argumentation and reasoning where these formalisms could find their proper and hospitable place.

An algebraic approach to unification of different KR formalisms has been suggested in [10], which describes a general method for deriving approximations of operators associated with particular knowledge representation systems. This approach has been successfully applied to ADFs in [16], which also contains comparisons with Logic Programming.

The above approximation theory can be viewed as a paradigmatic *abstraction approach*, in which a general algebraic formalism is shown to be capable of encompassing many particular KR systems. In contrast, our present study can be seen as an instance of a somewhat more specific *generalization approach*, which aims to single out conceptual principles common to a number of formalisms for argumentation and reasoning[3]. For instance, we take it to be a virtue of the original ADFs that they employ classical descriptions in the acceptance conditions. This makes an ADF a natural extension of classical reasoning (instead of being a replacement for the latter), an extension that incorporates, however, some key features of our commonsense reasoning that go beyond pure logical inference.

There is a number of concepts and features that are pervasive in commonsense reasoning, though they escape a purely logical description. The general field of nonmonotonic reasoning has considerably advanced our understanding of these features, which include concepts like explanation, justification, causation, and even definition. The key notions of the modern formal argumentation theory such as support, defeat and attack also belong to this class. It could even be argued that the main contribution of Dung's abstract argumentation theory has consisted not so much in suggesting a new abstract framework for argumentation, but rather in incorporating these notions as the main conceptual ingredients of argumentation. It is this conceptual advancement that has given

[3]The formal theory of justifications [9] could also be seen as a step in this direction.

the formal argumentation theory its current impetus. Accordingly, a systematic study of these novel features of argumentation should be viewed as one of the principal tasks of argumentation theory in general.

Finally, it is an undeniable fact that all the above mentioned notions are also intimately related, which could be seen as the ultimate reason why there are mutual translations between the associated formalisms, as well as why they are so often interchangeable in specific reasoning and argumentation settings. Accordingly, a large part of the task of studying and clarifying the scope of the main building blocks of argumentation consists in determining the relationships and translations among these diverse concepts (often formulated in entirely different formalisms). The correspondence between acceptance conditions of ADFs and causal rules of the causal calculus, established in this study, should hopefully facilitate this general effort.

References

[1] A. Bochman. A logic for causal reasoning. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 141–146, Acapulco, 2003. Morgan Kaufmann.

[2] A. Bochman. A causal approach to nonmonotonic reasoning. *Artificial Intelligence*, 160:105–143, 2004.

[3] A. Bochman. A causal logic of logic programming. In D. Dubois, C. Welty, and M.-A. Williams, editors, *Proc. Ninth Conference on Principles of Knowledge Representation and Reasoning, KR'04*, pages 427–437, Whistler, 2004.

[4] A. Bochman. Propositional argumentation and causal reasoning. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*, pages 388–393, 2005.

[5] A. Bochman. A causal theory of abduction. *Journal of Logic and Computation*, 17:851–869, 2007.

[6] A. Bochman and V. Lifschitz. Pearl's causality in a logical setting. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1446–1452. AAAI Press, 2015.

[7] G. Brewka, H. Strass, S. Ellmauthaler, J. P. Wallner, and S. Woltran. Abstract dialectical frameworks revisited. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.

[8] G. Brewka and S. Woltran. Abstract dialectical frameworks. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, 2010.

[9] M. Denecker, G. Brewka, and H. Strass. A formal theory of justifications. In *Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, pages 250–264, 2015.

[10] M. Denecker, V. W. Marek, and M. Truszczynski. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Inf. Comput.*, 192(1):84–121, 2004.

[11] S. Ellmauthaler. Abstract Dialectical Frameworks: Properties, Complexity, and Implementation. Master's thesis, Technische Universität Wien, Institut für Informationssysteme, 2012.

[12] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Nonmonotonic causal theories. *Artificial Intelligence*, 153:49–104, 2004.

[13] D. Makinson and L. van der Torre. Input/Output logics. *Journal of Philosophical Logic*, 29:383–408, 2000.

[14] N. McCain and H. Turner. Causal theories of action and change. In *Proceedings AAAI-97*, pages 460–465, 1997.

[15] J. Pearl. *Causality: Models, Reasoninig and Inference*. Cambridge UP, 2000. 2nd ed., 2009.

[16] H. Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell.*, 205:39–70, 2013.