NAKED: N-Ary Graphs from Knowledge Bases Expressed in Datalog±

Demonstration

Bruno Yun
INRIA GraphIK, Université de Montpellier, France
yun@lirmm.fr

Srdjan Vesic
CRIL - CNRS, Université d’Artois, France
vesic@cril.fr

Madalina Croitoru
INRIA GraphIK, Université de Montpellier, France
croitoru@lirmm.fr

Pierre Bisquert
IATE, INRA, INRIA GraphIK, France
pierre.bisquert@inra.fr

1 THE NAKED’S TIMELINESS

This demonstration paper will present NAKED, a new generator for n-ary logic-based argumentation frameworks instantiated from inconsistent knowledge bases expressed using Datalog±. The tool allows to import a knowledge base in DLGP format, generate, visualise and export the corresponding argumentation hypergraph. We show its application on a use-case from the NoAW project.

KEYWORDS
Logic-based Argumentation; Datalog±; Agent Reasoning

ABSTRACT
In this demonstration paper, we introduce NAKED: a new generator for n-ary logic-based argumentation frameworks instantiated from inconsistent knowledge bases expressed using Datalog±. The tool allows to import a knowledge base in DLGP format, generate, visualise and export the corresponding argumentation hypergraph. We show its application on a use-case from the NoAW project.

Our workflow will enable any data engineer to (1) input a KB in the well-known DLGP format [6] for Datalog±, (2) generate an argumentation hypergraph that instantiate the framework of Nielsen and Parsons [22], (3) interact with the graph representation by allowing arguments re-positioning, (4) observe a specific argument by highlighting the corresponding argument and its attackers in different colours and (5) export the generated argumentation hypergraph in the DOT format for a better tool interoperability. All of these functions could be useful for a non computer science expert who wants to reason over an inconsistent KB in a particular domain using argumentation [4, 23, 24]. It could also be useful for investigating the theoretical properties of the graph based representation of the generated AF [5, 30]. Given the fact that certain graph theoretical properties could radically improve the extension computation
2 USING THE NAKED TOOL

NAKED is a tool that assists domain experts and argumentation developers in the specification, visualisation and export of logic-based AFs built over the Datalog+ language.

2.1 Agent Techniques: Logic Argumentation

Let us first make a note about the logical language used for instantiating the KBs. Existential rules (whose computationally decidable subclasses are usually referred to as Datalog+) have been recently investigated on the Semantic Web for their generalisation w.r.t. Description Logic fragments [26]. It has been shown [14] that using argumentation techniques over inconsistent existential rules KBs yields extensions logically equivalent to the maximally consistent subsets of the KB, called repairs [19]. Using argumentation over existential rules has been shown to be of practical interest over existing repair based approaches [17]. Argumentation for handling inconsistency tolerant semantics enhance the human interaction with the KB [9, 19]. Such techniques have been shown to have implications w.r.t. human reasoning and bias detection [10].

An existential rule $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is composed of a finite set of facts $\mathcal{F}$ (such as $(\text{packaging}(a))$ representing the fact that the object $a$ is a packaging), a set of rules $\mathcal{R}$ (such as $(\forall X(\text{packaging}(X) \land \text{has}(X, \text{plasticFilm}) \rightarrow \text{pollute}(X)))$ representing the implication that a packaging that has a plastic film is polluting the environment) and a set of negative constraints $\mathcal{N}$ (such as $(\forall X(\text{pollute}(X) \land \text{protectEnv}(X) \rightarrow \bot))$ representing that a certain packaging cannot both protect the environment and pollute it at the same time).

The constraints are used to express negative knowledge about the world. In the considered setting, rules and constraints act as an ontology used to “access” different data sources. Therefore, we suppose that all of the inconsistencies come from the facts and that the set of rules is compatible with the set of negative constraints, i.e. the union of those two sets is satisfiable [19].

Example 2.1 (Datalog+ KB). In this KB, a packaging $a$ with a plastic film is said to protect the environment. However, since the possession of a plastic film leads to pollution, this KB is thus inconsistent. Formally, $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is such that:

- $\mathcal{F} = \{\text{packaging}(a), \text{has}(a, \text{plasticFilm}), \text{protectEnv}(a)\}$
- $\mathcal{R} = \{\forall X(\text{packaging}(X) \land \text{has}(X, \text{plasticFilm}) \rightarrow \text{pollute}(X))\}$
- $\mathcal{N} = \{\forall X(\text{pollute}(X) \land \text{protectEnv}(X) \rightarrow \bot)\}$

Starting from an inconsistent existential rule KB, we generate the arguments and the attacks corresponding to the KB. An argument in Datalog+ is either a fact or built upon other facts. The Skolem chase coupled with the use of decidable classes of Datalog+ ensures the finiteness of the AF proposed (following from [7]). The attack considered is a particular undermining: a set of arguments $S$ attacks $a$ if and only if the union of the conclusions of all arguments in $S$ and an element of the support of $a$ entails a negative constraint. Note that the attack relation is not symmetric.

Example 2.2 (Cont’d Example 2.1). We have six attacks on the following four arguments (represented in Figure 1):

- $A_0 : \text{has}(a, \text{plasticFilm})$
- $A_1 : \text{protectEnv}(a)$
- $A_2 : [A_1, A_3] \rightarrow \text{pollute}(a)$
- $A_3 : \text{packaging}(a)$

An example of attack is $((A_1, A_3), A_0)$.

The AF above outputs a set of preferred extensions [22] equivalent to the repairs [9, 19] of the KB (i.e. the maximal with respect to inclusion consistent sets of facts).

2.2 Usability Scenarios

We consider two usability scenarios of NAKED. All of these scenarios are easily employed using NAKED.

Scenario 1. We consider the task of a specialist inputting an inconsistent KB of his or her expertise and wanting to find the maximally consistent point of views. Please note that tools for assisting non domain experts in building such KBs without computer expertise exists [13]. Finding maximally consistent point of views (or repairs) consists in computing all maximal subsets of facts that do not trigger any negative constraints of $\mathcal{KB}$. There are three repairs: $\{\text{packaging}(a), \text{has}(a, \text{plasticFilm})\}$, $\{\text{packaging}(a), \text{protectEnv}(a)\}$ and $\{\text{has}(a, \text{plasticFilm}), \text{protectEnv}(a)\}$.

Scenario 2. We consider the task of an argumentation expert that wants to generate argumentation hypergraphs for benchmarking purposes. Although efficient algorithms that compute extensions exist for argumentation hypergraphs [21], there is a lack of such graphs. Our tool provides a DOT format output which enables interoperability with many graph tools.

ACKNOWLEDGEMENT

We were supported by the H2020 NoAW project (ID 688338).
REFERENCES


