September 2016

Demonstrating a benchmark for defeasible reasoning

Abdelraouf HECHAM [a], Madalina CROITORU [a] and Pierre BISQUERT [b]
[a] *GraphIK, INRIA, LIRMM, University of Montpellier, France*
[b] *GraphIK, INRIA, IATE, INRA, France*

Abstract. In this demonstration we focus on the task of a data engineer choosing what tool to perform defeasible reasoning with a first order logic knowledge base.

1. Motivation

Defeasible reasoning allows to reason with incomplete or inconsistent knowledge where conclusions can be challenged by additional information. To reason defeasibly about a conclusion, all possible inferences reaching that conclusion must be evaluated along with any conflict that arises from other potential inferences. This can be achieved either using the idea of extensions, where reasoning chains (arguments) are built then evaluated at a later stage; this encapsulates argumentation-based techniques such as grounded semantics [2] and dialectical trees [3]. Another class of approaches are based on the evaluation of arguments during their construction, such as defeasible logics [6,1].

An inherent characteristic of defeasible reasoning is its systematic reliance on a set of intuitions long debated between logicians: should an information that is derived from a contested claim be used to contest another claim (i.e. ambiguity handling)? Or, can different 'chains' of reasoning for the same claim be combined to defend against challenging statements (i.e. reinstatement)? Existing working implementations (**ASPIC+** [7], **DEFT** [4], **DeLP** [3]) do not make explicit the intuitions followed. For example, it is not stated in the companion papers of previously mentioned tools if they allow or not for ambiguity blocking/propagating, team defeat or floating conclusions. Unless the end user is familiar with the comparison between Defeasible Logics, Grounded Semantics and Dialectical Trees he would not be able to deduce these information.

We built upon a *propositional* defeasible logic *performance-oriented* benchmark from [5] that generates various parameterized knowledge bases (also known as theories). These theories serve two purposes: first, test the tools' ability to handle the features (especially when these features are not explicitly stated by the tools authors). Second, test their performance when faced with gradually complex situations requiring these features. For example: does the tool allow for team defeat? How does it perform when there are larger and larger instances requiring team defeat? In this demonstration abstract we show how the benchmark could be used in a practical scenario.

Please note that the benchmark is available for download at: `https://github.com/hamhec/defeasible-tools-benchmark`, where the interested reader can find more details on how it is implemented and how it can be run.

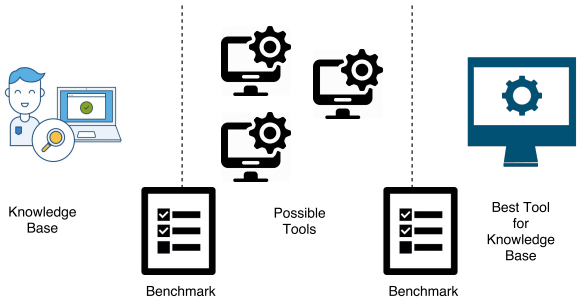

Figure 1. The workflow for using the benchmark for defeasible reasoning.

2. Emergency Response Application Scenario

In a emergency response scenario we need to determine if an accident victim is an organ donor. A person being hurt in an accident is considered a victim. Legally any victim is assumed not to be an organ donor. A person that gives her consent is considered an organ donor. A person in a critical condition generally cannot give her consent. The legal tutor of a person can give his consent for her being an organ donor. After formalising the above mentioned knowledge base using existential rules, let us suppose to be interested in using defeasible reasoning with ambiguity propagation. The formalised knowledge base is acyclic and has no priority relation. At this point the domain engineer verifies the results of the benchmark as instructed in the companion paper describing is at: `https://github.com/anoConf/Benchmark`. Given the results of the benchmark, the data engineer can choose between the three tools: ASPIC+, DeLP or DEFT as they all allow for ambiguity propagation. Since the number of rules in this knowledge base is small, according to the run times of the two tools on the benchmark, it is *recommended to use either DEFT or ASPIC+ tools*. Let us now modify the knowledge base and add the rule that a victim is probably someone who is hurt. In this case the knowledge base becomes cyclic. Therefore *only DEFT and DeLP tools can be used*. Finally, let us add a new rule stating that if someone is an organ donor then that somebody gave his consent (the person or her tutor). In this case we *can only use* DEFT. The demo workflow is represented in Figure 1.

References

[1] D. Billington. Defeasible Logic is Stable. *Journal of logic and computation*, 3(4):379–400, 1993.

[2] 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):321–357, 1995.

[3] A. J. García and G. R. Simari. Defeasible logic programming: An argumentative approach. *Theory and practice of logic programming*, 4(1+ 2):95–138, 2004.

[4] A. Hecham, M. Croitoru, and P. Bisquert. Argumentation-Based Defeasible Reasoning For Existential Rules. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1568—-1569, 2017.

[5] M. J. Maher, A. Rock, G. Antoniou, D. Billington, and T. Miller. Efficient defeasible reasoning systems. *International Journal on Artificial Intelligence Tools*, 10(04):483–501, 2001.

[6] D. Nute. *Defeasible reasoning: a philosophical analysis in prolog*. Springer, 1988.

[7] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.