Pakota: A System for Enforcement in Abstract Argumentation

Andreas Niskanen   Johannes P. Wallner   Matti Järvisalo

HIIT, Department of Computer Science
University of Helsinki
Finland

November 10, 2016 @ JELIA 2016, Larnaca, Cyprus

Motivation

Argumentation

- An active area of modern AI research
- Connections to logic, philosophy, and law
- Applications: decision support, legal reasoning, medical diagnostics, etc.

Dung's argumentation frameworks (AFs)

- Central KR formalism in **abstract argumentation**
- Recent interest in **dynamic aspects** of AFs
  - E.g., how to adjust a given AF in light of new knowledge?

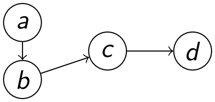
Motivation

Argumentation

- An active area of modern AI research
- Connections to logic, philosophy, and law
- Applications: decision support, legal reasoning, medical diagnostics, etc.

Dung's argumentation frameworks (AFs)

- Central KR formalism in **abstract argumentation**
- Recent interest in **dynamic aspects** of AFs
  - E.g., how to adjust a given AF in light of new knowledge?



Contributions

Pakota

System for solving enforcement via employing MaxSAT and SAT solvers.

- **Describe** the system in detail
  - System architecture overview
  - Features
    - Supported semantics and problem variants
    - MaxSAT and SAT solver interfaces
  - Algorithms
    - Problems in NP: direct MaxSAT encodings
    - Beyond NP: MaxSAT-based CEGAR procedures
  - Input format, usage and options
- **Provide** benchmarks and generators for enforcement
- **Evaluate** the impact of the choice of the MaxSAT solver on scalability

Argumentation Frameworks

Syntax

An argumentation framework (AF) is a directed graph $F = (A, R)$, where

- A is the set of **arguments**
- $R \subseteq A \times A$ is the **attack relation**
  - $a \rightarrow b$ means argument a attacks argument b

Semantics

Define sets of jointly accepted arguments or **extensions**

- a function σ mapping an AF $F = (A, R)$ to a collection $\sigma(F) \subseteq 2^A$
- e.g. **conflict-free:** $E \in cf(F)$ if E is an independent set

Acceptability of arguments

Given an AF $F = (A, R)$ and semantics σ, an argument $a \in A$ is

- **credulously** accepted under σ iff a is in **some** extension
- **skeptically** accepted under σ iff a is in **all** extensions

Argumentation Frameworks

Syntax

An argumentation framework (AF) is a directed graph $F = (A, R)$, where

- A is the set of **arguments**
- $R \subseteq A \times A$ is the **attack relation**
  - $a \rightarrow b$ means argument a attacks argument b

Semantics

Define sets of jointly accepted arguments or **extensions**

- a function σ mapping an AF $F = (A, R)$ to a collection $\sigma(F) \subseteq 2^A$
- e.g. **conflict-free:** $E \in cf(F)$ if E is an independent set

Acceptability of arguments

Given an AF $F = (A, R)$ and semantics σ, an argument $a \in A$ is

- **credulously** accepted under σ iff a is in **some** extension
- **skeptically** accepted under σ iff a is in **all** extensions

Argumentation Frameworks

Syntax

An argumentation framework (AF) is a directed graph $F = (A, R)$, where

- A is the set of **arguments**
- $R \subseteq A \times A$ is the **attack relation**
  - $a \rightarrow b$ means argument a attacks argument b

Semantics

Define sets of jointly accepted arguments or **extensions**

- a function σ mapping an AF $F = (A, R)$ to a collection $\sigma(F) \subseteq 2^A$
- e.g. **conflict-free:** $E \in cf(F)$ if E is an independent set

Acceptability of arguments

Given an AF $F = (A, R)$ and semantics σ, an argument $a \in A$ is

- **credulously** accepted under σ iff a is in **some** extension
- **skeptically** accepted under σ iff a is in **all** extensions

AF Reasoning Tasks

Static computational problems

Direct inference from a given AF—no change involved

- credulous and skeptical acceptance of an argument
- extension enumeration

Many system implementations available!

Dynamic computational problems

How to change a given AF to support new information?

Pakota

First system implementation in its generality for solving instances of

- extension enforcement
- status enforcement

AF Reasoning Tasks

Static computational problems

Direct inference from a given AF—no change involved

- credulous and skeptical acceptance of an argument
- extension enumeration

Many system implementations available!

Dynamic computational problems

How to change a given AF to support new information?

Pakota

First system implementation in its generality for solving instances of

- extension enforcement
- status enforcement

AF Reasoning Tasks

Static computational problems

Direct inference from a given AF—no change involved

- credulous and skeptical acceptance of an argument
- extension enumeration

Many system implementations available!

Dynamic computational problems

How to change a given AF to support new information?

Pakota

First system implementation in its generality for solving instances of

- extension enforcement
- status enforcement

Extension Enforcement

Problem definition

[Coste-Marquis et al., 2015; Wallner et al., 2016]

- Input: AF $F = (A, R)$, $T \subseteq A$, semantics σ
- Task: Find an AF $F' = (A, R')$ such that
  - $T \in \sigma(F')$ (**strict** extension enforcement)
  - $T \subseteq T' \in \sigma(F')$ (**non-strict** extension enforcement)

and the number of changes $|R \Delta R'|$ is minimized.

Example

Enforcing $T = \{a\}$ strictly under the preferred semantics.

Extension Enforcement

Problem definition

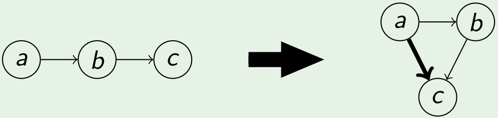
[Coste-Marquis et al., 2015; Wallner et al., 2016]

- Input: AF $F = (A, R)$, $T \subseteq A$, semantics σ
- Task: Find an AF $F' = (A, R')$ such that
  - $T \in \sigma(F')$ (**strict** extension enforcement)
  - $T \subseteq T' \in \sigma(F')$ (**non-strict** extension enforcement)

  and the number of changes $|R \Delta R'|$ is minimized.

Example

Enforcing $T = \{a\}$ strictly under the preferred semantics.



Status Enforcement

Credulous status enforcement

[Niskanen et al., 2016]

- Input: AF $F = (A, R)$, disjoint sets $P, N \subseteq A$, semantics σ
- Task: Find an AF $F' = (A, R')$ such that
  - all arguments in P are **credulously** accepted
  - all arguments in N are not **credulously** accepted

and the number of changes $|R \Delta R'|$ is minimized.

Skeptical status enforcement

[Niskanen et al., 2016]

- Input: AF $F = (A, R)$, disjoint sets $P, N \subseteq A$, semantics σ
- Task: Find an AF $F' = (A, R')$ such that
  - all arguments in P are **skeptically** accepted
  - all arguments in N are not **skeptically** accepted

and the number of changes $|R \Delta R'|$ is minimized.

Status Enforcement

Credulous status enforcement

[Niskanen et al., 2016]

- Input: AF $F = (A, R)$, disjoint sets $P, N \subseteq A$, semantics σ
- Task: Find an AF $F' = (A, R')$ such that
  - all arguments in P are **credulously** accepted
  - all arguments in N are not **credulously** accepted

and the number of changes $|R \Delta R'|$ is minimized.

Skeptical status enforcement

[Niskanen et al., 2016]

- Input: AF $F = (A, R)$, disjoint sets $P, N \subseteq A$, semantics σ
- Task: Find an AF $F' = (A, R')$ such that
  - all arguments in P are **skeptically** accepted
  - all arguments in N are not **skeptically** accepted

and the number of changes $|R \Delta R'|$ is minimized.

Computational Complexity of Enforcement

Table: Complexity of extension and status enforcement.

[Wallner et al., 2016; Niskanen et al., 2016]

	extension enf.		status enf. ($N = \emptyset$)		status enf. (unrestr. case)	
σ	strict	non-strict	credulous	skeptical	credulous	skeptical
cf	in P	in P	in P	trivial	in P	trivial
adm	**in P**	**NP-c**	**NP-c**	trivial	$\mathbf{\Sigma_2^P}$**-c**	trivial
stb	**in P**	**NP-c**	**NP-c**	$\mathbf{\Sigma_2^P}$**-c**	$\mathbf{\Sigma_2^P}$**-c**	$\mathbf{\Sigma_2^P}$**-c**
com	**NP-c**	**NP-c**	**NP-c**	NP-c	$\mathbf{\Sigma_2^P}$**-c**	NP-c
prf	$\mathbf{\Sigma_2^P}$**-c**	**NP-c**	**NP-c**	in Σ_3^P	$\mathbf{\Sigma_2^P}$**-c**	in Σ_3^P

Pakota

Features of the system

- Employs MaxSAT and SAT solvers for solving enforcement instances
- Allows for optimally solving
  - extension enforcement under $\sigma \in \{adm, com, stb, prf\}$
  - credulous status enforcement under $\sigma \in \{adm, com, stb, prf\}$
  - skeptical status enforcement under $\sigma \in \{adm, stb\}$
- Offers an interface for plugging in the MaxSAT solver of choice
- Output of MaxSAT encodings in standard WCNF and LP formats

Enforcement via Maximum Satisfiability

The (partial) maximum satisfiability problem

- **Input**: Hard clauses φ_h and soft clauses φ_s
- **Task**: Find a truth assignment that satisfies **all** hard clauses and **as many** soft clauses as possible

Used as a declarative language for solving optimization problems in NP.

NP-encodings

- Soft clauses encode **modifications** to the attack structure
- Hard clauses encode the **properties** of enforcement

Enforcement via Maximum Satisfiability

The (partial) maximum satisfiability problem

- **Input**: Hard clauses φ_h and soft clauses φ_s
- **Task**: Find a truth assignment that satisfies **all** hard clauses and **as many** soft clauses as possible

Used as a declarative language for solving optimization problems in NP.

NP-encodings

- Soft clauses encode **modifications** to the attack structure
- Hard clauses encode the **properties** of enforcement

Counterexample-Guided Abstraction Refinement

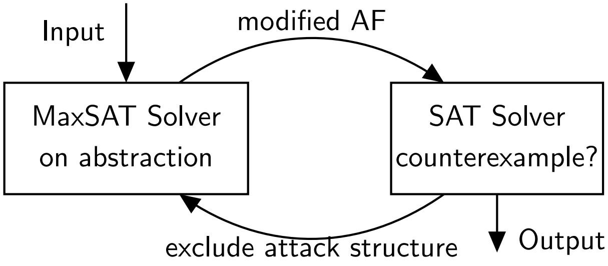
Beyond NP: Counterexample-guided abstraction refinement (CEGAR)

- **Start** with a NP-**abstraction**, solved using a MaxSAT solver
  - Lower bound on the cost of the solution
- **Refine** using a **counterexample**, provided by a SAT solver, until no counterexample is found
  - SAT check on the validity of the solution

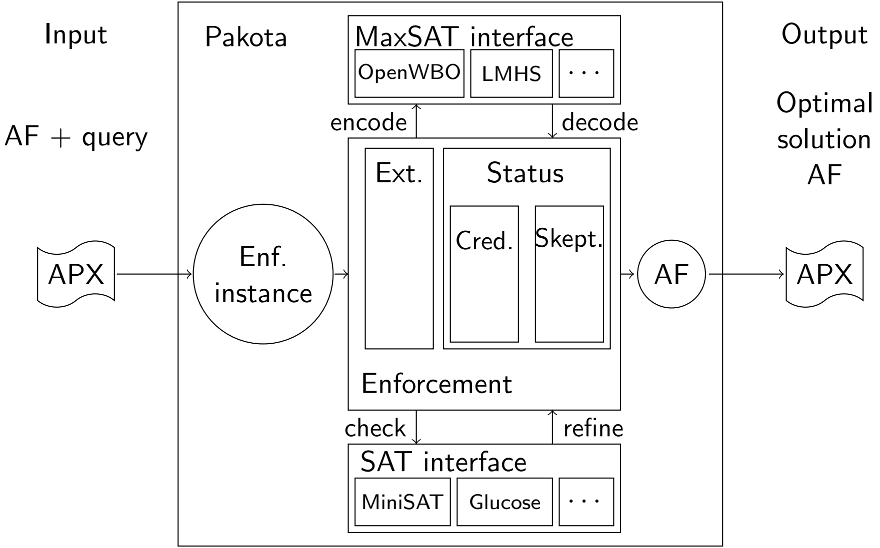
Counterexample-Guided Abstraction Refinement

Beyond NP: Counterexample-guided abstraction refinement (CEGAR)

- **Start** with a NP-**abstraction**, solved using a MaxSAT solver
  - Lower bound on the cost of the solution
- **Refine** using a **counterexample**, provided by a SAT solver, until no counterexample is found
  - SAT check on the validity of the solution



System Architecture



Performance Overview: First Level

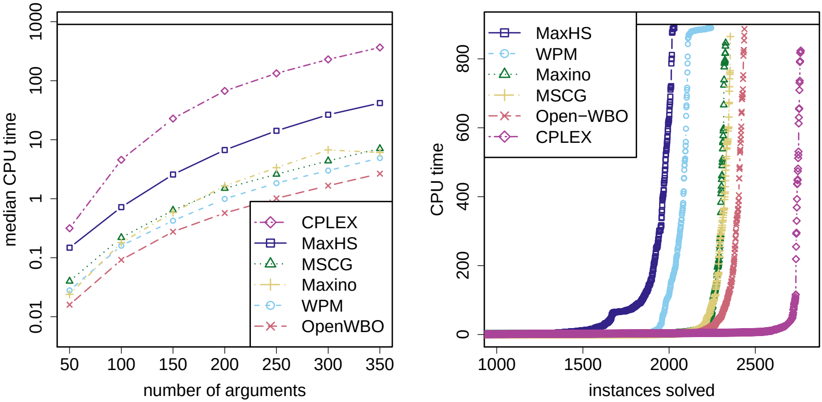


Figure: MaxSAT solver comparison on NP-complete extension enforcement; Left: strict enf. under complete; right: non-strict enf. under stable

Performance Overview: Second Level

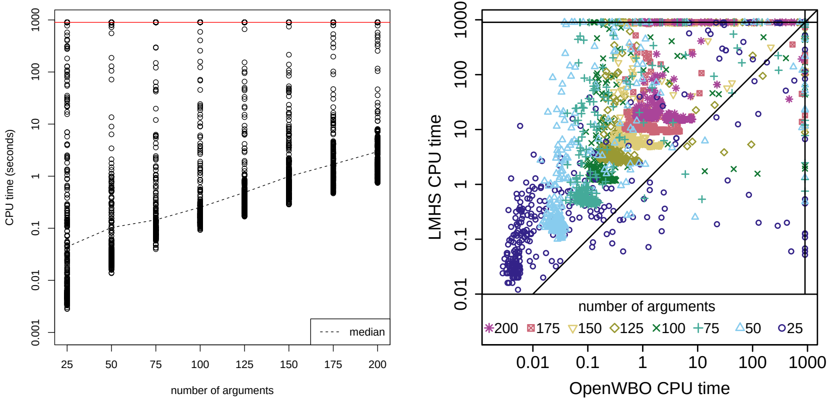


Figure: MaxSAT solver comparison on Σ_2^P-complete extension enforcement; Strict enforcement under preferred

Paper Summary

Pakota

- The first system implementation in its generality for solving problem instances of extension and status enforcement
- Utilizes MaxSAT solvers directly for the NP-complete variants and a CEGAR procedure for the problems beyond NP

Contributions

- Overview of the Pakota system:
  - System architecture and features
  - Details on encodings and algorithms
  - More in paper!
- Empirical evaluation of the impact of the choice of MaxSAT solvers
- System available online under an open source licence:

  `http://www.cs.helsinki.fi/group/coreo/pakota/`

- Future: Extending the system to support further central AF semantics

Paper Summary

Pakota

- The first system implementation in its generality for solving problem instances of extension and status enforcement
- Utilizes MaxSAT solvers directly for the NP-complete variants and a CEGAR procedure for the problems beyond NP

Contributions

- Overview of the Pakota system:
  - System architecture and features
  - Details on encodings and algorithms
  - More in paper!
- Empirical evaluation of the impact of the choice of MaxSAT solvers
- System available online under an open source licence:

  `http://www.cs.helsinki.fi/group/coreo/pakota/`
- Future: Extending the system to support further central AF semantics

References

Coste-Marquis, S., Konieczny, S., Mailly, J., and Marquis, P. (2015). Extension enforcement in abstract argumentation as an optimization problem. In *Proc. IJCAI*, pages 2876–2882. AAAI Press.

Niskanen, A., Wallner, J. P., and Järvisalo, M. (2016). Optimal status enforcement in abstract argumentation. In *Proc. IJCAI*, pages 1216–1222. IJCAI/AAAI Press.

Wallner, J. P., Niskanen, A., and Järvisalo, M. (2016). Complexity results and algorithms for extension enforcement in abstract argumentation. In *Proc. AAAI*, pages 1088–1094. AAAI Press.

Thank you for your attention!