

A structural benchmark for logical argumentation frameworks

Bruno Yun, Srdjan Vesic, Madalina Croitoru, Pierre Bisquert, Rallou Thomopoulos

► **To cite this version:**

Bruno Yun, Srdjan Vesic, Madalina Croitoru, Pierre Bisquert, Rallou Thomopoulos. A structural benchmark for logical argumentation frameworks. IDA: Intelligence Data Analysis, Oct 2017, Londres, United Kingdom. pp.334-346, 10.1007/978-3-319-68765-0_28 . lirmm-01668017

HAL Id: lirmm-01668017

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01668017>

Submitted on 19 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A structural benchmark for logical argumentation frameworks

Bruno Yun¹, Srdjan Vesic², Madalina Croitoru¹, Pierre Bisquert³, and Rallou Thomopoulos³

¹ INRIA GraphIK, LIRMM, Université de Montpellier, France

² CRIL - CNRS, Université d'Artois, France

³ INRIA GraphIK, INRA, Montpellier, France

Abstract. This paper proposes a practically-oriented benchmark suite for computational argumentation. We instantiate abstract argumentation frameworks with existential rules, a language widely used in Semantic Web applications and provide a generator of such instantiated graphs. We analyse performance of argumentation solvers on these benchmarks.

1 Introduction

Amongst the plethora of tools for reasoning in presence of inconsistency, argumentation has always held a particular place because of its proximity with real world interaction [Leite *et al.*, 2015]. In this paper, we focus on *logic based argumentation* where abstract argumentation frameworks [Dung, 1995] are instantiated by constructing arguments and attacks from inconsistent knowledge bases. Logic-based argumentation has been studied with *many frameworks proposed*: assumption-based argumentation frameworks [Bondarenko *et al.*, 1993], DeLP [García and Simari, 2004], deductive argumentation [Besnard and Hunter, 2008] or ASPIC/ASPIC+ [Amgoud *et al.*, 2006; Modgil and Prakken, 2014].

Despite argumentation being a mature field, *practically inspired benchmarks are currently missing*. As a rare example of a practical argumentation benchmark consider NoDE⁴, which contains graphs that model debates from Debatepedia⁵, the drama “Twelve Angry Men” by Reginald Rose and Wikipedia revision history. However, the graphs from this benchmark are small (many of them have less than 10 arguments) and their structure is simplistic. The lack of benchmark was acknowledged by the community long time ago, but became obvious with the appearance of the International Competition on Computational Models of Argumentation (ICCMA)⁶. This is why new algorithms are always tested on randomly generated graphs, e.g Nofal *et al.* [2014] and Cerutti *et al.* [2013].

The goal of this paper is to address this drawback by *generating argumentation graphs from knowledge bases and studying their properties empirically* (by running

⁴ <http://www-sop.inria.fr/NoDE/>

⁵ <http://debatepedia.org/>

⁶ <http://argumentationcompetition.org/>

the argumentation solvers). We use an existing logic-based argumentation framework [Croitoru and Vesic, 2013; Croitoru *et al.*, 2015] instantiated with existential rules. This language was chosen because of its practical interest on the Semantic Web [Thomazo and Rudolph, 2014; Thomazo, 2013; Zhang *et al.*, 2016]. Existential rules generalise Description Logics fragments (such as DL-Lite, etc.) that are underlying OWL profiles. Therefore, the choice of this language is significant for Semantic Web applications, notably Ontology-Based Data Access (OBDA) applications. Given the amount of ontologies and data sources made available by such applications, the paper positioning within this language demonstrates its practical interest and relevance for benchmarking argumentation frameworks. Existential rules possess particular features of interest for logic-based argumentation frameworks such as n-ary (as opposed to binary only) negative constraints or existential variables in the head of rules.

The contribution of the paper is the first benchmark in the literature that uses graphs generated from knowledge bases expressed with existential rules instead of random graphs. Using a suite of parametrised existential rule knowledge bases, *we produced the first large scale practically-oriented benchmark in the literature*. Furthermore, we run the top six solvers from ICCMA 2015 *on the generated benchmark and show that the ranking is considerably different* from the one obtained during the competition on randomly generated graphs.

This paper is of interest to both argumentation community and data analysis community. Indeed, for data analysis, the existence of real benchmarks of arguments could be of interest because it can pave the way for intelligent analysis of such instances. These results could then further our comprehension of argumentation graphs structural properties.

2 Background Notions

In this paper we use the existential rule instantiation of argumentation frameworks of Croitoru and Vesic [2013]. The existential rules language [Calì *et al.*, 2009] extends plain Datalog with *existential variables* in the rule head and is composed of formulae built with the usual quantifiers (\exists, \forall) and *only* two connectors: implication (\rightarrow) and conjunction (\wedge). A subset of this language, also known as *Datalog[±]*, refers to identified decidable existential rule fragments [Gottlob *et al.*, 2014; Baget *et al.*, 2011]. The language has attracted much interest recently in the Semantic Web and Knowledge Representation community for its suitability for representing knowledge in a distributed context such as Ontology Based Data Access applications [Baget *et al.*, 2011; Thomazo and Rudolph, 2014; Thomazo, 2013; Magka *et al.*, 2013; Zhang *et al.*, 2016]. The language is composed of the following elements. A *fact* is a ground atom of the form $p(t_1, \dots, t_k)$ where p is a predicate of arity k and $t_i, i \in [1, \dots, k]$ constants. An *existential rule* is of the form $\forall \vec{X}, \vec{Y} H[\vec{X}, \vec{Y}] \rightarrow \exists \vec{Z} C[\vec{Z}, \vec{X}]$ where H (called the hypothesis) and C (called the conclusion) are existentially closed atoms or conjunctions of existentially closed atoms and $\vec{X}, \vec{Y}, \vec{Z}$ their respective vectors of variables. A *rule is applicable* on a set of facts \mathcal{F} iff there exists a homomorphism from the hypothesis of the rule to \mathcal{F} . Applying a rule to a set of facts (also called *chase*) consists of adding the set of atoms of the conclusion of the rule to the facts according to the application homomorphism. A *negative constraint* (NC) is a particular kind of rule where

C is \perp (*absurdum*). It implements weak negation. A *knowledge base* $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is composed of a finite set of facts \mathcal{F} , a set of rules \mathcal{R} and a set of negative constraints \mathcal{N} . We denote by $\mathcal{C}\ell_{\mathcal{R}}^*(\mathcal{F})$ the *closure* of \mathcal{F} by \mathcal{R} (computed by all possible rule \mathcal{R} applications over \mathcal{F} until a fixed point). $\mathcal{C}\ell_{\mathcal{R}}^*(\mathcal{F})$ is said to be \mathcal{R} -consistent if no negative constraint hypothesis can be deduced. Otherwise $\mathcal{C}\ell_{\mathcal{R}}^*(\mathcal{F})$ is \mathcal{R} -inconsistent. Note that different *chase* mechanisms use different simplifications that prevent infinite redundancies [Baget *et al.*, 2011]. In fact, $\mathcal{C}\ell_{\mathcal{R}}^*(\mathcal{F})$ is a finite set when we restrict ourselves to recognisable *finite extension set* classes [Baget *et al.*, 2011] of existential rules (i.e. those sets of rules that when applied over a set of facts guarantee a finite closure) and use a skolem chase (i.e. the rule application operator that replaces every existential variable with a function depending on the the hypothesis' variables) for saturation [Marnette, 2009].

Example 1. Consider the following simple knowledge base \mathcal{K} : James is a cat. James is affectionate. James is handsome. James is intelligent. All cats are mammals. One cannot be affectionate, handsome and intelligent at the same time ⁷.

Formally, $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, where:

$$\begin{aligned}\mathcal{F} &= \{cat(James), affectionate(James), \\ &\quad handsome(James), intelligent(James)\}. \\ \mathcal{R} &= \{\forall x cat(x) \rightarrow mammal(x)\}. \\ \mathcal{N} &= \{\forall x (affectionate(x) \wedge handsome(x) \\ &\quad \wedge intelligent(x) \rightarrow \perp)\}.\end{aligned}$$

We can see that the set of facts is \mathcal{R} -inconsistent. Indeed, by using solely \mathcal{F} we are able to deduce the hypothesis of the negative constraint in \mathcal{N} .

An argument [Croitoru and Vesic, 2013] in *Datalog*[±] is composed of a minimal (with respect to set inclusion) set of facts called *support* and a *conclusion* entailed from the support. The Skolem chase coupled with the use of decidable classes of *Datalog*[±] ensures the finiteness of the proposed argumentation framework [Baget *et al.*, 2011].

Definition 1. Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base. An argument a is a tuple (H, C) with H a non-empty \mathcal{R} -consistent subset of \mathcal{F} and C a set of facts such that:

- $H \subseteq \mathcal{F}$ and $\mathcal{C}\ell_{\mathcal{R}}^*(H) \not\equiv \perp$ (consistency)
- $C \subseteq \mathcal{C}\ell_{\mathcal{R}}^*(H)$ (entailment)
- $\nexists H' \subset H$ s.t. $C \subseteq \mathcal{C}\ell_{\mathcal{R}}^*(H')$ (minimality)

The support H of an argument a is denoted by $Supp(a)$ and the conclusion C by $Conc(a)$. If X is a set of arguments, we denote by $Base(X) = \bigcup_{a \in X} Supp(a)$.

Example 1 (cont.). An example of an argument is $a_1 = (\{affectionate(James)\}, \{affectionate(James)\})$ which states that James is affectionate. Moreover, the minimality implies that arguments that possess excess information in their supports like

⁷ The example is obviously fictitious.

$(\{affectionate(James), cat(James)\}, \{affectionate(James)\})$ are not considered. Another example of argument is $a_2 = (\{intelligent(James), handsome(James)\}, \{intelligent(James), handsome(James)\})$. The argument $a_3 = (\{cat(James)\}, \{mammal(James)\})$ is another example of an argument.

To capture inconsistencies between arguments, we consider the undermining attack [Croitoru and Vesic, 2013]: a attacks b iff the union of the conclusion of a and an element of the support of b is \mathcal{R} -inconsistent.

Definition 2. An argument a attacks an argument b denoted by $(a, b) \in \mathcal{C}$ (or aCb) iff $\exists \phi \in Supp(b)$ s.t. $Conc(a) \cup \{\phi\}$ is \mathcal{R} -inconsistent.

Now that we defined the structure of arguments and attacks, the argumentation graph corresponding to a knowledge base simply consists of all arguments and attacks that can be generated.

Definition 3. An argumentation graph \mathcal{AS} is a tuple $(\mathcal{A}, \mathcal{C})$ where \mathcal{A} is a set of arguments and $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation between arguments called attacks. The argumentation graph instantiated over a knowledge base \mathcal{K} is denoted by $\mathcal{AS}_{\mathcal{K}}$, where the set of arguments and attacks follow Definition 1 and Definition 2 respectively.

Example 1 (cont.). We have that a_2 attacks a_1 but a_1 does not attack a_2 (as we consider a subset of the support of a_2 we cannot entail a negative constraint). This is an example that shows that the graph is not symmetric, which is due to the presence of n -ary constraints.

The complete graph for the knowledge base of Example 1 is composed of 27 arguments and 144 attacks and is represented in Fig. 1. For example, the argument $a_{7.0} = (\{intelligent(James)\}, \{intelligent(James)\})$ is attacked by the argument $a_{5.2} = (\{affectionate(James), handsome(James)\}, \{affectionate(James), handsome(James)\})$.

When considering an argumentation graph $\mathcal{AS} = (\mathcal{A}, \mathcal{C})$, one is often interested in the several consistent viewpoints (or subsets of arguments) that can be inferred. Let $E \subseteq \mathcal{A}$ and $a \in \mathcal{A}$. We say that E is *conflict-free* iff there exists no arguments $a, b \in E$ such that $(a, b) \in \mathcal{C}$. E *defends* a iff for every argument $b \in \mathcal{A}$, if we have $(b, a) \in \mathcal{C}$ then there exists $c \in E$ such that $(c, b) \in \mathcal{C}$. E is *admissible* iff it is conflict-free and defends all its arguments. E is a *complete* (CO) extension iff E is an admissible set which contains all the arguments it defends. E is a *preferred extension* (PR) iff it is maximal (with respect to set inclusion) admissible set. E is a *stable extension* (ST) iff it is conflict-free and for all $a \in \mathcal{A} \setminus E$, there exists an argument $b \in E$ such that $(b, a) \in \mathcal{C}$. E is a *grounded extension* (GR) iff E is a minimal (for set inclusion) complete extension.

3 The Benchmark

The aim of this section is to detail the generation of benchmarks based on argumentation graphs instantiated using existential rules. As seen in the previous section, existential

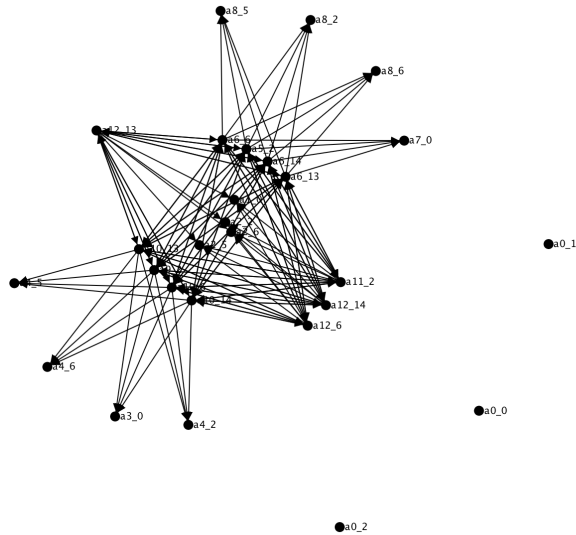


Fig. 1: Graph representation of the instantiated argumentation framework constructed on the knowledge base of Example 1

rules, as a logical language, provide many features (n-ary negative constraints, existential variables in the rule conclusion) that make the instantiated argumentation graph far from simplistic. Furthermore the instantiated graph is reflecting the structure of OBDA inconsistent knowledge bases and it is thus justifying its interest as practical benchmark. Generating such graphs is thus significant for a broader community interested in reasoning in presence of inconsistency on the Semantic Web.

We explain the generation of the *benchmark graphs* in Section 3.1. Then, in Section 3.2, we run *top argumentation solvers on the benchmark* and discuss the results. The goal of this experimental part is to see how the solvers perform on graphs generated from logical knowledge bases and compare performance with respect to randomly generated graphs.

All experiments presented in this section were performed on a VirtualBox Linux machine running with a clean Ubuntu installation with one allocated processor (100%) of an Intel core i7-6600U 2.60GHz and 8GB of RAM. The result files are available upon request (the files amount to more than 15 GB of data).

3.1 Benchmark Generation

Knowledge Base Generation We generated a total of 134 knowledge bases: 108 different knowledge bases for the set of small graphs (denoted b_1 to b_{108}) and 26 for the set of big graphs accessible online at https://github.com/anonymousIDA/Knowledge_bases. This was done in order to have graphs of similar sizes as those of the 2015 *International Competition on Computational Models of Argumentation (IC-*

CMA 2015). The ICCMA benchmark contains two sets of graphs: a set composed of small graphs (less than 383 arguments) and a set of big graphs (3783 to 111775 arguments). We define, for a fixed size of generated \mathcal{F} (that varied from 2 to 5), some knowledge bases with binary (respectively ternary when applicable) constraints in order to obtain an incremental coverage of the facts. We then add rules in a similarly incremental manner. Table 1 shows the characteristics of the knowledge bases we selected. For example, if considering 3 facts $a(m), b(m), c(m)$, we chose a representative of binary constraints as $\forall x(a(x) \wedge b(x) \rightarrow \perp)$ or $\forall x(a(x) \wedge b(x) \rightarrow \perp)$ or $\forall x(a(x) \wedge b(x) \rightarrow \perp)$. We then chose $\forall x(a(x) \wedge b(x) \wedge c(x) \rightarrow \perp)$.

Name of the KB	Number of facts	Number of rules	Number of NC	Type of NC	Number of Args	Number of Attacks
b_1 to b_6	2 to 7	\emptyset	1	Binary	2 to 95	2 to 2048
b_{32}	3	\emptyset	2	Binary	4	6
b_{33} to b_{35}	4	\emptyset	2 to 3	Binary	7 to 9	24 to 32
b_{36} to b_{40}	5	\emptyset	2 to 3	Binary	14 to 19	56 to 128
b_7 to b_{12}	2	1 to 6	1	Binary	4 to 30	5 to 240
b_{13} to b_{18}	2	2,4 or 6	1	Binary	6 to 30	15 to 450
b_{19} to b_{28}	2 to 7	1 or 3	1	Binary	11 to 383	32 to 32768
b_{29} to b_{31}	3	2	1	Binary	16	27 to 30
b_{57} to b_{58}	3	1	2	Binary	8	13 to 14
b_{59} to b_{82}	4	3	2 to 4	Binary	22 to 71	123 to 896
b_{41} to b_{56}	3 to 6	\emptyset	1 to 3	Ternary	6 to 55	9 to 752
b_{83} to b_{84}	3	1	1	Ternary	12	29 to 39
b_{85} to b_{87}	3	2	1	Ternary	24	93 to 147
b_{88} to b_{108}	4	3	1 to 2	Ternary	78 to 103	990 to 2496

Table 1: Characteristics of the small knowledge bases.

From Knowledge Bases To Argumentation Graphs In the argumentation graph generation process, we only kept knowledge bases whose argumentation framework is not automorphic to a previously generated graph. The KB format is *dlgp* [Baget *et al.*, 2015b], allowing translations to and from various Semantic Web languages such as RDF/S, OWL, RuleML or SWRL [Baget *et al.*, 2015a]. For graph generation we made use of the *Graal* [Baget *et al.*, 2015c] framework, a Java toolkit for reasoning within the framework of existential rules. Graal was used for storing the existential rule knowledge bases and for computing conflicts. On top of Graal we provided a graph generation program that works in three steps:

1. All possible arguments are generated: \mathcal{R} -consistent subsets of \mathcal{F} are used as supports and conclusions are deduced from them (see Definition 1).
2. Non minimal arguments are removed (see Definition 1).
3. Attacks are computed following Definition 2.

The obtained graphs were translated in the *Aspartix* (*apx*) format (the same format used in ICCMA 2015).

Example 2. Let us consider the knowledge base $b_{44} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, where $\mathcal{F} = \{a(m), b(m), c(m), d(m), e(m)\}$, $\mathcal{R} = \emptyset$ and $\mathcal{N} = \{\forall x(a(x) \wedge b(x) \wedge c(x) \rightarrow \perp)\}$. The corresponding argumentation graph $\mathcal{AS}_{\mathcal{K}}$ is composed of 26 arguments and 144 attacks and is represented in Fig. 2. We show by this example that some of our generated graphs also possess a sense of “symmetry”.

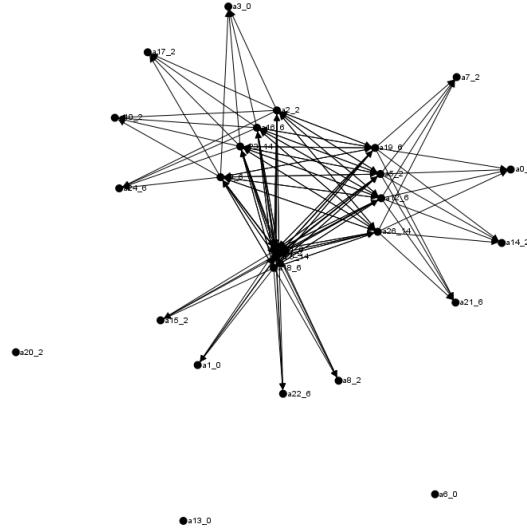


Fig. 2: Representation of the argumentation graph corresponding to b_{44} .

In the next section, we run the top 6 argumentation solvers on the proposed benchmark and discuss the obtained results.

3.2 Benchmark Solvers Results

We recall that the graphs used in the ICCMA 2015 benchmark were separated in three sets: a first set of large graphs (1152 to 9473 arguments) with large grounded extensions and an average density of 1.00%⁸, a second set of smaller graphs (141 to 400 arguments) with numerous complete/preferred/stable extensions and an average density of 3.68% and a third set of medium graphs (185 to 996 arguments) with rich structure of strongly connected components and an average density of 7.75%. Our benchmark graphs are denser, having an average density of 31.27% for small graphs and 29.69% for large graphs.

To see if the proposed benchmark graphs behave in a similar manner as the randomly generated graphs of ICCMA 2015, we ran the top six solvers of the competition:

⁸ Graph density for a directed $G = (V, E)$ is equal to $\frac{|E|}{|V|(|V|-1)}$ where V is the set of nodes and E the set of arcs.

CoQuiAAS, *ArgSemSAT (ArgS.SAT)*, *LabSATSolver (LabSATS.)*, *ASGL*, *ASPARTIX-D* and *ArgTools (ArgT.)*. We used the solvers to complete two computational tasks: SE (given an abstract argumentation framework, determine some extension) and EE (given an abstract argumentation framework, determine all extensions). These two computational tasks were to be solved with respect to the following standard semantics: *Complete Semantics (CO)*, *Preferred Semantics (PR)*, *Grounded Semantics (GR)* and *Stable Semantics (ST)*.

In order to have similar assessment conditions, we used exactly the same ranking method as ICCMA 2015. The solvers were ranked with respect to the number of time-outs on these instances and ties were broken by the actual runtime on the instances. Table 2 shows the average time needed for each solver to complete each task for each semantics in the case of small graphs. There were no errors or time-outs thus the average time reflects the actual ranking (see Table 3).

For large instances, many solvers did not support large inputs resulting in several crashes / errors. Ties were broken by the average time of successfully solved instances (Table 5). Please note that for large graphs, for some tasks, some solvers timed out for all instances resulting in equal rankings (*EE-CO: ASGL and ArgTools* for instance).

It is noticeable that *CoQuiAAS* comes first in the two batches of generated graphs. As an explanation, please note that *CoQuiAAS* is based on MiniSAT solver, which is known to work well in the presence of structured information (i.e. symmetries). It might be the case that the generated graphs keep some of their structure even after being translated into a SAT instance which could explain the obtained result.

In order to see how different the solver ranking on the random benchmark used by ICCMA 2015⁹ is from the solver ranking on the knowledge base benchmark, we used the normalised Kendall tau distance.¹⁰ The distance outputs 0 if two rankings are identical and 1 if one ranking is the reverse of the other. Table 7 shows the normalised Kendall tau distance between the rankings of the generated graphs and the competition ranking. What comes out is that:

- Although the ranking of the ICCMA 2015 benchmark and the one for large graphs for the task EE-GR is slightly different (we can not break the tie between ASPARTIX-d and ASGL), they are identical with respect to Kendall tau.
- We have the same normalised Kendall tau distance for the small graphs and the large graphs for the tasks SE-CO, SE-PR and SE-GR.
- The small graphs have a higher normalised Kendall tau distance than the large graphs for the tasks SE-ST, EE-CO, EE-PR and EE-GR.
- The small graphs have a lower normalised Kendall tau distance than the large graphs for the task EE-SET.
- In average, the results are more similar for the large graphs than for the small graphs.

This benchmark is interesting because it shows that for the instantiated graphs we generated, it is strongly advised to use *CoQuiAAS* as the solver. For relatively small

⁹ <http://argumentationcompetition.org/2015/results.html>

¹⁰ This distance is equal to the number of pairwise disagreements between two ranking lists and is normalised by dividing by $\frac{n(n-1)}{2}$, where n is the number of solvers.

	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	0,0138	0,1719	0,0059	0,0249	0,0031	0,3644
SE-PR	0,0165	0,2137	0,0059	0,4445	0,0007	0,2906
SE-GR	0,0339	0,2101	0,0057	0,3217	0,0010	0,1944
SE-ST	0,0148	0,2194	0,0060	0,0279	0,0018	0,2520
EE-CO	0,0694	0,2282	0,0096	0,0247	0,0024	0,2908
EE-PR	0,0517	0,1660	0,0085	0,5763	0,0029	0,3765
EE-GR	0,0325	0,1861	0,0052	0,3239	0,0016	0,2262
EE-ST	0,0486	0,1661	0,0065	0,0231	0,0027	0,3151

Table 2: Average time for small instances (in sec).

	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	3	5	2	4	1	6
SE-PR	3	4	2	6	1	5
SE-GR	3	5	2	6	1	4
SE-ST	3	5	2	4	1	6
EE-CO	4	5	2	3	1	6
EE-PR	3	4	2	6	1	5
EE-GR	3	4	2	6	1	5
EE-ST	4	5	2	3	1	6

Table 3: Corresponding ranking on small graphs.

	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	15	26	16	1	0	11
SE-PR	18	1	17	26	0	11
SE-GR	17	0	18	26	0	11
SE-ST	15	0	18	2	0	11
EE-CO	22	26	26	9	2	21
EE-PR	21	26	26	26	15	21
EE-GR	16	26	17	26	0	11
EE-ST	15	23	17	1	1	11

Table 4: Number of timeouts on the generated large graphs.

	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	4	6	5	2	1	3
SE-PR	5	2	4	6	1	3
SE-GR	4	2	5	6	1	3
SE-ST	5	2	6	3	1	4
EE-CO	4	5	5	2	1	3
EE-PR	2	4	4	4	1	3
EE-GR	3	5	4	5	1	2
EE-ST	4	6	5	2	1	3

Table 5: Ranking on the generated large graphs.

graphs, the choice of the solver can be bypassed as the differences are negligible. However, for larger graphs, we noticed several issues:

- It seems that ASGL uses a different algorithm for SE-GR and EE-GR (this is very noticeable by the difference in the number of timeouts).
- ASGL is not suitable for finding complete extensions.

	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	4	2	5	3	1	6
SE-PR	1	4	6	5	3	2
SE-GR	3	5	4	6	1	2
SE-ST	2	5	6	1	4	3
EE-CO	2	5	6	1	3	4
EE-PR	1	4	6	5	2	3
EE-GR	3	5	4	6	1	2
EE-ST	2	4	5	1	3	6

Table 6: Rankings extracted from the ICCMA 2015 website.

	Small graphs	Large graphs
SE-CO	0.400	0.400
SE-PR	0.467	0.467
SE-GR	0.200	0.200
SE-ST	0.600	0.467
EE-CO	0.467	0.200
EE-PR	0.400	0.067
EE-GR	0.267	0.000
EE-ST	0.333	0.400
Average	0.392	0.275

Table 7: Normalised Kendall tau distance between the rankings of the generated graphs and the competition ranking.

- Aspartix-D is not suitable for finding preferred and grounded extensions.
- There are 15 instances that were too big to perform the task EE-PR for all solvers.

4 Discussion

This paper starts from the observation that benchmarks of argumentation graphs generated from knowledge bases are currently missing in the literature. We thus propose to consider logic based argumentation frameworks instantiated with existential rules. We provided a tool for generating such graphs out of existential rule knowledge bases. We ran top argumentation solvers on the generated benchmark and analysed their performance with respect to performance on randomly generated graphs.

Note that constructing all the arguments from the knowledge base might result in a big number of arguments. One could reduce the number of arguments by preserving only some of them, i.e. by keeping only the so called *core* [Amgoud *et al.*, 2014]. In the present paper, we do not use the notion of a core because we do not want the choice of the core (there are several possibilities) to influence the results of this first study. As it is not convenient to generate too many arguments in practice, investigating benchmark generation using different notions of core is part of future work.

Acknowledgments

Srdjan Vesic benefited from the support of the project AMANDE ANR-13-BS02-0004 of the French National Research Agency (ANR).

Bibliography

- Leila Amgoud, Lianne Bodenstaff, Martin Caminada, Peter McBurney, Simon Parsons, Henry Prakken, Jelle Van Veenen, and Gerard Vreeswijk. Final review and report on formal argumentation system. *Deliverable D2*, 2006.
- Leila Amgoud, Philippe Besnard, and Srdjan Vesic. Equivalence in logic-based argumentation. *Journal of Applied Non-Classical Logics*, 24(3):181–208, 2014.
- Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.
- Jean-François Baget, Alain Gutierrez, Michel Leclère, Marie-Laure Mugnier, Swan Rocher, and Clément Sipieter. Datalog+, RuleML and OWL 2: Formats and Translations for Existential Rules. In *Proceedings of the RuleML 2015 Challenge, the Special Track on Rule-based Recommender Systems for the Web of Data, the Special Industry Track and the RuleML 2015 Doctoral Consortium hosted by the 9th International Web Rule Symposium (RuleML 2015), Berlin, Germany, August 2-5, 2015.*, 2015.
- Jean-François Baget, Alain Gutierrez, Michel Leclère, Marie-Laure Mugnier, Swan Rocher, and Clément Sipieter. DLGP: An extended Datalog Syntax for Existential Rules and Datalog+/- Version 2.0, June 2015.
- Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, Swan Rocher, and Clément Sipieter. Graal: A Toolkit for Query Answering with Existential Rules. In *Rule Technologies: Foundations, Tools, and Applications - 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings*, pages 328–344, 2015.
- Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. MIT Press, 2008.
- Andrei Bondarenko, Francesca Toni, and Robert A. Kowalski. An Assumption-Based Framework for Non-Monotonic Reasoning. In *LPNMR*, pages 171–189, 1993.
- Andrea Cali, Georg Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *Proceedings of the Twenty-Eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, June 19 - July 1, 2009, Providence, Rhode Island, USA*, pages 77–86, 2009.
- Federico Cerutti, Paul E. Dunne, Massimiliano Giacomin, and Mauro Vallati. Computing preferred extensions in abstract argumentation: A sat-based approach. In *Theory and Applications of Formal Argumentation - Second International Workshop, TFAFA 2013*, pages 176–193, 2013.
- Madalina Croitoru and Srdjan Vesic. What Can Argumentation Do for Inconsistent Ontology Query Answering? In *Scalable Uncertainty Management - 7th International Conference, SUM 2013, Washington, DC, USA, September 16-18, 2013. Proceedings*, pages 15–29, 2013.
- Madalina Croitoru, Rallou Thomopoulos, and Srdjan Vesic. Introducing Preference-Based Argumentation to Inconsistent Ontological Knowledge Bases. In *PRIMA*

- 2015: *Principles and Practice of Multi-Agent Systems - 18th International Conference, Bertinoro, Italy, October 26-30, 2015, Proceedings*, pages 594–602, 2015.
- Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artif. Intell.*, 77(2):321–358, 1995.
- Alejandro Javier García and Guillermo Ricardo Simari. Defeasible Logic Programming: An Argumentative Approach. *TPLP*, 4(1-2):95–138, 2004.
- Georg Gottlob, Thomas Lukasiewicz, and Andreas Pieris. Datalog+/-: Questions and Answers. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*, 2014.
- Joao Leite, Tran Cao Son, Paolo Torroni, and Stefan Woltran. Applications of logical approaches to argumentation. *Argument and Computation*, 6(1):1–2, 2015.
- Despoina Magka, Markus Krötzsch, and Ian Horrocks. Computing Stable Models for Nonmonotonic Existential Rules. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 1031–1038, 2013.
- Bruno Marnette. Generalized schema-mappings: from termination to tractability. In *Proceedings of the Twenty-Eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, June 19 - July 1, 2009, Providence, Rhode Island, USA*, pages 13–22, 2009.
- Sanjay Modgil and Henry Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.
- Samer Nofal, Katie Atkinson, and Paul E. Dunne. Algorithms for decision problems in argument systems under preferred semantics. *Artif. Intell.*, 207:23–51, 2014.
- Michaël Thomazo and Sebastian Rudolph. Mixing Materialization and Query Rewriting for Existential Rules. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 897–902, 2014.
- Michaël Thomazo. Compact Rewritings for Existential Rules. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 1125–1131, 2013.
- Heng Zhang, Yan Zhang, and Jia-Huai You. Expressive Completeness of Existential Rule Languages for Ontology-Based Query Answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1330–1337, 2016.