Computational Models of Argument
S. Modgil et al. (Eds.)

doi:10.3233/978-1-61499-906-5-475

How to Generate a Benchmark of Logical Argumentation Graphs?

Bruno YUN [a], Madalina CROITORU [a] and Srdjan VESIC [b]
[a] *INRIA LIRMM University of Montpellier*
[b] *CNRS CRIL University of Artois*

Abstract. In this demonstration extended abstract we present a workflow of how to generate a benchmark for logical argumentation graphs issued from knowledge bases expressed using existential rules.

Keywords. Logic Based Argumentation, Argumentation Graphs, Benchmarks.

1. Significance and demonstration workflow

The lack of large, practically inspired benchmarks in the argumentation field was acknowledged by the community long time ago, but became obvious with the appearance of the International Competition on Computational Models of Argumentation (ICCMA)[1]. Currently the competition uses randomly generated graphs with certain graph theoretical properties intuitively considered desirable (e.g Nofal et al. [4] and Cerutti et al. [1]). Reasoning methods in argumentation are based on graph theoretical operations, the graph properties of the underlying graph can "make or break" tool performance.

In this demonstration extended abstract we focus on argumentation graphs issued from inconsistent logical knowledge bases. We use the existential rule instantiation of argumentation frameworks from [2] where nodes of these graphs represent all possible arguments one can construct over the knowledge base while the directed edges represent the attacks that model the inconsistency between two arguments. The reason for using existential rules stems from their versatility: they supsume certain subsets of Description Logics and are widely used as an ontological layer over relational databases. Therefore using existential rules knowledge bases as backbones for the argumentation graphs offers the possibility of using any inconsistent existential rule knowledge base developed part of a project or available online. Moreover, the argumentation graphs issued from this instantiation are known to respect certain graph theoretical properties: presence of isolated nodes, the existence of at least one strongly connected component with more than one nodes, presence of repetitive patterns of subgraphs and, last but definetely not least an impressive size. In [7] we show that even for a modest knowledge base composed of 7 facts, 3 rules and 1 binary negative constraint one gets an argumentation graph of 383 arguments and 32768 attacks. Let us stress that in [6] we also provide a complete graph structural characterisation of argumentation graphs constructed as above from knowl-

[1] http://argumentationcompetition.org/

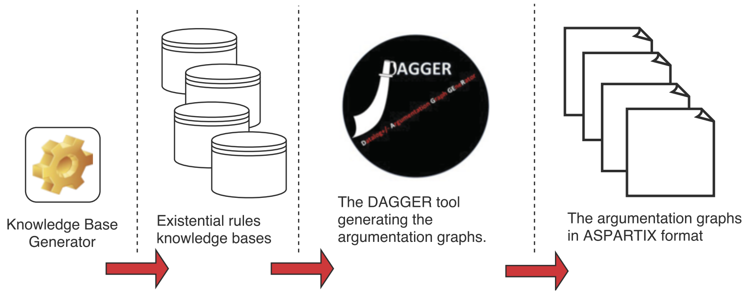
Knowledge Base Generator
Existential rules knowledge bases
DAGGER
The DAGGER tool generating the argumentation graphs.
The argumentation graphs in ASPARTIX format

Figure 1. The workflow for generating a benchmark of argumentation graphs issued from existential rules knowledge bases.

edge bases solely composed of factual knowledge and negative constraints. Using such structural knowledge one can, for instance, design argumentation solvers that, despite the huge size, perform better in the presence of symmetries [3].

This demonstration will show how one can generate a benchmark of argumentation graphs issued from logical knowledge bases expressed using existential rules. The demonstration workflow is visualised in 1. According to the workflow, the following steps need to be followed: first the existential rules knowledge base are generated or loaded. [7] provides a benchmark set of knowledge bases increasing in size with respect to number of facts, rules and negative constraints, as well as variations in how the negative constraints cover the facts. In the next step, for each generated or loaded knowledge base, the argumentation graph is constructed. This is done using the DAGGER (Datalog+/- Argumentation Graph GEneRator) tool demonstrated in [5]. The DAGGER tool takes as input an inconsistent knowledge base expressed using existential rules (in .dglp format) and outputs the corresponding argumentation graph (in aspartix format). If needed the user can also visualise the graph or compute the extensions.

References

[1] F. Cerutti, P. E. Dunne, M. Giacomin, and M. Vallati. Computing Preferred Extensions in Abstract Argumentation: A SAT-Based Approach. In *TAFA 2013*, pages 176–193, 2013.

[2] M. Croitoru and S. Vesic. What Can Argumentation Do for Inconsistent Ontology Query Answering? In *SUM 2013*, pages 15–29, 2013.

[3] J.-M. Lagniez, E. Lonca, and J.-G. Mailly. CoQuiAAS: A Constraint-Based Quick Abstract Argumentation Solver. In *ICTAI 2015*, pages 928–935, 2015.

[4] S. Nofal, K. Atkinson, and P. E. Dunne. Algorithms for decision problems in argument systems under preferred semantics. *Artif. Intell.*, 207:23–51, 2014.

[5] B. Yun, M. Croitoru, S. Vesic, and P. Bisquert. Dagger: Datalog+/- argumentation graph generator. *AAMAS Demos*, to appear, 2017.

[6] B. Yun, M. Croitoru, S. Vesic, and P. Bisquert. Graph theoretical properties of logic based argumentation frameworks: Proofs and general results. In *GKR 2017*, pages 118–138, 2017.

[7] B. Yun, S. Vesic, M. Croitoru, P. Bisquert, and R. Thomopoulos. A Structural Benchmark for Logical Argumentation Frameworks. In *IDA 2017*, pages 334–346, 2017.