Sets of Attacking Arguments for Inconsistent Datalog Knowledge Bases

Bruno YUN [1] Srdjan VESIC [2] Madalina CROITORU [3]

September 2020

1. University of Aberdeen
2. CNRS
3. University of Montpellier

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$
- An **argument** $a = (H, C)$ such that :
  - **Consistency :** $H \subseteq \mathcal{F}$ and $\mathcal{R}$-consistent.
  - **Reasoning :** $C \subseteq \mathbb{SAT}_{\mathcal{R}}(H)$.
  - **Minimality :** $\nexists H' \subset H$ s.t. $C \subseteq \mathbb{SAT}_{\mathcal{R}}(H')$.

  $H = Supp(a)$ and $C = Conc(a)$.

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$
- An **argument** $a = (H, C)$ such that :
  - **Consistency :** $H \subseteq \mathcal{F}$ and $\mathcal{R}$-consistent.
  - **Reasoning :** $C \subseteq \mathbb{SAT}_{\mathcal{R}}(H)$.
  - **Minimality :** $\nexists H' \subset H$ s.t. $C \subseteq \mathbb{SAT}_{\mathcal{R}}(H')$.

  $H = Supp(a)$ and $C = Conc(a)$.
- There are 33 arguments in this example.

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$

- $\mathcal{A} = \{a_1, a_2, \ldots\}$
  - $a_1 = (\{contains(m, saltC), contains(m, yogurt)\}, \{tzaziki(m)\})$
  - $a_2 = (\{notSoup(m)\}, \{notSoup(m)\})$

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$

- **Attacks :** a attacks b iff there exists $\varphi \in Supp(b)$ s.t. $Conc(a) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent.

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$

- **Attacks :** a attacks b iff there exists $\varphi \in Supp(b)$ s.t. $Conc(a) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent.
  We get an argumentation framework (AF) $\mathcal{AS}_{\mathcal{K}} = (\mathcal{A}, \mathcal{C})$.

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$

- **Attacks :** a attacks b iff there exists $\varphi \in Supp(b)$ s.t. $Conc(a) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent.
  We get an argumentation framework (AF) $\mathcal{AS}_{\mathcal{K}} = (\mathcal{A}, \mathcal{C})$.
  - $a_1 = (\{contains(m, saltC), contains(m, yogurt)\}, \{tzaziki(m)\})$
  - $a_2 = (\{notSoup(m)\}, \{notSoup(m)\})$

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$

- **Attacks :** a attacks b iff there exists $\varphi \in Supp(b)$ s.t. $Conc(a) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent.
  We get an argumentation framework (AF) $\mathcal{AS}_{\mathcal{K}} = (\mathcal{A}, \mathcal{C})$.
  - $a_1 = (\{contains(m, saltC), contains(m, yogurt)\}, \{tzaziki(m)\})$
  - $a_2 = (\{notSoup(m)\}, \{notSoup(m)\})$

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$

- **Attacks :** a attacks b iff there exists $\varphi \in Supp(b)$ s.t. $Conc(a) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent.
  We get an argumentation framework (AF) $\mathcal{AS}_{\mathcal{K}} = (\mathcal{A}, \mathcal{C})$.
  - $a_1 = (\{contains(m, saltC), contains(m, yogurt)\}, \{tzaziki(m)\})$
  - $a_2 = (\{notSoup(m)\}, \{notSoup(m)\})$

Background

- Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that :
  - $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
  - $\mathcal{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
  - $\mathcal{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \bot), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \bot)\}$

- **Attacks :** a attacks b iff there exists $\varphi \in Supp(b)$ s.t. $Conc(a) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent.
  We get an argumentation framework (AF) $\mathcal{AS}_{\mathcal{K}} = (\mathcal{A}, \mathcal{C})$.
  - $a_1 = (\{contains(m, saltC), contains(m, yogurt)\}, \{tzaziki(m)\})$
  - $a_2 = (\{notSoup(m)\}, \{notSoup(m)\})$

The Idea Behind The Paper

- We use instantiated argumentation frameworks to handle inconsistencies. [4]

4. Croitoru & Vesic, What Can Argumentation Do for Inconsistent Ontology Query Answering ?

The Idea Behind The Paper

- We use instantiated argumentation frameworks to handle inconsistencies. [4]
- We can make two observations :
  - We have a huge number of arguments !
  - It satisfies the rationality postulates

4. Croitoru & Vesic, What Can Argumentation Do for Inconsistent Ontology Query Answering ?

The Idea Behind The Paper

- We use instantiated argumentation frameworks to handle inconsistencies. [4]
- We can make two observations :
  - We have a huge number of arguments !
    → Can we make a more efficient AF ?
  - It satisfies the rationality postulates
    →Can we ensure that we keep desirable properties ?

4. Croitoru & Vesic, What Can Argumentation Do for Inconsistent Ontology Query Answering ?

The Idea Behind The Paper

- We use instantiated argumentation frameworks to handle inconsistencies. [4]
- We can make two observations :
  - We have a huge number of arguments !
    → Can we make a more efficient AF ?
  - It satisfies the rationality postulates
    →Can we ensure that we keep desirable properties ?
- **Aim of the paper :** Provide, analyse and evaluate a more efficient argumentation framework based on sets of attacking arguments frameworks (SETAFs).

4. Croitoru & Vesic, What Can Argumentation Do for Inconsistent Ontology Query Answering ?

Plan

A SETAF-based Framework

The SETAF for $\mathcal{K}$ is $\mathcal{AS}_{\mathcal{K}} = (\mathcal{A}, \mathcal{C})$ with $\mathcal{C} \subseteq (2^{\mathcal{A}} \setminus \{\emptyset\}) \times \mathcal{A}$ s.t. :

- An argument $a \in \mathcal{A}$ is either
  1. a fact f, where $f \in \mathcal{F}$ s.t. $Conc(a) = f$ and $Prem(a) = \{f\}$
  2. $a_1, \ldots, a_n \rightarrow f'$ where $a_1, \ldots, a_n \in \mathcal{A}$ s.t. there exists $r \in \mathcal{R}$ that can be applied 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 $\mathcal{R}$-consistent.

Example

There are six arguments :

$a_1 = contains(m, sugar)$, $a_2 = contains(m, saltC)$,
$a_3 = contains(m, yogurt)$, $a_4 = notSoup(m)$, $a_5 = edible(m)$ and
$a_6 = a_2, a_3 \rightarrow tzaziki(m)$.

A SETAF-based Framework

The SETAF for $\mathcal{K}$ is $\mathcal{AS}_{\mathcal{K}} = (\mathcal{A}, \mathcal{C})$ with $\mathcal{C} \subseteq (2^{\mathcal{A}} \setminus \{\emptyset\}) \times \mathcal{A}$ s.t. :

- $(X, a) \in \mathcal{C}$ iff X is minimal for set inclusion s.t. $\bigcup_{x \in X} Prem(x)$ is $\mathcal{R}$-consistent and there exists $\varphi \in Prem(a)$ s.t. $(\bigcup_{x \in X} Conc(x)) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent.

Example

There are 11 attacks.

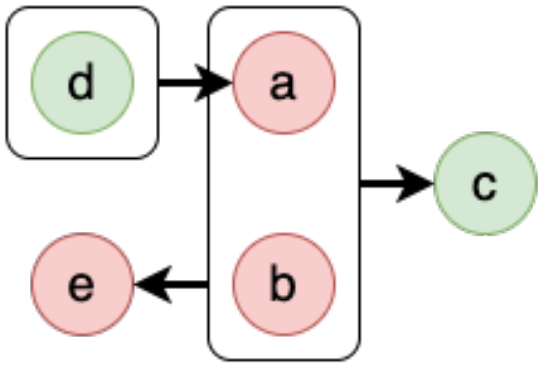
$a_1 = contains(m, sugar)$

$a_2 = contains(m, saltC)$

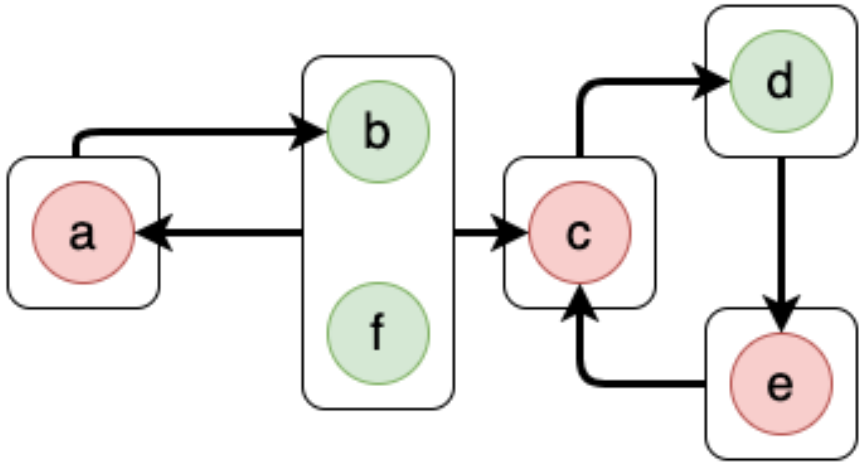
$a_3 = contains(m, yogurt)$

It holds that $(\{a_1, a_2\}, a_3) \in \mathcal{C}$

Argumentation Semantics for SETAFs



Example of an **admissible** extension
{d,b,c} is a **complete** extension
{d,b} is the **grounded** extension



Example of a **preferred** extension (no **stable** extensions)

Some Theoretical Properties

Proposition (Preferred & Stable Characterisation)

For any KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *and SETAF* $\mathcal{AS}_{\mathcal{K}}$*, it holds that :*
$Ext_s(\mathcal{AS}_{\mathcal{K}}) = Ext_p(\mathcal{AS}_{\mathcal{K}}) = \{Arg(A') \mid A' \in Repair(\mathcal{K})\}$

Example

We have 3 preferred/stable extensions :

- $E_1 = Arg(\{contains(m, saltC), contains(m, yogurt), edible(m)\}) = \{a_2, a_3, a_5\}$
- $E_2 = Arg(\{contains(m, sugar), contains(m, saltC), notSoup(m), edible(m)\}) = \{a_1, a_2, a_4, a_5\}$
- $E_3 = Arg(\{contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}) = \{a_1, a_3, a_4, a_5\}$

Some Theoretical Properties

Proposition (Non-attacked characterisation)

For any KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$*, SETAF* $\mathcal{AS}_{\mathcal{K}} = (\mathcal{A}, \mathcal{C})$ *and* $a \in \mathcal{A}$*. There exists no* S *s.t.* $(S, a) \in \mathcal{C}$ *iff* $Prem(a) \subseteq \bigcap_{R \in Repair(\mathcal{K})} R$.

Proposition (Attack upper-bound)

For any KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *and SETAF* $\mathcal{AS}_{\mathcal{K}} = (\mathcal{A}, \mathcal{C})$*. If* $|\mathcal{A}| = n$ *then* $|\mathcal{C}| \leq n \times (2^{n-1} - 1)$.

- A set of arguments will not attack one of its members.
- An argument that is attacked will always be defended
- The instantiated SETAF framework satisfies the indirect, direct consistency and closure rationality postulates.

Plan

The Dataset

We evaluate the new framework on an existing set of KBs [5]. These inconsistent KBs are composed of four sets :

- 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
- 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.

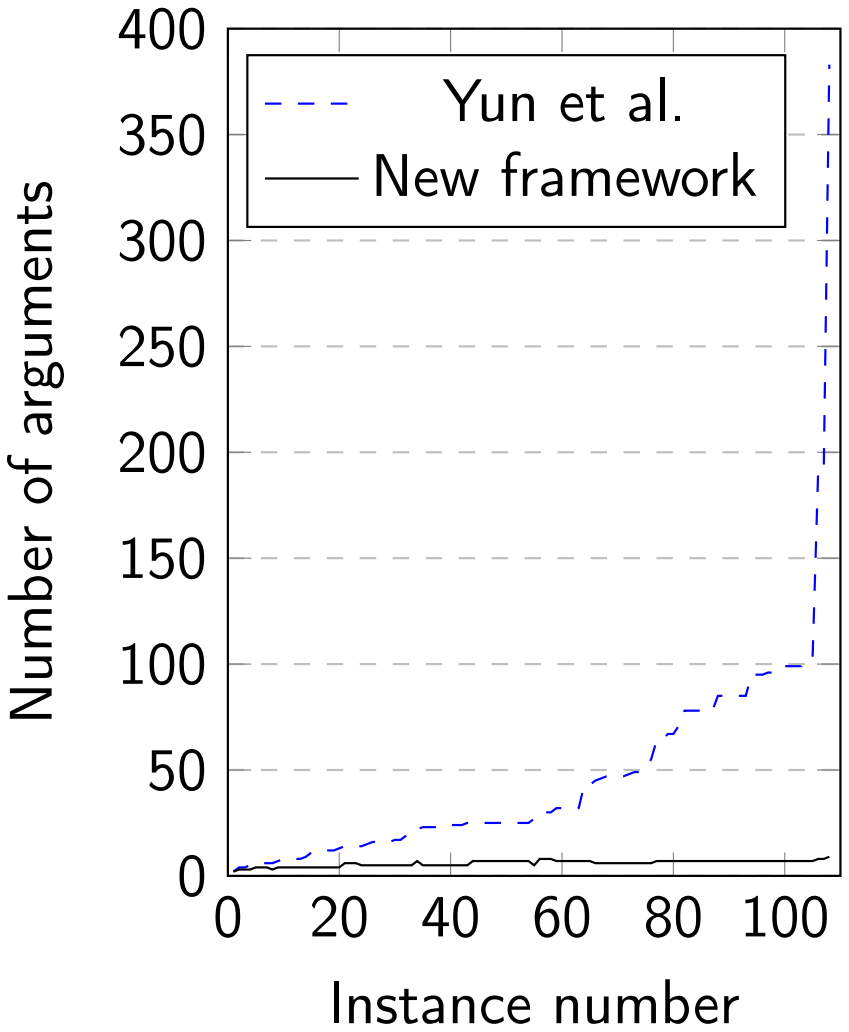
5. Yun et al. - A Structural Benchmark for Logical Argumentation Frameworks (IDA 2017)

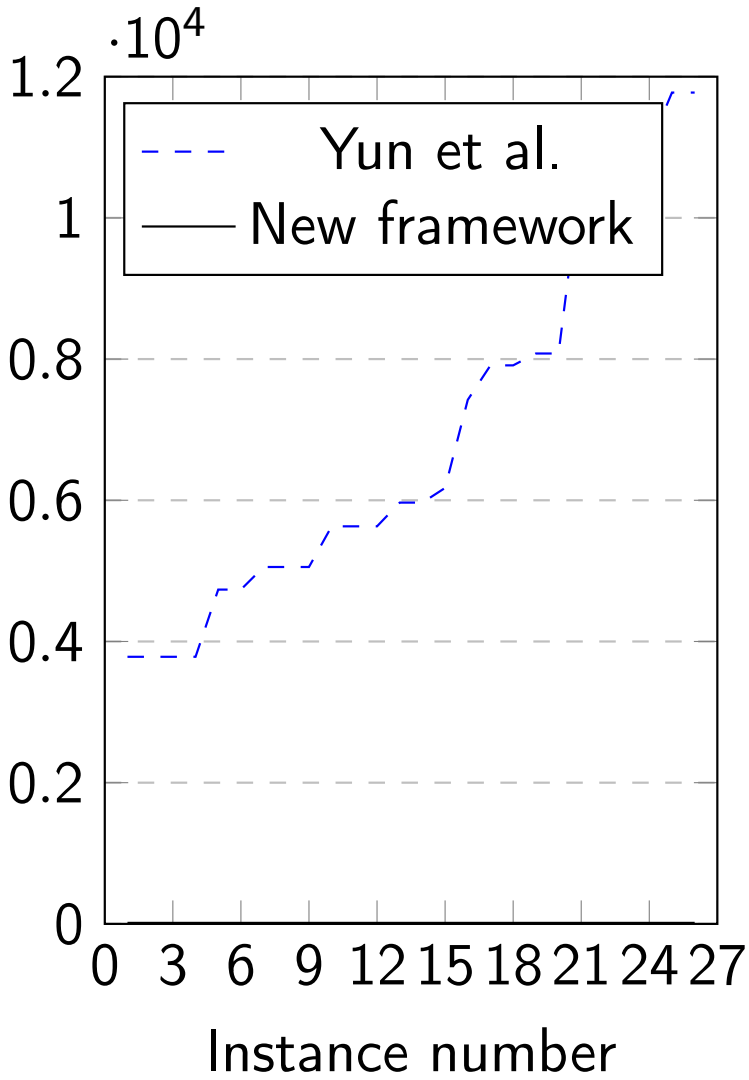
The Generation

	Existing Framework		
$\mathcal{K}$	# Arg.	# Att.	Gen. Time
A_1	22	128	160
A_2	25	283	133
A_3	85	1472	399,5
B	5967	11542272	533089

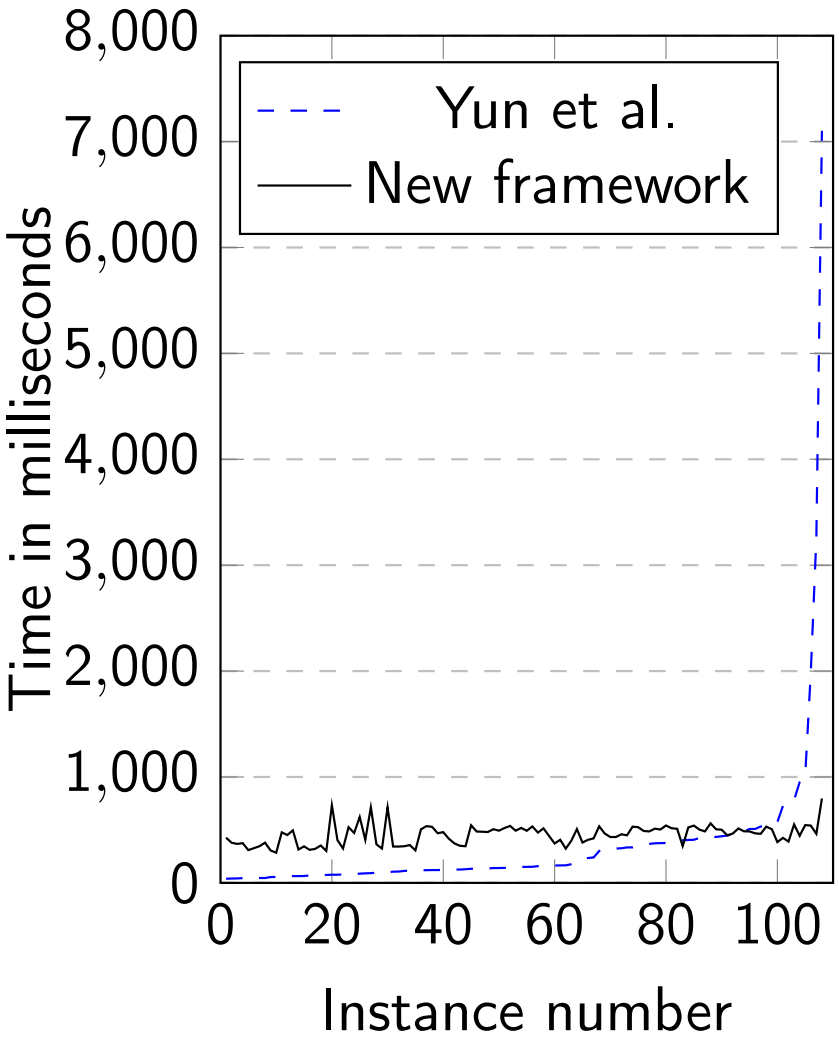
	New SETAF Framework $\mathcal{AS}_{\mathcal{K}}$					
$\mathcal{K}$	# Arg.	% Arg. $\downarrow$	# Att.	% Att. $\downarrow$	Gen. Time	% Time $\downarrow$
A_1	5	77,27	6	93,75	276,00	-81,48
A_2	7	72,00	8	92,93	342,00	-183,57
A_3	7	91,76	9	99,26	369,50	1,66
B	14	99.77	20.5	99.99	7814.5	98.08

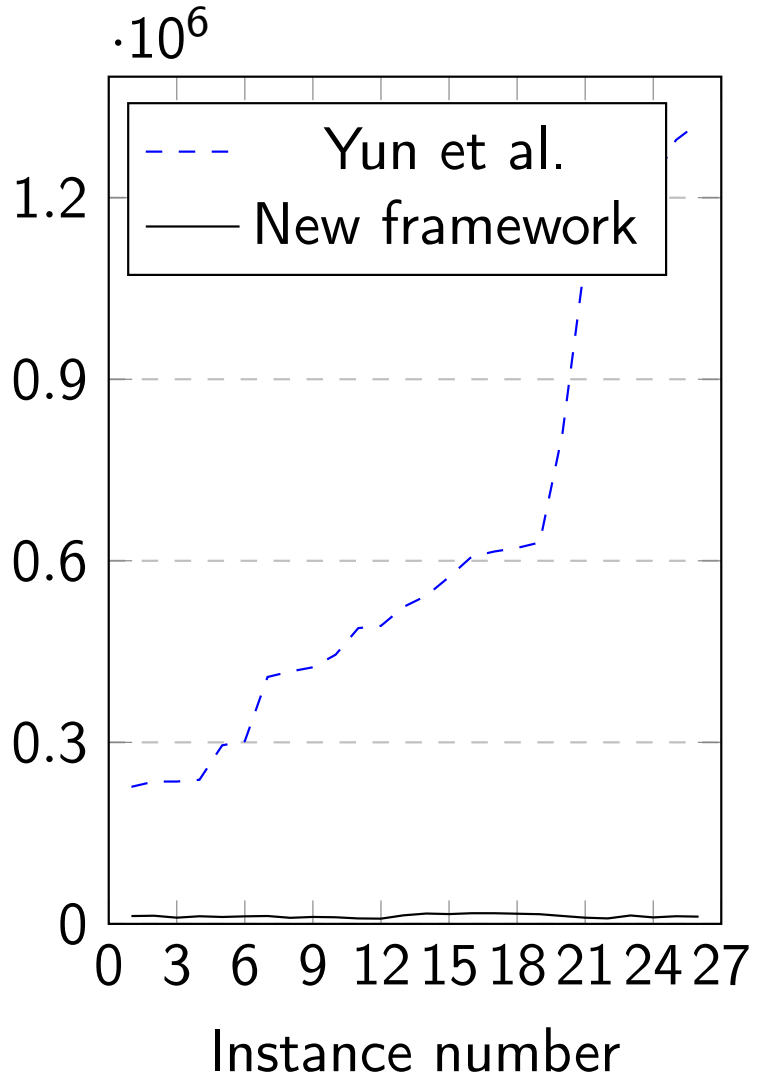
The Generation





The Generation





Conclusion

We provide a workflow for efficiently generating a SETAF framework from an inconsistent Datalog KB :

- We study the structural properties of this new framework
- Give an upper-bound to the number of arguments and attacks
- Prove the satisfaction of the rationality postulates
- Showed a comparison of the existing and new framework w.r.t. the number of arguments and generation time.