Acceptance in Incomplete Argumentation Frameworks

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

HIIT HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

Dorothea Baumeister Matti Järvisalo Daniel Neugebauer Andreas Niskanen Jörg Rothe

hhu Heinrich Heine Universität Düsseldorf

Motivation: The study of **computational models of argumentation** is an active and vibrant area of modern **AI** research. **Incomplete argumentation frameworks (IAFs)** generalize Dung's argumentation frameworks for **reasoning under uncertainty**.

Contributions: **Complexity analysis** of **acceptance problems in IAFs** under various central AF semantics. **Design of algorithms** for solving concrete instances of acceptance problems: direct **SAT encodings** and SAT-based **CEGAR algorithms**. **Strong refinements** for CEGAR algorithms by analyzing redundant atomic changes in IAFs. **System implementation** and **empirical evaluation**: SAT-based approach viable in practice.

INCOMPLETE ARGUMENTATION FRAMEWORKS

Argumentation Framework (AF)

A directed graph $AF = (A, R)$:

- A is the set of **arguments**
- $R \subseteq A \times A$ is the **attack relation**

Semantics σ define σ-**extensions**

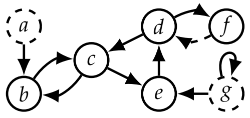
- jointly acceptable sets of arguments
- required to be **conflict-free**
- other desired properties give rise to different semantics: admissible, stable, complete, grounded, preferred

Incomplete Argumentation Framework (IAF)

$IAF = (A, A^?, R, R^?)$, where

- A and R are **definite** arguments and attacks,
- $A^?$ and $R^?$ are **uncertain** arguments and attacks,

$R, R^? \subseteq (A \cup A^?) \times (A \cup A^?)$.



Acceptance Problems in IAFs

Completion: standard AF containing the fixed part and a valid subset of uncertain elements

An argument $a \in A$ is

- **credulously** accepted (CA): a is in **some** extension
- **skeptically** accepted (SA): a is in **all** extensions
  - + existence of an extension (ExSA)

Acceptance of $a \in A$ holds

- **possibly** (P) if it holds for **some** completion
- **necessarily** (N) if it holds for **all** completions

SAT-BASED ALGORITHMS

Input: $IAF = (A, A^?, R, R^?)$, $a \in A$.

Basis: SAT Encodings

Boolean variables:

- y_a for each argument a, $r_{a,b}$ for each attack (a, b)
  - true iff element included in completion
- x_a for each argument a
  - true iff argument included in σ-extension
- z_a for each argument a
  - true iff argument attacked by σ-extension

Propositional formulas:

- $\varphi_?(CAF)$ encodes valid completions
- $\varphi_\sigma(CAF)$ for $\sigma \in \{\text{AD}, \text{ST}, \text{CP}, \text{GR}\}$ encodes the semantics

The encoding for complete semantics is

$$\varphi_{\text{CP}}(IAF) = \varphi_{\text{AD}}(IAF) \wedge \bigwedge_{a \in A \cup A^?} \left(\bigwedge_{(b,a) \in R \cup R^?} \left((y_a \wedge y_b \wedge r_{b,a}) \rightarrow z_b \right) \rightarrow x_a \right).$$

PCA and NSA solved via direct SAT calls.

SAT-based CEGAR Algorithms

Algorithms where a **SAT solver** is **called iteratively** and incrementally using different assumptions.

Task: Possible skeptical acceptance under preferred.

```
φ ← ABSTRACTION(IAF, a)                ▷ Initialize abstraction
while true do
    (sat, τ) ← SAT(φ)                  ▷ Solve abstraction
    if sat = false then return reject
    AF* ← COMPLETION(τ)
    (sat, τ) ← CEGAR(CHECK(IAF, AF*, a))  ▷ Counter-
    if sat = false then return accept     ▷ example?
    φ ← φ∧ REFINE(IAF, AF*)            ▷ Exclude completion
end while
```

Strong refinement: take into account which atomic changes preserve the counterexample extension of the completion: e.g. for preferred semantics adding arguments which are attacked by the extension can be safely ignored.

In journal article: SAT encodings, CEGAR algorithms, and strong refinements for all other variants

COMPUTATIONAL COMPLEXITY

s	s-PCA	s-NCA	s-PSA (s-PExSA)	s-NSA (s-NExSA)
CF	$\in$ P	$\in$ P	trivial	trivial
CF$_{\neq\emptyset}$			$\in$ P	$\in$ P
AD	NP-c.	Π_2^p-c.	trivial	trivial
AD$_{\neq\emptyset}$			Σ_2^p-c.	coNP-c. (Π_2^p-c.)
ST	NP-c.	Π_2^p-c.	Σ_2^p-c.	coNP-c. (Π_2^p-c.)
CP	NP-c.	Π_2^p-c.	NP-c.	coNP-c.
GR	NP-c.	coNP-c.	NP-c.	coNP-c.
PR	NP-c.	Π_2^p-c.	Σ_3^p-c.	Π_2^p-c.

- Lower bounds follow from quantifier representations of the problem
  - Possible: $\exists$ completion
  - Necessary: $\forall$ completions
- Upper bounds via reductions from quantified satisfiability

In journal article: full proofs and illustrations of reductions

EMPIRICAL EVALUATION

Implementation: TAEYDENNAE

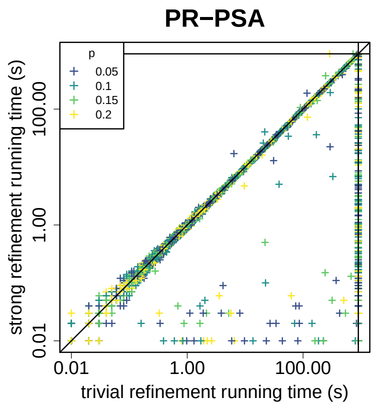
- Solver for acceptance in incomplete AFs
- Direct SAT calls and SAT-based CEGAR algorithms
- Makes use of incremental SAT solving
- Available online in open source:

`bitbucket.org/andreasniskanen/taeydennae`

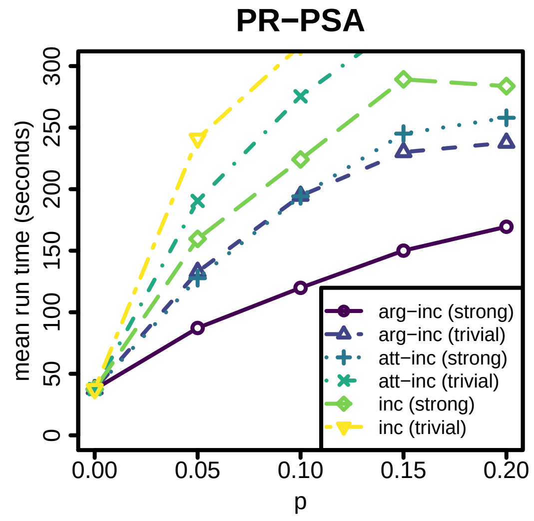
Benchmark instances

4200 IAFs generated from ICCMA'17 AFs

- Parameter: probability p of element being uncertain



Task: Possible skeptical acceptance under preferred



In journal article: Empirical results for other variants, comparison to basic enumeration-based approach