HAL open science

Sets of Attacking Arguments for Inconsistent Datalog Knowledge Bases

Bruno Yun, Srdjan Vesic, Madalina Croitoru

To cite this version:

Bruno Yun, Srdjan Vesic, Madalina Croitoru. Sets of Attacking Arguments for Inconsistent Datalog Knowledge Bases. COMMA 2020 - 8th International Conference on Computational Models of Argument, Sep 2020, Perugia / Virtual), Italy. pp.419-430, 10.3233/FAIA200526 . hal-03002140

HAL Id: hal-03002140

https://hal.science/hal-03002140v1

Submitted on 12 Nov 2020

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.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Computational Models of Argument
H. Prakken et al. (Eds.)

© 2020 The authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA200526

Sets of Attacking Arguments for Inconsistent Datalog Knowledge Bases

Bruno YUN [a,1], Srdjan VESIC [b] and Madalina CROITORU [c]
[a] *University of Aberdeen, United Kingdom*
[b] *CRIL - CNRS & Univ. Artois, France*
[c] *University of Montpellier, France*

Abstract. Logic-based argumentation is a well-known approach for reasoning with inconsistent logic knowledge bases. Such frameworks have been shown to suffer from a major practical drawback consisting of a large number of arguments and attacks. To address this issue, we provide an argumentation framework that considers sets of attacking arguments and provide a theoretical analysis of the new framework with respect to its syntactic and semantic properties. We provide a tool for generating such argumentation frameworks from a Datalog knowledge base and study their characteristics.

Keywords. argumentation, datalog, SETAF

1. Introduction

In this paper, we place ourselves in the setting of logic-based argumentation instantiated over Datalog. The use of this language ensures that the work of this paper studies potentially real world argumentation graphs and unveils genuine structural behaviour. Logic-based argumentation is a well known approach for reasoning with inconsistent logic knowledge bases (KB). While its strength, in the instantiated case, might not lie in its reasoning efficiency, particularly when compared to other inconsistent tolerant reasoning methods such as ASP [23] or dedicated tools [12]. Its added value is two fold. First, its explanatory power benefits to increase the scrutability of the system by users [3,8]. Second, the use of ranking semantics can induce a stratification of the inconsistent KB [2] that might be of use for query answering techniques [27].

Starting from an inconsistent KB (composed of a set of factual knowledge and an ontology stating positive and negative rules about the factual knowledge), one can attempt to generate the arguments and the attacks corresponding to the KB using existing logic-based AFs: Deductive argumentation [9], ASPIC+ [20], Assumption-Based Argumentation (ABA) [24,11] or DeLP [17]. However, none of these argumentation frameworks (AF) are straightforwardly applicable in the context of Datalog. Indeed, the aforementioned frameworks are not usable without adding or removing any rules or facts in the KB. Let us now illustrate this statement. In the case of ASPIC+, we cannot instantiate it because the definition of the contrariness relation is not general enough to account

[1] Corresponding Author: University of Aberdeen, United Kingdom; E-mail: bruno.yun@ed.ac.uk

for negative constraints. Let us show this on an example. Suppose we are given three facts about the shape and taste of a biscuit: (f_1) "the biscuit has a square shape", (f_2) "the biscuit has a round shape" and (f_3) "the biscuit is sweet". Now, further suppose that there are no rules and one negative constraint: the biscuit cannot have a square and round shape at the same time. As the fact f_3 is a *free-fact* (i.e. it is not involved in any minimal conflict), there is no way to define its contrary intuitively without modifying the set of rules of the KB. Namely, the third item of *Definition 5.1* in the work of [20] specifies that each formula of the language must have at least one contradictory, which is not the case for the latter fact in our example. Of course, it is possible to declare that a fact "the biscuit is not sweet" is the contradictory of f_3. However, for each contradictory, the corresponding negative constraint has to be added to the KB. In the case of ABA, although it is abstract enough to function with a language that has neither implication nor negation, it needs a contrariness function that returns a single contrary sentence for each formula of the language. This is not enough in the case where a fact appears in multiple conflicts and the language does not allow for the disjunction. In the case of DeLP, we cannot instantiate it since the original work only consider ground rules.

Specifically crafted instantiations for Datalog, such as the instantiations of Croitoru and Vesic [14], Yun et al. [28] and Arioua et al. [4], have been proven to respect the argumentation rationality desiderata [1,13] and to output a set of extensions equivalent to the set of repairs [19,10] of the KB (i.e. the maximum consistent sets of facts w.r.t. inclusion). Unfortunately, it was shown that these instantiations suffer from a major drawback: a large number of arguments and attacks [25]. This problem even occurs in the case where there are no rules in the KB (for instance, a graph with 13 arguments and 30 attacks can be generated with a meagre KB with solely 4 facts, no positive rules and a single negative rule). As a consequence, the argumentation graph for a "normally-sized" KB cannot be held in main memory, requires dedicated large-graph visualisation tools, and, despite their polynomial complexity regarding the number of arguments, still poses combinatorial challenges for the computation of ranking techniques. The question that arose is whether or not we can find more efficient AFs for Datalog. To this end, we provide an AF that considers *sets of attacking arguments* (n-ary attacks) [22,21,16] and possesses arguments that are built upon other arguments (*à la* ASPIC+) and n-ary attacks. We show that this new framework retains desirable properties with fewer arguments and attacks compared to the existing frameworks.

There are three main contributions in this paper. First, we introduce a logic-based AF with n-ary attacks for an inconsistent KB expressed using Datalog. Second, we provide a theoretical analysis of the new AF w.r.t. its syntactic and semantic properties. Last, we provide a tool for generating this AF from a KB expressed in Datalog Plus (DLGP) format and study its performance in terms of argumentation graph compression rate and generation time.

The structure of the paper is as follows. In Section 2, we recall the necessary definitions of Datalog and the AF of [28] and [4]. In Section 3, we introduce a new AF and study its theoretical properties. In Section 4, we empirically compare the two frameworks w.r.t. the number of arguments and attacks on a set of KBs.

2. Background

We start by introducing the Datalog language[2]. It is composed of formulae built with the usual quantifier ($\forall$) and *only* two connectors: implication ($\rightarrow$) and conjunction ($\wedge$) and is composed of facts, rules and negative constraints. A *fact* is a ground atom of the form $p(t_1,\ldots,t_k)$ where p is a predicate of arity k and t_i, with $i \in [1,\ldots,k]$, constants. A *(positive) rule* r is of the form $\forall \overrightarrow{X}, \overrightarrow{Y}\ B_r[\overrightarrow{X}, \overrightarrow{Y}] \rightarrow H_r[\overrightarrow{Y}]$ where B_r and H_r are closed atoms or conjunctions of closed atoms, respectively called the body and the head of r, and $\overrightarrow{X}, \overrightarrow{Y}$ their respective vectors of variables. For simplicity purposes, we will consider that the rules only have one atom in the head. This is not a big assumption as it has been proved that an arbitrary set of rules can be transformed into a set of rules with atomic head; see the work of [7] for more details. However, note that the results of this paper can be extended to the case where rule heads are not atomic.

Let X be a set of variables and T be a set of terms (constants or variables). A substitution of X to T is a function from X to T. A homomorphism π from a set of atoms S to a set of atoms S' is a substitution of the variables of S with the terms of S' s.t. $\pi(S) \subseteq S'$. A *rule is applicable* to a set of facts $\mathscr{F}$ iff there exists a homomorphism [5] from its body to $\mathscr{F}$. Applying a rule to a set of facts (also called *chase*) consists of adding the set of atoms of its head to the facts according to the application homomorphism. A *negative constraint* is a rule r of the form $\forall \overrightarrow{X} B_r[\overrightarrow{X}] \rightarrow \bot$ where B_r is a closed atom or conjunctions of closed atoms, $\overrightarrow{X}$ the respective vector of variables and $\bot$ is *absurdum*.

Definition 1 (Knowledge base)**.** *A KB $\mathscr{K}$ is a tuple $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ where $\mathscr{F}$ is a finite set of facts, $\mathscr{R}$ a set of positive rules and $\mathscr{N}$ a set of negative constraints.*

Example 1. *Suppose that one is indecisive about what to eat for an appetiser. He decides that the dish should contain salted cucumbers, sugar, yogurt, not be a soup and be edible. However, he finds out that combining together salted cucumbers, sugar and yogurt may not be a good idea. Furthermore, combining salted cucumbers with yogurt is a dish called "tzaziki" which is a famous greek soup. We model the situation with the KB $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$, where:*

- $\mathscr{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
- $\mathscr{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
- $\mathscr{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$

In the Ontology Based Data Access (OBDA) setting, rules and constraints are used to "access" different data sources. These sources are prone to inconsistencies. We assume that the rules of the KB are compatible with the negative constraints, i.e. the union of those two sets is satisfiable [19]. Indeed, the ontology is believed to be reliable as it is the result of a robust construction by domain experts. However, as data can be heterogeneous due to merging and fusion, the data is assumed to be the source of inconsistency.

[2]For simplicity purposes we use the Datalog language but this work can be easily extended to the Datalog$\pm$ formalism if we restrict ourselves to the class of FES rules.

The *saturation* of a set of facts $\mathscr{F}$ by $\mathscr{R}$ is the set of all possible atoms and conjunctions of atoms that are entailed, after using all rule applications from $\mathscr{R}$ over $\mathscr{F}$ until a fixed point. The output of this process is called the closure and is denoted by $\mathbb{SAT}_{\mathscr{R}}(\mathscr{F})$. A set $\mathscr{F}$ is said to be $\mathscr{R}$*-consistent* if no negative constraint hypothesis can be entailed, i.e. $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(\mathscr{F}) \not\models \bot$. Otherwise, $\mathscr{F}$ is said to be $\mathscr{R}$*-inconsistent*. We introduce the notion of repair (maximal consistent subset) and free-fact.

Definition 2 (Repair)**.** *A repair of $\mathscr{K} = (\mathscr{F},\mathscr{R},\mathscr{N})$ is $X \subseteq \mathscr{F}$ s.t. X is $\mathscr{R}$-consistent and there exists no X' s.t. $X \subset X'$ and X' is $\mathscr{R}$-consistent. The set of all repairs of a KB $\mathscr{K}$ is denoted by $Repair(\mathscr{K})$.*

Definition 3 (Free-fact)**.** *Let $\mathscr{K}$ be a KB, a fact $f \in \mathscr{F}$ is a free-fact iff for every repair $R \in Repair(\mathscr{K}), f \in R$.*

Example 2 (Cont'd Example 1)**.** *In our example, there are three repairs, each representing one alternative: yogurt with sugar (which is common), tzaziki or sugar with salter cucumbers (sweet pickles). Namely, we have that $Repair(\mathscr{K}) = \{R_1, R_2, R_3\}$, where:*

- $R_1 = \{contains(m, saltC), contains(m, yogurt), edible(m)\}$,
- $R_2 = \{contains(m, sugar), contains(m, saltC), notSoup(m), edible(m)\}$,
- $R_3 = \{contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$.

Here, $edible(m)$ is a free-fact.

We now recall the AF provided by [28] and [4] and based on the original framework of [14]. This AF has deductive arguments and an asymmetric attack relation based on the notion of undermining. An argument a attacks an argument b if the conclusion of the argument a is incompatible with one element of the hypothesis of the argument b.

Definition 4 (Argumentation framework $\mathfrak{AS}'$)**.** *Let $\mathscr{K} = (\mathscr{F},\mathscr{R},\mathscr{N})$ be a KB. The corresponding AF, denoted by $\mathfrak{AS}'_{\mathscr{K}}$, is the pair $(\mathscr{A}',\mathscr{C}')$ with $\mathscr{C}' \subseteq \mathscr{A}' \times \mathscr{A}'$ such that:*

- *An argument $a' \in \mathscr{A}'$ is a tuple (H,C) with H a non-empty $\mathscr{R}$-consistent subset of $\mathscr{F}$ and C a set of facts s.t. (1) $C \subseteq \mathbb{SAT}_{\mathscr{R}}(H)$ and (2) there is no $H' \subset H$ s.t. $C \subseteq \mathbb{SAT}_{\mathscr{R}}(H')$. The support H of an argument a' is denoted by $Supp(a')$ and the conclusion C by $Conc(a')$.*
- *a' attacks b', denoted by $(a',b') \in \mathscr{C}'$, iff there exists $\varphi \in Supp(b')$ s.t. $Conc(a') \cup \{\varphi\}$ is $\mathscr{R}$-inconsistent.*

Example 3 (Cont'd Example 1)**.** *The AF $\mathfrak{AS}'_{\mathscr{K}}$ has 33 arguments and 360 attacks. Moreover, a'_1 defined by $(\{contains(m, saltC), contains(m, yogurt)\}, \{tzaziki(m)\})$ attacks argument a'_2 defined by $(\{notSoup(m)\}, \{notSoup(m)\})$ but a'_2 does not attack a'_1 because $\{notSoup(m)\}$ is not $\mathscr{R}$-inconsistent with either the atom $contains(m, saltC)$ or $contains(m, yogurt)$.*

The AF $\mathfrak{AS}'_{\mathscr{K}}$ generated from a KB $\mathscr{K}$, has been proven to possess good properties such as the equivalence between the set of repairs and the set of preferred (resp. stable) extensions, the desirable postulates and the equivalence results for query answering in the OBDA field [14].

However, one of the main drawbacks of this method is the huge number of arguments. Indeed, [25] proved that the number of arguments is exponential w.r.t. the num-

ber of free-facts. Moreover, it was empirically shown that for a KB with eight facts, six rules and two ternary negative constraints, we might generate 11,007 arguments and 23,855,104 attacks [28].

3. New Argumentation framework $\mathfrak{AS}$

In this section, we show a novel AF for generating arguments and attacks from an inconsistent KB. We also show that this AF possesses all of the desirable properties of $\mathfrak{AS}'_{\mathscr{K}}$.

Note that although the framework described in this section has some similarities with the ASPIC+ framework, the ASPIC+ cannot be directly instantiated with Datalog because the language does not have the negation and the contrariness function is not general enough for this language. Moreover, when instantiating ASPIC+, one usually has to add all the tautologies of the language in the set of rules to guarantee that the result will be consistent, i.e. to satisfy the rationality postulates defined by [13]. To avoid adding this huge number of rules and also with the goal of decreasing the number of arguments, we propose not to add them. However, the cost of forgetting to add those rules (and the arguments generated using them) would result in a violation of rationality postulates. We propose to solve this problem in a more elegant way. Namely, we allow for the use of sets of attacking arguments (i.e. n-ary attacks).

Definition 5 (Argumentation framework $\mathfrak{AS}$)**.** *Let us consider the KB $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$. The corresponding AF, denoted by $\mathfrak{AS}_{\mathscr{K}}$, is the pair $(\mathscr{A}, \mathscr{C})$ with $\mathscr{C} \subseteq (2^{\mathscr{A}} \setminus \{\emptyset\}) \times \mathscr{A}$ such that:*

- *An argument $a \in \mathscr{A}$ is either **(1)** a fact f, where $f \in \mathscr{F}$ s.t. $Conc(a) = f$ and $Prem(a) = \{f\}$ or **(2)** $a_1, \ldots, a_n \rightarrow f'$ where $a_1, \ldots, a_n \in \mathscr{A}$ s.t. there exists a tuple (r, π) where $r \in \mathscr{R}, \pi$ is a homomorphism from the body of r to $\{Conc(a_1), \ldots, Conc(a_n)\}$ and f' is the resulting atom from the rule application. $Conc(a) = f'$ and $Prem(a) = Prem(a_1) \cup \cdots \cup Prem(a_n)$. Note that in both cases, $Prem(a)$ must be $\mathscr{R}$-consistent.*
- *An attack in $\mathscr{C}$ is a pair (X, a) s.t. X is minimal for set inclusion s.t. $\bigcup_{x \in X} Prem(x)$ is $\mathscr{R}$-consistent and there exists $\varphi \in Prem(a)$ s.t. $(\bigcup_{x \in X} Conc(x)) \cup \{\varphi\}$ is $\mathscr{R}$-inconsistent.*

With a slight abuse of notation, we also use the notation $Conc(a)$ to refer to the conclusion of an argument in $\mathfrak{AS}$. However, the conclusion is not a set anymore (see Definition 4). The reason is that $\mathscr{K}$ can be processed w.l.o.g. to contain only rules with atomic head [7].

Notation: Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a KB, $X \subseteq \mathscr{F}$ be a set of facts and $X' \subseteq \mathscr{A}$ be a set of arguments of $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$. We define the set of arguments generated by X as $Arg(X) = \{a \in \mathscr{A} \mid Prem(a) \subseteq X\}$ and the base of a set of arguments X' as $Base(X') = \bigcup_{x' \in X'} Prem(x')$. We define $Concs(X') = \bigcup_{x' \in X'} Conc(x')$.

In case of binary attacks [15], a set of arguments X is said to attack an argument a iff there exists $b \in X$ s.t. b attacks a. We need a similar notion here except that we

already have a notion of attack from a set towards an argument. In order not to mix up the two notions, we introduce the notation $\mathscr{C}^*$, which stands for the saturated set of attacks. For example, if $(\{a,b\},c) \in \mathscr{C}$ then each set X' containing a and b (i.e. s.t. $\{a,b\} \subseteq X'$) attacks c too (i.e. $(X',c) \in \mathscr{C}^*$).

Definition 6 (Saturated set of attacks)**.** *Let $\mathfrak{AS} = (\mathscr{A},\mathscr{C})$ be an AF. The saturated set of attacks of $\mathfrak{AS}$ is $\mathscr{C}^* = \{(X,a) \mid$ there exists $(X',a) \in \mathscr{C}$ with $X' \subseteq X \subseteq \mathscr{A}\}$.*

Example 4 (Cont'd Example 1)**.** *The argumentation graph $\mathfrak{AS}_{\mathscr{K}}$ is composed of the six following arguments and 11 attacks: $a_1 = contains(m,sugar)$, $a_2 = contains(m,saltC)$, $a_3 = contains(m,yogurt)$, $a_4 = notSoup(m)$, $a_5 = edible(m)$ and $a_6 = a_2,a_3 \rightarrow tzaziki(m)$. An example attack of $\mathscr{C}$ is $(\{a_1,a_2\},a_3)$.*

From $\mathscr{K}$, one can build an AF $\mathfrak{AS}_{\mathscr{K}}$ with sets of attacking arguments (see Definition 5). Please note that although the work of Yun et al. [26] seems similar, it is based on building all arguments using Definition 4, filtering specific arguments and filling up the loss of information induced by the missing arguments with sets of attacking arguments to keep the rationality postulates. Let us illustrate the difference between the framework of Yun et al. [26] and the new AF on a KB with 3 facts and a single negative constraint on those three facts. In the framework of Yun et al., there will be six arguments and nine attacks whereas there are three arguments and three attacks in the new AF.

3.1. Argumentation framework properties of $\mathfrak{AS}_{\mathscr{K}}$

The AF $\mathfrak{AS}$ is an instantiation of the abstract SETAF framework proposed by Nielsen and Parsons [21,22]. For the purpose of the paper being self-contained, we recall the necessary definitions.

Definition 7 (Argumentation semantics)**.** *Let $\mathfrak{AS} = (\mathscr{A},\mathscr{C})$, $\mathscr{C}^*$ the corresponding saturated set of attacks and $S_1,S_2 \subseteq \mathscr{A}$. We say that: S_1 is **conflict-free** iff there is no argument $a \in S_1$ s.t. $(S_1,a) \in \mathscr{C}^*$. S_1 attacks S_2 iff there exists $a \in S_2$ s.t. $(S_1,a) \in \mathscr{C}^*$[3]. S_1 defends an argument a iff for every $S_2 \subseteq \mathscr{A}$ s.t. $(S_2,a) \in \mathscr{C}$, we have that $(S_1,S_2) \in \mathscr{C}^*$. S_1 is said to be **admissible** if each argument in S_1 is defended by S_1. An admissible set S_1 is called a **preferred extension** if there is no admissible set $S_2 \subseteq \mathscr{A}$, $S_1 \subset S_2$. A conflict-free set S_1 is a **stable extension** if S_1 attacks all arguments in $\mathscr{A} \setminus S_1$. An admissible set S_1 is called a **grounded extension** if S_1 is minimum (w.r.t. $\subseteq$) s.t. it contains every argument defended by S_1.*

The set of all preferred (resp. stable and grounded) extensions of an AF $\mathfrak{AS}$ is denoted by $Ext_p(\mathfrak{AS})$ (resp. $Ext_s(\mathfrak{AS})$ and $Ext_g(\mathfrak{AS})$). The output of an AF for an argumentation semantics is $Output_x(\mathfrak{AS}_{\mathscr{K}}) = \bigcap_{E \in Ext_x(\mathfrak{AS}_{\mathscr{K}})} Concs(E)$ where $x \in \{s,p,g\}$.

Example 5 (Cont'd Example 4)**.** *The preferred (resp. stable) extensions of $Ext_p(\mathfrak{AS}_{\mathscr{K}})$ (resp. $Ext_s(\mathfrak{AS}_{\mathscr{K}})$) are $E_1 = \{a_2,a_3,a_5,a_6\}$, $E_2 = \{a_1,a_2,a_4,a_5\}$ and $E_3 = \{a_1,a_3,a_4,a_5\}$. The grounded extension is $E_{GE} = \{a_5\}$*

[3] By abuse of notation, we will use the notation $(S_1,S_2) \in \mathscr{C}^*$ for the case when S_1 attacks a set of arguments S_2.

We now show that there is a correspondence between the set of preferred (resp. stable) extensions and the set of repairs.

Proposition 1 (Preferred & Stable Characterisation). *Let $\mathfrak{AS}_{\mathscr{K}}$ be an AF and $x \in \{s, p\}$. Then, $Ext_x(\mathfrak{AS}_{\mathscr{K}}) = \{Arg(A') \mid A' \in Repair(\mathscr{K})\}$*

Example 6 (Cont'd Example 5). *As explained in Proposition 1, we have a correspondence between repairs and preferred (resp. stable) extensions. Hence:*

- $E_1 = Arg(\{contains(m, saltC), contains(m, yogurt), edible(m)\})$,
- $E_2 = Arg(\{contains(m, sugar), contains(m, saltC), notSoup(m), edible(m)\})$
- $E_3 = Arg(\{contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\})$.

Next, we show the equivalence between the non-attacked arguments and the arguments generated from free-facts.

Corollary 1 (Non-attacked characterisation). *Let $\mathscr{K}$ be a KB, $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$ and $a \in \mathscr{A}$. There exists no S s.t. $(S, a) \in \mathscr{C}$ iff $Prem(a) \subseteq \bigcap_{R \in Repair(\mathscr{K})} R$.*

Note that although it is tempting to say that the non-attacked arguments do not contribute to attacks because they are based on free-facts, this is not true in the general case. In the next proposition, we show that the grounded extension is equal to the intersection of the preferred extensions. Note that the grounded extension is always included in the intersection of the preferred extensions in the general case.

Proposition 2 (Grounded & Preferred). *Let $\mathfrak{AS}_{\mathscr{K}}$ be an AF and $Ext_g(\mathfrak{AS}_{\mathscr{K}}) = \{E_{GE}\}$. Then $E_{GE} = \bigcap_{E \in Ext_p(\mathfrak{AS}_{\mathscr{K}})} E$*

We show the equality between the grounded extension and arguments generated by the intersection of all the repairs.

Proposition 3 (Grounded Characterisation). *Let $\mathfrak{AS}_{\mathscr{K}}$ be an AF and $Ext_g(\mathfrak{AS}_{\mathscr{K}}) = \{E_{GE}\}$. Then $E_{GE} = Arg(\bigcap_{R \in Repair(\mathscr{K})} R)$.*

Example 7 (Cont'd Example 5). *We have that the grounded extension $E_{GE} = E_1 \cap E_2 \cap E_3 = \{a_5\}$ and that the grounded extension is $E_{GE} = \{a_5\} = Arg(\{edible(m)\})$.*

We now show that for any arbitrary KB $\mathscr{K}$, the generated AF $\mathfrak{AS}_{\mathscr{K}}$ does not contain self-attacking arguments.

Proposition 4 (Self-attacking Arguments). *Let $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$ be an AF. There is no $(S, t) \in \mathscr{C}$ s.t. $t \in S$.*

In Proposition 5 below, we show that an attacked argument is always defended by a set of arguments.

Proposition 5 (Defense). *Let $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$ be an AF. If there is $(S, t) \in \mathscr{C}$ then there exists $(S', s) \in \mathscr{C}$ s.t. $s \in S$.*

We introduce the definition of cycle for our AF.

Definition 8 (Cycle). *A cycle in $\mathfrak{AS} = (\mathscr{A}, \mathscr{C})$ is a sequence of attacks in $\mathscr{C}$ of the form $((S_1, t_1), \ldots, (S_n, t_n))$ s.t. for every $i \in \{1, \ldots, n-1\}, t_i \in S_{i+1}$ and $t_n \in S_1$.*

The following corollary follows directly from Proposition 5 and shows that if the number of arguments is finite then there exists at least one cycle in the framework.

Corollary 2 (Cycle Existence). *Let $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$ be an AF. If $|\mathscr{A}|$ is finite and non empty and $\mathscr{C} \neq \emptyset$ then there exists a cycle in $\mathfrak{AS}_{\mathscr{K}}$.*

Example 8 (Cont'd Example 4). *The sequence of attacks $((\{a_2, a_3\}, a_1), (\{a_1, a_3\}, a_2))$ is a cycle in $\mathfrak{AS}_{\mathscr{K}}$.*

Contrary to the AF described in Definition 4 where the number of arguments can be exponential even in the case where the set of rules is empty, we show that in the framework described in Definition 5, the set of arguments is at most equal to the number of facts.

Observation 1 (Argument upper-bound). *Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ s.t. $\mathscr{R} = \emptyset$ and $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$, then $|\mathscr{A}| \leq |\mathscr{F}|$.*

In the next proposition, we show an upper bound to the number of attacks w.r.t. the number of arguments.

Proposition 6 (Attack upper-bound). *Let $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$. If $|\mathscr{A}| = n$ then $|\mathscr{C}| \leq n \times (2^{n-1} - 1)$.*

In the general case, this upper-bound on attacks is almost never reached because of the minimality condition on attacks.

3.2. Rationality postulates

In this section, we prove that the framework we propose in this paper satisfies the rationality postulates for instantiated AFs. We first prove the indirect consistency postulate.

Proposition 7 (Indirect consistency). *Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a KB, $\mathfrak{AS}_{\mathscr{K}}$ be the corresponding AF and $x \in \{s, p, g\}$. Then, for every $E \in Ext_x(\mathfrak{AS}_{\mathscr{K}}), Concs(E)$ is $\mathscr{R}$-consistent and $Output_x(\mathfrak{AS}_{\mathscr{K}})$ is $\mathscr{R}$-consistent.*

Proof. Let E be a stable or preferred extension of $\mathfrak{AS}_{\mathscr{K}}$. From Proposition 1, there exists a repair $A' \in Repair(\mathscr{K})$ s.t. $E = Arg(A')$. By definition, $Concs(E) = \mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(A')$. Formally, $\mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(\mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(A')) = \mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(Concs(E))$. Since $\mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}$ is idempotent, this means that we have $\mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(A') = \mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(Concs(E))$. Since it holds that $\mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(A') \not\models \bot$, then $\mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(Concs(E)) \not\models \bot$ and $Concs(E)$ is $\mathscr{R}$-consistent.

Let us consider the case of grounded semantics. Denote E_{GE} the grounded extension of $\mathfrak{AS}_{\mathscr{K}}$. We just proved that for every $E \in Ext_p(\mathfrak{AS}_{\mathscr{K}})$, it holds that $\mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(Concs(E)) \not\models \bot$. Since the grounded extension is a subset of the intersection of all the preferred extensions, and since there is at least one preferred extension [22], say E_1, then $E_{GE} \subseteq E_1$. Since $\mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(Concs(E_1)) \not\models \bot$ then $\mathbb{SAT}_{\mathscr{R} \cup \mathscr{N}}(Concs(E_{GE})) \not\models \bot$ and $Concs(E_{GE})$ is $\mathscr{R}$-consistent.

Consider the case of stable or preferred semantics. We prove that $Output_x(\mathfrak{AS}_{\mathscr{K}})$ is $\mathscr{R}$-consistent. Recall that $Output_x(\mathfrak{AS}_{\mathscr{K}}) = \bigcap_{E \in Ext_x(\mathfrak{AS}_{\mathscr{K}})} Concs(E)$. Since every KB has at least one repair then, there is at least one stable or preferred extension E. From the definition of the output, $Output_x(\mathfrak{AS}_{\mathscr{K}}) \subseteq Concs(E)$. Since $Concs(E)$ is $\mathscr{R}$-consistent then $Output_x(\mathfrak{AS}_{\mathscr{K}})$ is $\mathscr{R}$-consistent. Note that since there is only one grounded extension, we get that $\mathbb{SAT}_{\mathscr{R}}(Output_g(\mathfrak{AS}_{\mathscr{K}})) = \mathbb{SAT}_{\mathscr{R}}(Concs(E_{GE}))$. □

Since our instantiation satisfies indirect consistency then it satisfies direct consistency. Indeed, if a set is $\mathscr{R}$-consistent, then it is consistent. Thus, we obtain the following corollary.

Corollary 3 (Direct consistency). *Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a KB, $\mathfrak{AS}_{\mathscr{K}}$ the corresponding AF and $x \in \{s, p, g\}$. Then, for every $E \in Ext_x(\mathfrak{AS}_{\mathscr{K}}), Concs(E) \not\models \bot$ and $Output_x(\mathfrak{AS}_{\mathscr{K}}) \not\models \bot$.*

Proposition 8 shows that the AF satisfies Closure.

Proposition 8 (Closure). *Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a KB, $\mathfrak{AS}_{\mathscr{K}}$ be the corresponding AF and $x \in \{s, p, g\}$. Then, for every $E \in Ext_x(\mathfrak{AS}_{\mathscr{K}}), Concs(E) = \mathbb{SAT}_{\mathscr{R}}(Concs(E))$ and $Output_x(\mathfrak{AS}_{\mathscr{K}}) = \mathbb{SAT}_{\mathscr{R}}(Output_x(\mathfrak{AS}_{\mathscr{K}}))$.*

	Existing Framework $\mathfrak{AS}'_{\mathscr{K}}$			New Framework $\mathfrak{AS}_{\mathscr{K}}$					
$\mathscr{K}$	# Arg.	# Att.	Gen. Time	# Arg.	% Arg. ↓	# Att.	% Att. ↓	Gen. Time	% Time ↓
A_1	22	128	160	5	77,27	6	93,75	276,00	-81,48
A_2	25	283	133	7	72,00	8	92,93	342,00	-183,57
A_3	85	1472	399,5	7	91,76	9	99,26	369,50	1,66
B	5967	11542272	533089	14	99.77	20.5	99.99	7814.5	98.08

Table 1. Comparison of the median number of arguments, attacks and generation time needed (in ms) between the two frameworks $\mathfrak{AS}_{\mathscr{K}}$ and $\mathfrak{AS}'_{\mathscr{K}}$ on the sets of KBs A_1, A_2, A_3 and B.

4. Empirical Analysis

We now compare our approach with the existing AF for Datalog w.r.t. the number of arguments and the number of attacks. All experiments were conducted on a Debian computer with an Intel Xeon E5-1620 processor and 64GBs of RAM. We chose to work with the set of KBs extracted from the study of [28,26]. These inconsistent KBs are composed of two main sets:

- A set A composed of 108 KBs. A is further split into three smaller sets of KBs: A set A_1 of 31 KBs without rules, two to seven facts, and one to three negative constraints, a set A_2 of 51 KBs generated by fixing the size of the set of facts and adding negative constraints until saturation and a set A_3 of 26 KBs with ternary negative constraints, three to four facts and one to three rules.
- A set B of 26 KBs with eight facts, six rules and one or two negative constraints. This set contains more free-facts than the KBs in set A.

For each of these two sets, we compare the number of arguments and attacks of the new framework defined in Definition 5 with the one of Definition 4. We provided a tool based on the Graph of Atom Dependency defined by [18] and the Graal Java Toolkit [6] for generating the new AF from an inconsistent KB expressed in the DLGP format. The tool is available online at: https://www.dropbox.com/sh/dlpmr07gqvpuc61/AABDgwfHJRNVYcsqpDg7kMfEa?dl=0

4.1. Experimental results

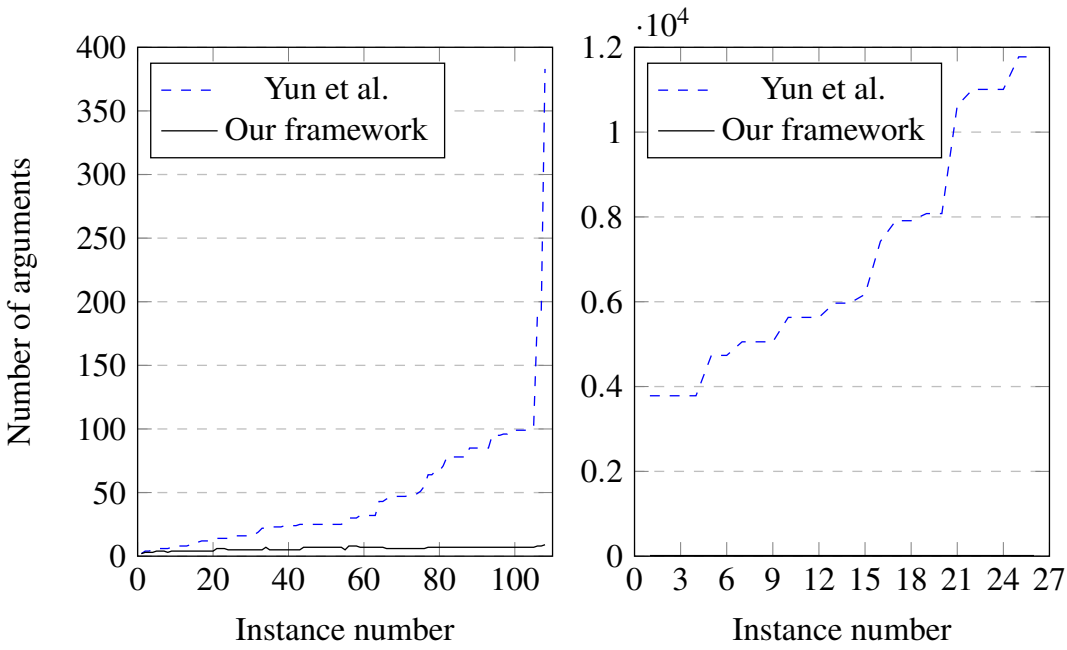

Figure 1. Comparison of the number of arguments between the two AFs on sets A (left) and B (right).

In Table 1, we show the number of arguments and attacks of the two frameworks $\mathfrak{AG}$ and $\mathfrak{AG}'$ for the two sets of KBs (A and B). We make the following observations: First, contrary to the framework $\mathfrak{AG}'$, there is no exponential increase in the number of arguments with the number of free-facts in $\mathfrak{AG}$ as seen with the KBs in set B. Moreover, for all the KBs considered in sets A and B, the number of arguments and attacks in $\mathfrak{AG}$ is less or equal to the number of arguments and attacks in $\mathfrak{AG}'$. We can notice that the efficiency brought by this new framework is obvious in the case where the KBs contain more free facts (see Figures 1). Second, when the set of facts and the set of rules are fixed and only the set of negative constraint is modified, the number of arguments of $\mathfrak{AG}$ seems to be unchanged whereas in $\mathfrak{AG}'$, it is varying. $\mathfrak{AG}'$ is also much denser than $\mathfrak{AG}$. Indeed, the median density[4] of $\mathfrak{AG}'$ is 26.34% and 31,03% whereas the median density of $\mathfrak{AG}$ is 4,69% and 0.02% for the set A and B respectively. Third, the generation of $\mathfrak{AG}$ is slower than the one for $\mathfrak{AG}'$ when the number of arguments and attacks is relatively low (see A_1, A_2 and Figure 2) but when the number of arguments and attacks increases, we can notice that the generation of $\mathfrak{AG}$ is much faster (see B and Figure 2).

[4]The density is equal to the number of attacks divided by the maximum number of possible attacks. In the case of a directed graph, the maximum number of attacks is given by $n(n-1)$ where n is the number of nodes. In the case of $\mathfrak{AG}$, we use the formula in Proposition 6 to obtain the maximum number of attacks.

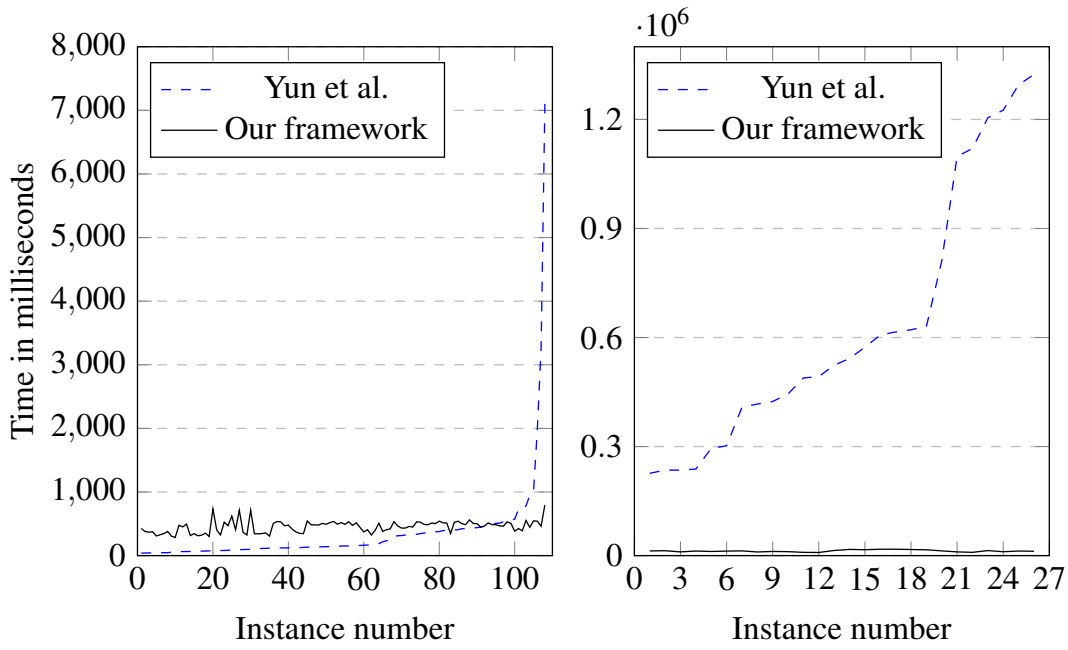

Figure 2. Generation time needed between the two AFs on sets A (left) and B (right).

5. Conclusion

We introduced a new logic-based AF with n-ary attacks that is built over an inconsistent Datalog KB and analysed the syntactic and semantic properties of this AF. We showed that it has all of the desirable properties of the existing AF for Datalog. Namely: (1) the rationality postulates for instantiated AFs defined by [13] are satisfied, (2) there is a bijection between the stable (resp. preferred) extensions and the sets of arguments generated from the repairs, (3) the grounded extension is equal to both the intersection of the preferred extensions and the set of arguments generated from free-facts, (4) the non-attacked arguments are generated from the free-facts and for each attacked argument there exists a set of arguments that defends it. (5) there are no self-attacking arguments and there is at least one cycle if the set of arguments is finite, (6) we give an upper-bound on the number of arguments and the number of attacks.

Second, we provided a tool for generating this n-ary AF from a knowledge base expressed in DLGP format and used it to conduct en empirical comparison between this n-ary framework and the existing AF [28,4] w.r.t. the number of arguments, attacks and time needed for the generation. We highlighted that this n-ary framework possesses fewer arguments and attacks than the existing framework mainly because it avoids the problem of the exponential increase of arguments when free-facts are added. Moreover, although the generation of the new framework is slower than the existing framework when the number of arguments and attacks is low, as soon as the number of arguments and attacks increases, the generation of n-ary framework is faster than for the existing framework.

References

[1] L. Amgoud. Postulates for logic-based argumentation systems. *Int. J. Approx. Reasoning*, 55(9):2028–2048, 2014.

[2] L. Amgoud and J. Ben-Naim. Argumentation-based Ranking Logics. In *AAMAS 2015*, pages 1511–1519, 2015.

[3] A. Arioua, M. Croitoru, and P. Buche. DALEK: A Tool for Dialectical Explanations in Inconsistent Knowledge Bases. In *COMMA 2016*, pages 461–462, 2016.

[4] A. Arioua, M. Croitoru, and S. Vesic. Logic-based argumentation with existential rules. *Int. J. Approx. Reasoning*, 90:76–106, 2017.

[5] J.-F. Baget, S. Benferhat, Z. Bouraoui, M. Croitoru, M.-L. Mugnier, O. Papini, S. Rocher, and K. Tabia. Inconsistency-Tolerant Query Answering: Rationality Properties and Computational Complexity Analysis. In *JELIA 2016*, pages 64–80, 2016.

[6] J.-F. Baget, M. Leclère, M.-L. Mugnier, S. Rocher, and C. Sipieter. Graal: A Toolkit for Query Answering with Existential Rules. In *RuleML 2015*, pages 328–344, 2015.

[7] J.-F. Baget, M. Leclère, M.-L. Mugnier, and E. Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.

[8] P. Besnard, A. J. García, A. Hunter, S. Modgil, H. Prakken, G. R. Simari, and F. Toni. Introduction to structured argumentation. *Argument & Computation*, 5(1):1–4, 2014.

[9] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artif. Intell.*, 128(1-2):203–235, 2001.

[10] M. Bienvenu. On the Complexity of Consistent Query Answering in the Presence of Simple Ontologies. In *AAAI 2012*, 2012.

[11] A. Bondarenko, F. Toni, and R. A. Kowalski. An Assumption-Based Framework for Non-Monotonic Reasoning. In *LPNMR*, pages 171–189, 1993.

[12] C. Bourgaux. *Inconsistency Handling in Ontology-Mediated Query Answering*. PhD thesis, Université Paris-Saclay, Paris, Sept. 2016.

[13] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

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

[15] P. M. 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.

[16] G. Flouris and A. Bikakis. A comprehensive study of argumentation frameworks with sets of attacking arguments. *IJAR*, 109:55–86, June 2019.

[17] A. J. García and G. R. Simari. Defeasible Logic Programming: An Argumentative Approach. *TPLP*, 4(1-2):95–138, 2004.

[18] A. Hecham, P. Bisquert, and M. Croitoru. On the Chase for All Provenance Paths with Existential Rules. In *RuleML+RR 2017*, pages 135–150, 2017.

[19] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-Tolerant Semantics for Description Logics. In *RR 2010*, pages 103–117, 2010.

[20] S. Modgil and H. Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[21] S. H. Nielsen and S. Parsons. Computing Preferred Extensions for Argumentation Systems with Sets of Attacking Arguments. In *COMMA 2006*, pages 97–108, 2006.

[22] S. H. Nielsen and S. Parsons. A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In N. Maudet, S. Parsons, and I. Rahwan, editors, *Argumentation in Multi-Agent Systems*, pages 54–73. Springer Berlin Heidelberg, 2007.

[23] M. Ostrowski and T. Schaub. ASP modulo CSP: The clingcon system. *TPLP*, 12(4-5):485–503, 2012.

[24] F. Toni. A tutorial on assumption-based argumentation. *Argument & Computation*, 5(1):89–117, 2014.

[25] B. Yun, M. Croitoru, P. Bisquert, and S. Vesic. Graph Theoretical Properties of Logic Based Argumentation Frameworks. In *AAMAS 2018*, pages 2148–2149, 2018.

[26] B. Yun, S. Vesic, and M. Croitoru. Toward a More Efficient Generation of Structured Argumentation Graphs. In *COMMA 2018*, 2018.

[27] B. Yun, S. Vesic, M. Croitoru, and P. Bisquert. Inconsistency Measures for Repair Semantics in OBDA. In *IJCAI 2018*, pages 1977–1983, 2018.

[28] 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.