Legal Knowledge and Information Systems
M. Araszkiewicz and V. Rodríguez-Doncel (Eds.)

doi:10.3233/FAIA190320

Defeasible Systems in Legal Reasoning: A Comparative Assessment

Roberta CALEGARI [a,1], Giuseppe CONTISSA [b], Francesca LAGIOIA [b,c], Andrea OMICINI [a], and Giovanni SARTOR [b,c]

[a] *Dipartimento di Informatica – Scienza e Ingegneria (DISI),* ALMA MATER STUDIORUM–*Università di Bologna, Italy*
[b] *CIRSFID-Faculty of Law,* ALMA MATER STUDIORUM–*Università di Bologna, Italy*
[c] *European University Institute, Law Department, Italy*

Abstract. Different formalisms for defeasible reasoning have been used to represent legal knowledge and to reason with it. In this work, we provide an overview of the following logic-based approaches to defeasible reasoning: Defeasible Logic, Answer Set Programming, ABA+, ASPIC+, and DeLP. We compare features of these approaches from three perspectives: the logical model (knowledge representation), the method (computational mechanisms), and the technology (available software).On this basis, we identify and apply criteria for assessing their suitability for legal applications. We discuss the different approaches through a legal running example.

Keywords. AI and Law, legal reasoning, defeasible reasoning, argumentation

1. Introduction

Different approaches have been adopted to deal with defeasibility in law, including argumentation frameworks, which capture defeasibility through the interaction of conflicting arguments [1]. Even though defeasibility is a key aspect of legal reasoning, no comparative analysis of existing approaches has been carried out so far including their features, the available software tools, and more generally the advantages and disadvantages offered by different legal applications.

The present work aims to make a first step in this direction by pursuing two main goals. The first is to provide an assessment of some existing formalisms which may be useful in supporting informed choices by developers. The second is to identify some general methodological guidelines and criteria for determining which which formalisms for defeasible reasoning are more suitable for intended applications. We hope that by providing a framework for the analysis, comparison, and selection of the appropriate computable models of defeasible reasoning, we will contribute to strengthening the link between theory and application, and fostering successful integration. In our contribution, we have taken into account previous works dealing with the comparison of formalisms for defeasible reasoning [2,3], considering a more diverse set of formalisms, and focusing

[1]Corresponding Author: Roberta Calegari E-mail: roberta.calegari@unibo.it.

not only on expressiveness, but also on inference methods and the availability of software tools.

2. Running Example

To highlight the differences and similarities between the selected approaches, let us consider a hypothetical but realistic legal case concerning medical malpractice.

Patient John seeks compensation against Doctor Mary, claiming that Mary caused harm to him, and appeals to a legal rule stating that if a doctor causes harm to a patient, then the doctor has an obligation to pay damages, unless it is proven that the doctor was not negligent. This rule establishes a presumption of negligence against the doctor and a conditioned presumption of non-negligence favouring the doctor —the doctor was careful if he followed medical guidelines. Let us assume that expert evidence is provided by the two parties. In the following we consider different combinations of claims and see the conclusions generated by different approaches.

2.1. Nonmononic reasoners

Defeasible Logic DL is a well-know formalism for defeasible reasoning, originally proposed by Nute [4], and later extended in various directions, including deontic logic. Let us assume that expert witness Mark claims that there was harm, while expert witness Edward claims that the knowledge (the guidelines) was correctly followed.

```
patient('John'). doctor('Mary'). expert('Mark'). expert('Edward').
say('Mark',harmed(doctor('Mary'), patient('John'))). say('Edward',careful(doctor('Mary'))).
liable(doctor(D)) := harmed(doctor(D), patient(P)).
neg liable(doctor(D)) := used_correctly(knowledge, doctor(D)).
harmed(doctor(D), patient(P)) := say(X,harmed(doctor(D), patient(P))), expert(X).
used_correctly(knowledge, doctor(D)) := say(X,careful(doctor(D))), expert(X).
Answer for @liable(doctor('Mary'): no.
```

Running the query *@liable(doctor('Mary')*, as well as *@neg liable('Mary')*, we obtain *false*, since the inferences for liable and not liable defeat one another. Adding a priority for the rule against liablility over the rule for liablilty, we obtain *yes* for Mary's non-liability. Assume now that Marks also intervenes on the issue of compliance with the guidelines, claiming that Mary did not follow the guidelines. The outcome is surprisingly that Mary is liable. In fact, the rule on the exclusion of liability would not be triggered, given that the antecedent *used_correctly(knowledge, doctor('Mary'))* could not be established, given the contradictory claims of the two experts. This aspect of the functioning of DL is called ambiguity blocking: when two conflicting inferences clash and there is no priority, the inferences cancel each other out.

ASP Answer set programming (ASP) is an approach to logic programming oriented towards difficult (primarily NP-hard) search problems. This input yields no results because of the unsolved contradiction between rules one and two. Note that the standard ASP format, used by systems such as Clingo and DLV2, does not support the use of preferences over rules. To express that the rule with conclusion *harmed(doctor(D), patient(P))* applies unless the doctor uses the knowledge correctly, we have to introduce a negation by failure *not used_correctly(knowledge, doctor(D))* in the body of that rule. If the input is so modified, Clingo provides a stable model according to which there is no liability.

```
liable(doctor(D)) :- harmed(doctor(D), patient(P)).
not liable(doctor(D)) :- used_correctly(knowledge, doctor(D)).
harmed(doctor(D), patient(P)) :- say(X,harmed(doctor(D), patient(P))), expert(X).
used_correctly(knowledge, doctor(D)) :- say(X,careful(doctor(D))), expert(X).
patient(john). doctor(mary). expert(mark). expert(edward).
say(mark,harmed(doctor(mary), patient(john))). say(edward,careful(doctor(mary))).
Answer: UNSATISFIABLE
```

2.2. Structured Argumentation

DeLP DeLP is a formalisation of defeasible reasoning in which the results of Defeasible Logic and Argumentation are combined [5]. The behavior is the same as DL, but it allows for ambiguity propagation, i.e., it may develop inferences based on conflicting propositions (as in ASPIC's preferred semantics).

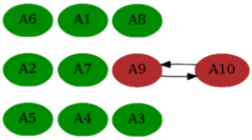
```
Patient(john). Doctor(mary). Expert(mark). Expert(edward).
Say_harmed(mark, mary, john). Say_careful(edward, mary).
Liable(D) -< Harmed(D, P). ~Liable(D) -< Used_correctly(knowledge, D).
Harmed(D,P) -< Say_harmed(X,D, P), Doctor(D), Patient(P), Expert(X).
Used_correctly(knowledge, D) -< Say_careful(X,D), Doctor(D), Expert(X).
```

ASPIC+ ASPIC+ is a popular framework for structured argumentation, exploiting Dung's abstract semantics [6]. ASPIC allows users to choose from different semantics: grounded, preferred, semi-stable, and stable. The preferred semantic is particularly significant for the law, since it shows alternative extensions for unsolved conflicts. The use case is encoded as in the following listing with its corresponding argumentation graph under the grounded semantics, where both arguments A9 and A10 are rejected, since they defeat each other. The assessment changes if we add rule priorities. If we add a preference for rule r2 over rule r1 we find that A9 is now justified, while A10 is rejected. This shows an interesting difference between DL and ASPIC. In DL an unsolved conflict between two inferences means that such inferences (and the inferences expanding them) are irrelevant. In ASPIC the conflicting arguments can still defeat other arguments, and prevent the defeated arguments from being included in all preferred extensions.

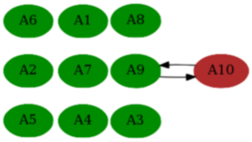
```
Premises:patient(john);doc(mary);exp(mark);exp(edw)
Assumption:say_harm(mark,mary,john);say_careful(edw,mary)
Rules: [r1] harm(D,P) => liable(D);
[r2]used(K,D),doc(D) => ~liable(D);
[r3]say_harm(X,D,P),doc(D),patient(P),exp(X)=>harm(D,P);
[r4]say_careful(X,D),doc(D),exp(X)=>used(kb,D);
```

```
A1:say_careful(edw,mary)
A2:exp(edw) A6:exp(mark)
A3:say_harm(mark,mary,john)
A4: patient(john) A5: doc(mary)
A7: A1,A2,A5=>used(kb,mary)
A8: A3,A4,A5,A6=>harm(mary,john)
A9: A7,A5=>~liable(mary)
A10: A8=>liable(mary)
```

Grounded semantics



Preferred semantics (1)



Preferred semantics (2)

ABA+ In ABA+ arguments are sets of assumptions used to infer conclusions. Each rule has to be ground (i.e., no variables allowed). This is due to the fact that the tool uses a semantics-preserving mapping from ABA+ to abstract argumentation and uses ASPARTIX, for determining extensions. Moreover, ABA+ does not deal with preferences over rules, it only supports preferences over assumptions. The contraries of each assumption must be explicitly declared.

Table 1. Comparison under the *modelling* perspective: what aspects of legal argument can be captured.

	Defeasible Logic	**Argumentation**			**ASP**
		ABA+	ASPIC+	DeLP	
Model	DL Nute	AA Dung	AA Dung	DL Nute	ASP
Rules & Presumption	no argument notion	✓	✓	✓	no argument notion
Defeaters	✓	contraries	undercut, rebut, undermine	✓	✓
Preferences	✓	encoded	✓	✓	✓
Deontic Logic	✓	no	on strict rules	no	no
Argumentation Schemes	no	no	meta-ASPIC	no	no

3. Guidelines for Comparison and Evaluation

3.1. Model Perspective

Logical model and argument structure Even though different approaches to defeasible reasoning share a common background, they often adopt different logical models. DL is based on an inferential semantics, while ASP is based on the stable-set semantics of logic programming. On these approaches understanding an argument means exploring the inference tree derived by the application of the rules. On the other side, argumentation approaches explain their outcome through the attack and defeat relations between the applicable arguments. An advantage of Dung's abstract argumentation systems is the possibility of dealing with different semantics: the alternative between grounded and preferred semantics offers a choice between focalising on "sure" outcomes, or exploring alternatives that depend on possible solutions to rule conflicts.

Strict rules, defeasible rules, and presumptions. DL, as well as DeLP, ASPIC+ and ABA+, provides for the use of both strict and defeasible rules even thought in ABA+ defeasible rules are strict rules plus assumptions. While in some frameworks (ASPIC+ or ABA) assumptions are explicitly introduced, in other frameworks, such as DL, they can be modelled as rules with an empty antecedent.

Defeaters and attacks. Contrarily to other approaches, defeaters in DL and DeLP can be expressed via explicit rules (in the latter in the form of explicit undercutting defeaters, too). In DL and in argumentation-based approaches different types of attacks are distinguished (e.g., undercutting, rebutting, or undermining, while DeLP defines a single general notion of attack.

Preferences. Preferences among rules are supported by all approaches considered (in ABA preferences concern assumptions).

Deontic logic. Deontic modalities have been introduced in various logics to make them more suitable for legal reasoning. DL has been extended to support deontic modalities [7]. Such modalities are not supported natively by any ASP representation. With respect to ASPIC+ extensions to deontic logic have been defined, but have not yet been implemented in any reasoning tool.

Argumentation schemes. Patterns of informal argumentation often occur in real-world decision-making and in discussions between humans. These consideration lead to their formalization into argumentation frameworks, such as the meta-ASPIC model [8]. To the best of our knowledge no implementation is provided.

Table 2. Comparison of legal reasoning approaches from the *method* perspective, i.e., focusing on the reasoning/computational method used for legal inference and argumentation.

	Defeasible Logic	**Argumentation**			**ASP**
		ABA+	ASPIC+	DeLP	
Complexity	Polinomial/ linear	Polinomial			NP
Inconsistency handling (conflicting rules)	ambiguity blocking vs ambiguity propagation	ambiguity propagation →undecided			ambiguity propagation →unsatisfability
Inconsistency handling (conflicting facts)	derive results despite them	derive results despite them			unsatisfability
Credoulous/Skeptical	skeptical	✓	✓	skeptical	✓

3.2. Method Perspective

Complexity. All approaches are efficient in terms of reasoning time. However, acceptability of a proposition in an argumentation framework under grounded semantics can be computed in polynomial time, while defeasible logic, if restricted to propositional logic, has linear complexity. Finally ASP is NP.

Credoulous/skeptical and inconsistency handling: conflicting rules. An important difference among the three approaches is the way they handle inconsistency. DL, originally based on ambiguity blocking, has been "tuned" to obtain ambiguity propagation, i.e., the inferences based on conflicting claims. Outcomes similar to those obtained in ambiguity blocking and ambiguity propagation in DL can be obtained by a grounded or preferred Dung's semantics in ABA+ and ASPIC+, as discussed above. Conflicting facts in ASP lead to unsatisfiability because of the standard definition of consistency. DL as well as argumentation frameworks, on the other hand, handle inconsistencies to deliver defeasible outcomes according to their semantics.

3.3. Technology Perspective

Table 3. Comparison of legal reasoning approaches from the *technology* perspective (availability, accessibility and usability of software resources).

	Defeasible Logic	**Argumentation**			**ASP**
		ABA+	ASPIC+	DeLP	
Technology	d-Prolog/SPINdle	ABAPlus	TOAST	TweetyProject	Clingo/DLV
Open source	✓	✓	✓	✓	✓
IDE KB Support	no	no			no
Contradiction warning	no	no			✓

Technology. In terms of tools for reasoning support, there is at least one stable open source reasoner available for each approach. Sometimes no complete documentation manual is provided, leading to some difficulties in rule transcription.

IDE and Contradiction warning. While a number of reasoning tools have been developed, no tool is currently available to support knowledge encoding, to the best of our knowledge. This means that any legislation has to be manually written in the language supported by the reasoners. All tools lack a form of inconsistency highlighting.

4. Conclusion

Our analysis has shown that there is a strong convergence between different systems for defeasible reasoning. However, some differences exist, which may be relevant to different application domains. The possibility of using open (non-ground) rules in knowledge, and of using different instances of the same predicates in different rules, could be a key advantage, especially when the same rule has to be applied to different instances within a single argument. All the described systems, except for ABA+ and SPINDLE, have this feature. When a system has to deal with a high number of uncertain conflicts, the ability to rely not only on skeptical, but also on credulous reasoning may be important. Argumentation approaches (such as DeLP, ASPIC+, and ABA+) have this ability natively (though also ambiguity propagation in DL can also lead to similar results). When a system has to address complex issues of legal reasoning, and full explainability is required; the ability to provide a picture of existing arguments and of the relations between them, and an explanation on what arguments should or could be finally endorsed, may become a decisive feature. This is a feature we could find in ASPIC+ and ABA. From a technological perspective, many improvements need to be made in order to make existing tools really usable and effective in a distributed environment, as well as, documented and easily downloadable/deployable. The results presented here represent just a preliminary exploration of the logic-based approaches to defeasible reasoning, but it can provide starting guidelines for a methodological comparison of the various approaches.

Acknowledgments. This work has been partially supported by the European Union's Justice programme under Grant Agreement No. 800839 for the project "InterLex: Advisory and Training System for Internet-related private International Law".

References

[1] H. Prakken and G. Sartor, Law and logic: A review from an argumentation perspective, *Artificial Intelligence* **227** (2015), 214–245.

[2] S. Batsakis, G. Baryannis, G. Governatori, I. Tachmazidis and G. Antoniou, Legal Representation and Reasoning in Practice: A Critical Comparison, in: *Legal Knowledge and Information - JURIX 2018*, M. Palmirani, ed., Frontiers in Artificial Intelligence and Applications, IOS Press, United States, 2018, pp. 31–40.

[3] G. Charwat, W. Dvorak, S. Gaggl, J. Wallner and S. Woltran, Methods for solving reasoning problems in abstract argumentation – A survey, *Artificial Intelligence* (2014).

[4] D. Nute, *Defeasible Reasoning: A Philosophical Analysis in Prolog*, in: *Aspects of Artificial Intelligence*, J.H. Fetzer, ed., Springer Netherlands, Dordrecht, 1988, pp. 251–288. ISBN ISBN 978-94-009-2699-8.

[5] A. Garcia and G. Simari, Defeasible logic programming: an argumentative approach, *Theory and Practice of Logic Programming* **4**(2) (2004), 95–138.

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

[7] G. Governatori, A. Rotolo and E. Calardo, Possible World Semantics for Defeasible Deontic Logic, in: *Deontic Logic in Computer Science*, T. Agotnes, J. Broersen and D. Elgesem, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 46–60. ISBN ISBN 978-3-642-31570-1.

[8] J. Muller, A. Hunter and P. Taylor, Meta-level argumentation with argument schemes, in: *International Conference on Scalable Uncertainty Management*, Springer, 2013, pp. 92–105.