Legal Knowledge and Information Systems
E. Schweighofer (Ed.)

doi:10.3233/FAIA210334

Computing Private International Law

Guido GOVERNATORI [a], Francesco OLIVIERI [a], Antonino ROTOLO [b], Abdul SATTAR [a], Matteo CRISTANI [c]

[a] *Institute for Integrated and Intelligent Systems, Griffith University, Australia*
[b] *Alma AI, University of Bologna, Italy*
[c] *Department of Computer Science, University of Verona, Italy*

Abstract. This paper develops a new comprehensive computational framework for reasoning about private international law that encompasses the reasoning patterns modeled by previous works [3,8,9]. The framework is a multi-modal extension of [10] preserving some nice properties of the original system, including some efficient algorithms to compute the extensions of normative theories representing legal systems.

Keywords. Computational Logic, Private International Law, Defeasible Reasoning

1. Introduction

An increasing attention has been paid in the last few years to the interaction among distinct normative systems with regard to the allocation of jurisdiction and choice-of-law characterising private international law (henceforth, PIL). PIL consists in the body of rules and principles governing the choice of law to be applied when there are conflicts in the domestic law of different countries related to private legal facts and transactions [11]. Of course, this is relevant whenever private individuals exhibit aspects of extraneousness with respect to a specific domestic system, and these aspects refer to the law of other countries. The issue of legal pluralism and the fundamental mechanisms of conflict of laws was consequently been studied through argumentation and logics [3,6,8,9,2]. The focus was maintained on legal dogmatics or at the level of virtual conflicts between legal systems, each considered as potentially competent to rule the case, or, again at the level of conflict among different interpretive solutions: precisely the kind of conflicts that PIL in fact prevents.

Several problems such as the following ones often arise.

Case 1 (Conflict across Legal Systems and Overriding Mandatory Rules) *PIL principles may require to apply foreign law and such provisions can be in conflict with domestic law. However, there could exist a domestic piece of legislation that is considered mandatory. In this way, mandatory rules prevent and override any other rule, including the possible foreign law they identify, which is ex ante seen as incompatible with the domestic legal system and its fundamental goals.*

Case 2 (Public Policy Exception) *The foreign interpretive argument gives an interpretive result whose effects are contrary to the public policy of the domestic legal system.*

Case 3 (Same Interpretive Canons Conflict) *The same interpretive argument gives opposite interpretive results in the foreign and in the domestic legal systems.*

Hence, several formal methods can be used to model how domestic courts should apply and reason about foreign law.

The cases above in fact show that PIL requires to handle two distinct reasoning processes:

- Conflict-detection and conflict-resolution among legal rules belonging to different legal systems;
- Conflict-detection and conflict-resolution among interpretive arguments used in different legal systems.

As discussed by [3,8,9], such processes lead to different logical solutions. However, if seen at a more abstract level, all these approaches—and, in fact, any logical model for PIL, we claim—are based on some common formal intuitions. In this paper we accordingly develop a new comprehensive computational framework for reasoning about PIL that encompasses the reasoning patterns modeled by [3,8,9]: Section 2 discusses some examples; Section 3 accounts for previous literature; Section 4 presents the intuitions behind the proposed logic (Section 5.1) and the algorithms to compute the extensions of normative theories (Section 5.2).

2. Problems and Examples

Methods for conflict of laws may occur when norms of different systems collide, or when interpretive arguments collide when used in distinct systems. Consider the following two examples.

Example 1 *[Contract Law] "An Italian company and a British one make a contract according to which the Italian company has to deliver certain goods. A clause says that the contract is governed by US law. The English company sues the Italian company for breach of contract. The jurisdiction issue, in both English and Italian laws, has to be decided on the basis of the Brussels Convention (on Jurisdiction and the Enforcement of Judgments in Civil and Commercial Matters), which establishes the jurisdiction of the Italian judge. However, the Italian judge has to apply the law chosen by the parties, i.e., US law, on the basis of the Rome Convention (on the Law Applicable to Contractual Obligations)". [3]*

In this example, it is crucial whether a contract is regulated by Italian or US law: the two legal systems lead to different outcomes. As argued by [3], the Italian law tends to limit liability of the diligent defaulting party, while US law is stricter in this regard: in several cases, if Italian law had to be applied, the diligent defaulting party would not pay for damages. On the contrary, under US law damages have to be paid. Here, we have a clear *conflict of norms*.

Example 2 *[Interpretation in PIL] "A woman, Cameroonian citizen, put forward an Italian court a paternity action with respect to her daughter, also Cameroonian citizen, underage at the time, on the basis of Art. 340 Cameroonian Civil Code and Art. 33 Law no. 218/1995. She alleged that the child was born within a relationship she had with* *an Italian citizen, who initially took care of the girl and provided financial support for her, then refusing to recognise the child. The judicial question is thus the recognition of the legitimate paternity in favour of the girl, whose main legal consequence would be to burden the presumed father with the duty to give her due support in the form of maintenance and education. [...]*

*Art. 340, Civil Code of Cameroon, states that the judicial declaration of paternity outside marriage can only be done if the suit is filed within the two years that follow the cessation, either of the cohabitation, or of the participation of the alleged father in the support [*entretien*] and education of the child. At a first glance, it appears crucial to properly interpret the term* entretien *for it represents a condition for lawfully advancing the judicial request of paternity. Different interpretations of this term can be offered in Cameroon's law, and may fit differently within the Italian legal system". [8]*

In this second example, once it is made clear which norm has to be applied (Art. 340, Civil Code of Cameroon), we have still a potential conflict to solve because the Italian judge may interpret such a piece of foreign legislation using different interpretive standards: here, we rather have a *conflict of interpretations*.

3. Background

[3] proposed a formal model of the interaction between legal systems based on the so-called modular argumentation, namely, an argumentation system where reasoning in regard to different legal contexts is managed by separate knowledge bases (modules). As expected, the authors assume the existence of different legal systems $\mathrm{LS}_i, \dots, \mathrm{LS}_z$. Each system LS_i, contains three sets of rules: (a) a set of *choice of jurisdiction rules* $ChJur(\mathrm{LS}_i)$; (b) a set of *choice of competence rules* $ChComp(\mathrm{LS}_i)$; and (c) a set of *choice of law rules* $ChLaw(\mathrm{LS}_i)$.

Since a representation of PIL refers to distinct sets of legal rules, modular argumentation offers itself as an appropriate platform for representing PIL and different national laws as it allows knowledge to be split in separate modules.

Indeed, PIL rules establish, respectively, whether courts of LS_i can decide the case (jurisdiction), what particular court of LS_i can do that (competence), and what set of norms, of LS_i's or of another legal system LS_j, the court should apply (applicable law).

The reasoning mechanism handles such sets of rules. First of all, a court should consider the issue of jurisdiction, thus pointing to a certain system LS_i. Having established jurisdiction for the courts of its legal system LS_i, the court k will have to address competence, i.e., to establish whether k itself, among all courts of LS_i, has the task to decide that case, according to $ChComp(\mathrm{LS}_k)$. Finally, court k should apply $ChLaw(\mathrm{LS}_k)$ in order to establish according to what legal system LS_j (that could possibly be different from LS_i) the case should be decided.

[8,9] proposed instead a Defeasible Logic for reasoning about interpretive arguments or canons. As is well-known, interpretive canons are different doctrinal methods that are employed in legal systems as patterns for constructing arguments aimed at justifying certain interpretations [7]. Examples are the Argument by coherence, according to which a statutory provision should be interpreted in light of the whole statute it is part of, or in light of other statutes it is related to, or Teleological argument, according to which a statutory

provision should be interpreted as applied to a particular case in a way compatible with the purpose that the provision is supposed to achieve

The logical structure of interpretive arguments must be analysed using a rule-based logical system. In particular, interpretation canons are represented by *interpretation rules*, such as the following:

$$s : \mathrm{OBL}\mathsf{I}_s^{\mathrm{LS}_j}(n_2^{\mathrm{LS}_i}, d) \Rightarrow^I \mathsf{I}_c^{\mathrm{LS}_j}(n_1^{\mathrm{LS}_i}, p) \tag{3.1}$$

Rule s states that, if provision n_2 belonging to the legal system LS_i ought to be interpreted in another system LS_j by substantive reasons (I_s) as d, then the interpretive canon to be applied in legal system LS_j for provision n_1 is the interpretation by coherence (I_c), which returns p.

Reasoning about interpretive canons across legal systems thus requires to specify in the formal language to which legal systems legal provisions belong and in which legal system canons are applied. In addition, we need the introduction of meta-rules to reason about interpretation rules; such meta-rules support the derivation of interpretation rules; in other words, the conclusions of meta-rules are interpretation rules, while the the antecedents may include any conditions. Consider, for instance, the following meta-rule:

$$r : (\mathrm{OBL}\mathsf{I}_t^{\mathrm{LS}_i}(n_1^{\mathrm{LS}_i}, p), a \Rightarrow_C (s : \mathrm{OBL}\mathsf{I}_s^{\mathrm{LS}_j}(n_2^{\mathrm{LS}_i}, d) \Rightarrow^I \mathsf{I}_c^{\mathrm{LS}_j}(n_1^{\mathrm{LS}_i}, p)))$$

Meta-rule r states that, if (a) it is obligatory the teleological interpretation (I_t) in legal system LS_i of legal provision n_1 belonging to that system and returning p, and (b) a holds, then the interpretive canon to be applied in legal system LS_j for n_1 is the interpretation by coherence, which returns p as well, but which is conditioned in LS_j by the fact that n_2 in this last system is interpreted by substantive reasons as d. In other words, r allows for importing interpretive results from LS_i into LS_j in regard to the legal provision n_1 in LS_i which can be applied in LS_j.

4. Logical Intuition

If we abstract from the peculiarities of [3,8,9], both approaches share a number of general intuitions. On account of the discussion of previous sections, we argue that any formal system for PIL is expected

- to have a **formal language**
  - ∗ able to encode the *existence of different legal systems* $\mathrm{LS}_i, \ldots, \mathrm{LS}_z$;
  - ∗ with *propositional expressions* representing any piece of information which is parametrised by legal systems; for example, we may write a^{LS_i} to mean that a (an obligation, a contract, an interpretive outcome...) holds in the legal system LS_i;
- to have a **reasoning mechanism** that allows for concluding that, if something holds in some legal system, then something else holds in this or another legal system, or that allows for importing in a given system any piece of information holding in another system; for example, the reasoning mechanism should be based on handling
  - ∗ *rules* such as

$$r : a^{\mathrm{LS}_i} \Rightarrow^{\mathrm{LS}_j} b^{\mathrm{LS}_j}$$

which may represent, e.g., a norm r of LS_j (this is represented by the fact that the arrow is labelled accordingly) stating that if a holds in another legal system LS_i, then b holds in LS_j;

∗ *meta-rules* such as

$$s : p^{\mathrm{LS}_k} \Rightarrow (r : a^{\mathrm{LS}_i} \Rightarrow^{\mathrm{LS}_j} b^{\mathrm{LS}_j})$$

which are meant to reason about, and across legal systems (for this reason, the arrow of s in not labelled by any legal system); meta-rule s means that, if p holds in the legal system LS_k, then we can use norm r in LS_j;

∗ since legal systems can be incompatible, different rules and meta-rules can collide, so we need to establish a priority orderings.

The above list shows in a nutshell the basic requirements for developing a general computational framework for reasoning about PIL. The next section will present the details of it.

5. The Framework

The computational framework for reasoning about PIL we are proposing is based on Defeasible Logic [1], which is is a simple and efficient rule-based non-monotonic formalism that proved to be suitable for the logical modelling of different application areas, including the law (see [5,4]). The logic is extended as informally discussed in the previous section. A first result was offered [10]. Here we extend the machinery to handle more legal systems and all requirements mentioned in Section 4.

5.1. Logic

Let PROP be a set of propositional atoms, $\mathbf{LS} = \{\mathrm{LS}_i, \dots, \mathrm{LS}_z\}$ a finite set of legal systems, and Lab be a set of arbitrary labels (the names of the rules). $\mathrm{BLit} = \mathrm{PROP} \cup \{\neg l \mid l \in \mathrm{PROP}\}$ is the set of *basic literals*. The *complement* of a literal l is denoted by $\sim l$: if l is a positive literal p then $\sim l$ is $\neg p$, and if l is a negative literal $\neg p$ then $\sim l$ is p. Hence, $\mathrm{Lit} = \{l^{\mathrm{LS}} | l \in \mathrm{BLit},\ \mathrm{LS} \in \mathbf{LS}\}$ is the set of *literals*.

The set of rules is made of two sets: standard rules R^S, and meta-rules R^M. A *standard rule* $\beta \in R^S$ is an expression of the type '$\beta : A(\beta) \hookrightarrow^{\mathrm{LS}} C(\beta)$', and consists of: (i) the unique name $\beta \in \mathrm{Lab}$, (ii) the *antecedent* $A(\beta) \subseteq \mathrm{Lit}$, (iii) an *arrow* $\hookrightarrow \in \{\rightarrow, \Rightarrow, \rightsquigarrow\}$ denoting, respectively, a strict rule, a defeasible rule and a defeater, (iv) a legal system LS, (v) its *consequent* $C(\beta) \in \mathrm{Lit}$, a single literal. Hence, the statement "Minors are in Italy persons under the age of 18 years" is formulated through a strict rule (as there is no exception to it), whilst "EU citizens may visit the USA without green card" is instead formalised through a defeasible rule as "During pandemic travels to USA might be prohibited" is a defeater representing an exception to it.

A meta rule is a slightly different concept than a standard rule: (i) standard rules can appear in its antecedent, and (ii) the conclusion itself can be a standard rule. Accordingly, a *meta rule* $\beta \in R^M$ is an expression of the type '$\beta : A(\beta) \hookrightarrow C(\beta)$', and consists of: (i) a unique name $\beta \in \mathrm{Lab}$, (ii) the antecedent $A(\beta)$ is now a finite subset of $\mathrm{Lit} \cup R^S$, (iii) the *arrow* $\hookrightarrow$ with the same meaning as for standard rules, and (iv) its *consequent* $C(\beta) \in \mathrm{Lit} \cup R^S$, that is either a single literal or a standard rule (meta-rules can be used to derive standard rules).

A *defeasible meta-theory* (or simply theory) D is a tuple $(\mathrm{F}, R, >)$, where $R = R^{stand} \cup R^{meta}$ such that $R^{stand} \subseteq R^S$ and $R^{meta} \subseteq R^M$. F is the set of facts, indisputable statements that are considered to be always true, and which can be seen as the inputs for a case. As usual in Defeasible Logic, rules in R can be of three types: *strict rules*, *defeasible rules*, or *defeaters*. Finally, we have the *superiority* $>$ among rules, which is binary and irreflexive, and is used to solve conflicts. The notation $\beta > \gamma$ means $(\beta, \gamma) \in >$.

Some abbreviations. The set of strict rules in R is R_s, and the set of strict and defeasible rules is R_{sd}. $R[X]$ is the rule set with head $X \in \{\mathrm{Lit} \cup R^S\}$. R^{LS} is the set of rules whose arrow is labelled by LS. A *conclusion of D* is either a *tagged literal* or a *tagged label* (for a standard rule), and can have one of the following forms with the standard meanings in Defeasible Logic:

- $\pm\Delta l$ means that $l \in \mathrm{Lit}$ is *definitely provable* (resp. *refuted*, or *non provable*) in D, i.e. there is a definite proof for l (resp. a definite proof does not exist);
- $\pm\Delta^{meta}\alpha$, $\alpha \in R^{stand}$, with same meaning as above;
- $\pm\partial l$ means that l is *defeasibly provable* (resp. *refuted*) in D;
- $\pm\partial^{meta}\alpha$, $\alpha \in R^{stand}$, with the same meaning as above.

The definition of proof is also the standard in DL. Given a defeasible meta-theory D, a proof P of length n in D is a finite sequence $P(1), P(2), \dots, P(n)$ of tagged formulas of the type $+\Delta X$, $-\Delta X$, $+\partial X$, $-\partial X$, where the proof conditions defined in the rest of this section hold. $P(1..n)$ denotes the first n steps of P.

Derivations are based on the notions of a rule being *applicable* or *discarded*.

Definition 1 (Applicability) *Given a defeasible meta-theory* $D = (\mathrm{F}, R, >)$*,* $R = R^{stand} \cup R^{meta}$*, a rule* $\beta \in R$ *is #-applicable,* $\# \in \{\Delta, \partial\}$*, at* $P(n+1)$ *iff*

1. $\forall l \in \mathrm{Lit} \cap A(\beta).\ +\#l \in P(1..n)$,
2. $\forall \alpha \in R^S \cap A(\beta)$ *either (a)* $\alpha \in R^{stand}$*, or (b)* $+\#^{meta}\alpha \in P(1..n)$.

Definition 2 (Discardability) *Given a defeasible meta-theory* $D = (\mathrm{F}, R, >)$*,* $R = R^{stand} \cup R^{meta}$*, a rule* $\beta \in R$ *is #-discarded,* $\# \in \{\Delta, \partial\}$*, at* $P(n+1)$ *iff*

1. $\exists l \in \mathrm{Lit} \cap A(\beta).\ -\#l \in P(1..n)$*, or*
2. $\exists \alpha \in R^S \cap A(\beta)$ *such that (a)* $\alpha \notin R^{stand}$ *and (b)* $-\#^{meta}\alpha \in P(1..n)$

When β is a meta-rule and α is not in R^{stand} (hence α is the conclusion of a meta-rule), then β will stay dormant until a decision on α (of being proved/refuted) is made. The following example is to get acquainted with the concepts introduced.

Example 3 *Let* $D = (F = \{a, b\}, R, \emptyset)$ *be a theory such that*

$$R = \{\alpha : a \Rightarrow \beta; \quad \beta : b, \beta \Rightarrow \zeta; \quad \gamma : c \Rightarrow^{\mathrm{LS}} d; \quad \varphi : \psi \Rightarrow d\}.$$

Here, both α and β are applicable (we will see right below how to prove $+\partial^{meta}\beta$*), whilst* γ *and* φ *are discarded as we cannot prove* $+\partial c$ *nor* $\partial^{meta}\psi$*.*

The language of the logic is designed in such a way that all proof tags for literals are the standard ones for Defeasible Logic, so they are omitted for space reasons [1].

We are finally ready to propose the proof tags to prove (standard) rules.

$+\Delta^{meta}\alpha$: If $P(n+1) = +\Delta^{meta}\alpha$ then
(1) $\alpha \in R^{stand}$, or (2) $\exists \beta \in R_s^{meta}[\alpha]$ s.t. β is Δ-applicable.

A standard rule is strictly proven if either (1) such a rule is in the initial set of standard rules, or (2) there exists an applicale, strict meta-rule for it. Since defeasible rule provability requires to detect and solve conflicts between meta-rules, we need to clarify the meaning of $\sim\alpha$, where α is a standard rule.

Definition 3 (Rule complement) *Let* α *be any rule. Then*

$$\begin{array}{ll}
\beta = \alpha : A(\alpha) \Rightarrow^{\mathrm{LS}} C(\alpha) & \sim\beta = \{\neg\alpha,\ \gamma : A(\alpha) \hookrightarrow^{\mathrm{LS}'} \sim C(\alpha), \gamma \in R_{\mathrm{sd}}\} \\
\beta = \alpha : A(\alpha) \rightarrow^{\mathrm{LS}} C(\alpha) & \sim\beta = \{\neg\alpha,\ \gamma : A(\alpha) \rightarrow^{\mathrm{LS}'} \sim C(\alpha)\} \\
\beta = \alpha : A(\alpha) \rightsquigarrow^{\mathrm{LS}} C(\alpha) & \sim\beta = \{\neg\alpha,\ \gamma : A(\alpha) \hookrightarrow^{\mathrm{LS}'} \sim C(\alpha), \gamma \in R_{\mathrm{sd}}\} \\
\beta = \neg(\alpha : A(\alpha) \hookrightarrow^{\mathrm{LS}} C(\alpha)) & \sim\beta = \{\alpha\}.
\end{array}$$

$+\partial^{meta}\alpha$: If $P(n+1) = +\partial^{meta}\alpha$ then
(1) $+\Delta^{meta}\alpha \in P(1..n)$, or
(2) (1) $-\Delta^{meta}\sim\alpha \in P(1..n)$, and
(2) $\exists\beta \in R^{meta}_{\mathrm{sd}}[(\alpha : a_1, \dots, a_n \hookrightarrow c)]$ s.t.
(3) β is ∂-meta-applicable, and
(4) $\forall\gamma \in R^{meta}[\sim(\zeta : a_1, \dots, a_n \hookrightarrow c)]$, then either
(1) γ is ∂-meta-discarded, or
(2) $\exists\varepsilon \in R^{meta}[(\chi : a_1, \dots, a_n \hookrightarrow c)]$ s.t.
(1) $\chi \in \{\alpha, \zeta\}$, (2) ε is ∂-meta-applicable, and (3) $\varepsilon > \gamma$.

A standard rule α is defeasibly proven if it has previously strictly proven (1), or (2.1) the opposite is not strictly proven and (2.2-2.3) there exists an applicable (defeasible or strict) meta-rule β such that every meta-rule γ for $\sim\zeta$ ($A(\alpha) = A(\zeta)$ and $C(\alpha) = C(\zeta)$) either (2.4.1) γ is discarded, or defeated (2.4.2.3) by (2.4.2.1-2.4.2.2) an applicable meta-rule for the same conclusion c. Note that in Condition 2.3 we do not impose that $\alpha \equiv \zeta$, whilst for γ-rules we do impose that the label of the rule in $C(\gamma)$ is either α or ζ.

The condition for $-\partial^{meta}$ is omitted for space reasons, since it is simply obtained from the positive case. Given a defeasible meta-theory D, we define the set of positive and negative conclusions of D as its *meta-extension*:

$$E(D) = (+\Delta, -\Delta, +\Delta^{meta}, -\Delta^{meta}, +\partial, -\partial, +\partial^{meta}, -\partial^{meta}),$$

where $\pm\# = \{l \mid l \text{ appears in } D \text{ and } D \vdash \pm\# l\}$ and $\pm\#^{meta} = \{\alpha \in R^S \mid \alpha$ appears as consequent of a meta-rule β and $D \vdash \pm\#^{meta}\alpha\}$, $\# \in \{\Delta, \partial\}$.

Example 4 *Let* $D = (\mathrm{F} = \{a, c, d, g\}, R, >= \{(\beta, \gamma)(\zeta, \eta)\})$ *be a theory where*

$$R^{stand} = \{\alpha : a \Rightarrow^{\mathrm{LS}_1} b, \qquad \zeta : g \Rightarrow^{\mathrm{LS}_2} \sim b\},$$

$$R^{meta} = \{\beta : c, (\alpha : a \Rightarrow^{\mathrm{LS}_1} b) \Rightarrow (\eta : d \Rightarrow^{\mathrm{LS}_3} b), \qquad \gamma : d \Rightarrow \sim(\chi : d \Rightarrow^{\mathrm{LS}_4} b)\}.$$

As a, c, d *and* g *are facts, we strictly and defeasibly prove all of them. Hence,* α, ζ, β *and* γ *are all* ∂*-applicable. As before,* $\alpha \in R^{stand}$*, thus* $D \vdash +\Delta^{meta}\alpha$ *and* $D \vdash +\partial c$ *make* β *being* ∂*-applicable as well. As* $\beta > \gamma$*, we conclude that* $D \vdash +\partial^{meta}\eta$*, but we prove also* $D \vdash -\partial^{meta}\chi$*. Again,* d *being a fact makes* η *to be* ∂*-applicable.* ζ *has been dormant so far, but it can now be confronted with* η*: since* η *is weaker than* ζ*, then* $D \vdash +\partial\sim b$ *(and naturally* $D \vdash -\partial b$*).*

5.2. Algorithms

The algorithms presented in this section compute the meta-extension of a defeasible meta-theory. The main idea being to compute, at each iteration step, a *simpler* theory than the one at the previous step. By simpler, we mean that, by proving and disproving literals and standard rules, we can progressively simplify the rules of the theory itself.

Let us consider the case of meta-rules. A meta-rule is applicable when each standard rule in its antecedent is either in the initial set of rules (i.e., in R^{stand}), or has been proved later on during the computation and then added to the set of standard rules. This is the reason for the support sets at Lines 1 and 2: R_{appl} is the rule set of the initial standard rules, $R^{\alpha C}$ is the set of standard rules which are not in the initial set but are instead conclusions of meta-rules. As rules in $R^{\alpha C}$ are proved/disproved during the algorithms' execution, both these sets are updated.

At Line 3, we populate the Herbrand Base (HB), which consists of all literals that appear in the antecedent, or as a conclusion of a rule. As literals not in the Herbrand base do not have any standard rule supporting them, such literals are already disproved (Line 4). For every literal in HB, we create the support set of the rules supporting that particular conclusion (Line 6), and we initialise the relative set used later on to manage conflicts and team defeater (Line 7).

We need to do the same for those labels for standard rules that are conclusions of a meta-rule. First, if a label for standard rule is neither in the initial set of standard rules, nor a conclusion of a meta-rules, then such a rule is disproved (Line 8). We assume such sets to have empty intersection, as previously motivated. Second, the following loop at Lines 17–20 initialises three support sets: $R[\alpha]$ contains the meta-rules whose conclusion is α, $R[\alpha]_{opp}$ contains the meta-rules attacking α (γ-like rules in $\pm\partial^{meta}$), while $R[\alpha]_{supp}$ contains the meta-rules supporting α (ε-like rules in $\pm\partial^{meta}$).

The following **for** loop takes care of the factual literals, as they are proved without any further computation. We assume the set of facts to be consistent. Analogously, loop at Lines 17–20 does the same for rules in the initial set of standard rules that may appear in the antecedent of meta-rules.

The algorithm now enters the main cycle (**Repeat-Until**, Lines 21–40). For every literal l in HB (Lines 23–29), we first verify whether there is a rule supporting it, and, if not, we refute l (Line 24). Otherwise, if there exists an applicable rule β supporting it (**if** at Line 25), we update the set of *defeated* rules supporting the opposite conclusion $R[{\sim}l]_{infd}$ (Line 26). Given that $R[{\sim}l]$ contains the γ rules supporting ${\sim}l$, and given that we have just verified that β for l is applicable, we store in $R[{\sim}l]_{infd}$ all those γs defeated by β. The next step is to check whether there actually exists any rule supporting ${\sim}l$ stronger than β: if not, ${\sim}l$ can be refuted (Line 27).

The idea behind the **if** at Lines 28–29 is the following: if $D \vdash +\partial l$, eventually the **repeat-until** cycle will have added to $R[{\sim}l]_{infd}$ enough rules to defeat all (applicable) supports for ${\sim}l$. We thus invoke **Prove** on l, and **Refute** on ${\sim}l$.

Similarly, when we prove a rule instead of a literal, but we now use $R[\alpha]_{opp}$ and $R[\alpha]_{supp}$ in a slightly different way than $R[l]_{infd}$, to reflect the differences between $+\partial$ and $+\partial^{meta}$. Every time, a meta-rule β for α is applicable (**if** at Lines 34–38), we remove from $R[\alpha]_{opp}$ all the γs defeated by β itself (Line 35). If now there are enough applicable ε rules supporting α (**if** check at Line 36), then: (i) we prove α, and (ii) we refute all ζ rules conclusion of γ rules in $R[\alpha]_{opp}$.

Input: Defeasible meta-theory $D = (\mathrm{F}, R, >)$, $R = R^{stand} \cup R^{meta}$
Output: The defeasible meta-extension $E(D)$ of D

$\pm\partial \leftarrow \emptyset$; $\pm\partial^{meta} \leftarrow \emptyset$; $R_{appl} \leftarrow R^{stand}$
$R^{\alpha C} \leftarrow \{\alpha \in R^S \mid \exists\beta \in R^{meta}.\, C(\beta) = \alpha\}$
$HB = \{l \in \mathrm{Lit} \mid \exists\beta \in R^{stand}.\, l \in A(\beta) \cup C(\beta)\} \cup \{l \in \mathrm{Lit} \mid \exists\beta \in R^{meta}.\exists\alpha \in R^S\, (\alpha \in A(\beta) \cup C(\beta)) \wedge (l \in A(\alpha) \cup C(\alpha))\}$
for $l \in \mathrm{Lit} \wedge l \notin HB$ **do** $-\partial \leftarrow -\partial \cup \{l\}$;
for $l \in HB$ **do**
  $R[l] = \{\beta \in R^S \mid C(\beta) = l \wedge (\beta \in R^{stand} \vee \exists\gamma \in R^{meta}.\beta \in C(\gamma))\}$
  $R[l]_{infd} \leftarrow \emptyset$
for $\alpha \notin R^{stand} \cup R^{\alpha C}$ **do** $-\partial^{meta} \leftarrow -\partial^{meta} \cup \{\alpha\}$;
for $(\alpha : A(\alpha) \hookrightarrow C(\alpha)) \in R^{\alpha C}$ **do**
  $R[\alpha] \leftarrow \{\beta \in R^{meta} \mid \alpha = C(\beta)\}$
  $R[\alpha]_{opp} \leftarrow \{\gamma \in R^{meta} \mid C(\gamma) = \sim(\zeta : A(\alpha) \hookrightarrow C(\alpha))\}$
  $R[\alpha]_{supp} \leftarrow \{\varepsilon \in R^{meta} \mid (C(\varepsilon) = (\chi : A(\alpha) \hookrightarrow C(\alpha))) \wedge (\exists\gamma \in R[\alpha]_{opp}.\, \varepsilon > \gamma) \wedge (\chi = \alpha \vee (\exists\gamma \in R[\alpha]_{opp}.C(\gamma) = \sim(\zeta : A(\alpha) \hookrightarrow C(\alpha)) \wedge \chi = \zeta))\}$
for $l \in \mathrm{F}$ **do**
  $+\partial \leftarrow +\partial \cup \{l\}$
  $R \leftarrow \{A(\beta) \setminus \{l\} \hookrightarrow C(\beta) \mid \beta \in R\} \setminus \{\beta \in R \mid \sim l \in A(\beta)\}$
  $> \leftarrow > \setminus \{(\beta, \gamma), (\gamma, \beta) \in >\ \mid \sim l \in A(\beta)\}$
for $\alpha \in R^{stand}$ **do**
  $+\partial^{meta} \leftarrow +\partial^{meta} \cup \{\alpha\}$
  $R^{meta} \leftarrow \{A(\beta) \setminus \{\alpha\} \hookrightarrow C(\beta) \mid \beta \in R^{meta}\} \setminus \{\gamma \in R^{meta} \mid \{\sim\alpha\} \in A(\gamma)\}$
  $> \leftarrow > \setminus \{(\beta, \gamma), (\gamma, \beta) \in >\ \mid \{\sim\alpha\} \in A(\beta)\}$
repeat
  $\partial^{\pm} \leftarrow \emptyset$
  for $l \in HB$ **do**
    if $R[l] = \emptyset$ **then** REFUTE(l);
    if $\exists\beta \in R[l].\, A(\beta) = \emptyset$ **then**
      $R[\sim l]_{infd} \leftarrow R[\sim l]_{infd} \cup \{\gamma \in R[\sim l] \mid \beta > \gamma\}$
      if $\{\gamma \in R[\sim l] \mid \gamma > \beta\} = \emptyset$ **then** REFUTE($\sim l$);
      if $R[\sim l] \setminus R[\sim l]_{infd} = \emptyset$ **then**
        PROVE(l); REFUTE($\sim l$)
  $\pm\partial \leftarrow \pm\partial \cup \partial^{\pm}$
  $\pm\partial_{meta} \leftarrow \emptyset$
  for $(\alpha : A(\alpha) \hookrightarrow C(\alpha)) \in R^{\alpha C}$ **do**
    if $R[\alpha] = \emptyset$ **then** REFUTE(α);
    if $\exists\beta \in R[\alpha].\, A(\beta) = \emptyset$ **then**
      $R[\alpha]_{opp} \leftarrow R[\alpha]_{opp} \setminus \{\gamma \in R^{meta} \mid \beta > \gamma\}$
      if $(R[\alpha]_{opp} \setminus \{\gamma \in R[\alpha]_{opp} \mid \varepsilon \in R[\alpha]_{supp} \wedge A(\varepsilon) = \emptyset \wedge \varepsilon > \gamma\}) = \emptyset$ **then**
        PROVE(α)
        for $\gamma \in R[\alpha]_{opp}.\, C(\gamma) = \sim(\zeta)$ **do** REFUTE($\sim\zeta$);
  $\pm\partial^{meta} \leftarrow \pm\partial^{meta} \cup \partial^{\pm}_{meta}$
until $\partial^{+} = \emptyset$ *and* $\partial^{-} = \emptyset$ *and* $\partial^{+}_{meta} = \emptyset$ *and* $\partial^{-}_{meta} = \emptyset$;
return $E(D) = (+\partial, -\partial, +\partial^{meta}, -\partial^{meta})$

Algorithm 1. Existence

Procedures Prove and Refute are the same as in [10] and are invoked when a literal or a standard rule is proved/refuted.

In order to discuss termination and computational complexity, we start by defining the *size* of a meta-theory D as $\Sigma(D)$ to be the number of the occurrences of literals plus the number of occurrences of rules plus 1 for every tuple in the superiority relation. Thus, the theory $D = (\mathrm{F}, R, >)$ such that $\mathrm{F} = \{a, b, c\}$, $R^{stand} = \{(\alpha : a \Rightarrow^{\mathrm{LS}_1} d), (\beta : b \Rightarrow^{\mathrm{LS}_2} {\sim}d)\}$, $R^{meta} = \{(\gamma : c \Rightarrow (\zeta : a \Rightarrow^{\mathrm{LS}_3} d))\}$, $>= \{(\zeta, \beta)\}$, has size $3 + 6 + 5 + 1 = 15$.

Note that, by implementing hash tables with pointers to rules where a given literal occurs, each rule can be accessed in constant time. We also implement hash tables for the tuples of the superiority relation where a given rule appears as either of the two element, and even those can be accessed in constant time.

Theorem 1 *Algorithm 1 Existence terminates and its complexity is* $O(\Sigma^2)$.

6. Summary

This paper presented a new computational framework for reasoning about PIL. The system in abstracts from the peculiarities of approaches such as [3,8,9]. The formal language assumes the existence existence of different legal systems and of propositional expressions such as a^{LS_i} to mean that a holds in the legal system LS_i. Also, the reasoning mechanism, through meta-rules, allows for concluding that, if something holds in some legal system, then something else holds in this or another legal system, or that allows for importing in a given system any piece of information holding in another system. Finally, since legal systems can be incompatible, different rules and meta-rules can collide, so we make use of priority orderings among rules as in standard Defeasible Logic. The resulting system simply extends [10] and preserves the same nice computational properties.

References

[1] G. Antoniou, D. Billington, G. Governatori, and M. J. Maher. Representation results for defeasible logic. *ACM Trans. Comput. Log.*, 2(2):255–287, 2001.

[2] M. Cristani, F. Olivieri, and A. Rotolo. Changes to temporary norms. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law, ICAIL 2017, London, United Kingdom, June 12-16, 2017*, pages 39–48, 2017.

[3] P. M. Dung and G. Sartor. The modular logic of private international law. *Artif. Intell. Law*, 19(2-3):233–261, 2011.

[4] G. Governatori, F. Olivieri, A. Rotolo, and S. Scannapieco. Computing strong and weak permissions in defeasible logic. *J. Philos. Log.*, 42(6):799–829, 2013.

[5] G. Governatori and A. Rotolo. Changing legal systems: legal abrogations and annulments in defeasible logic. *Log. J. IGPL*, 18(1):157–194, 2010.

[6] J. Hage. Logical tools for legal pluralism. *Maastricht European Private Law Institute Working Paper 7*, 2015.

[7] D. MacCormick and R. Summers, editors. *Interpreting Statutes: A Comparative Study*. Ashgate, 1991.

[8] A. Malerba, A. Rotolo, and G. Governatori. Interpretation across legal systems. In *Proc. JURIX 2016*. IOS Press, 2016.

[9] A. Malerba, A. Rotolo, and G. Governatori. A logic for the interpretation of private international law. In *New Developments in Legal Reasoning and Logic*. Springer, Dordrecht, 2021.

[10] F. Olivieri, G. Governatori, M. Cristani, and A. Sattar. Computing defeasible meta-logic. In *Proc. JELIA 2021*. Springer, 2021.

[11] P. Stone. *EU Private International Law*. Elgar European law. Edward Elgar, 2014.