Preprocessing Argumentation Frameworks via Replacement Patterns

Wolfgang Dvořák[1] Matti Järvisalo[2] Thomas Linsbichler[1]
Andreas Niskanen[2] Stefan Woltran[1]

[1] Institute of Logic and Computation, TU Wien, Austria
[2] HIIT, Department of Computer Science, University of Helsinki, Finland

May 9th, 2019 @ JELIA 2019, Rende, Italy

Motivation

Argumentation in Artificial Intelligence (AI)

- Active area of modern AI research
- Applications: law, medicine, eGovernment, debating technologies
- Central formalism: Dung's argumentation frameworks (AFs)

Computational Models of Argumentation

- Multiple practical AF reasoning systems (AF solvers) available
  - argument acceptance, extension enumeration
- Biennial AF solver competition: ICCMA
- Less attention on preprocessing and simplification techniques

Motivation

Argumentation in Artificial Intelligence (AI)

- Active area of modern AI research
- Applications: law, medicine, eGovernment, debating technologies
- Central formalism: Dung's argumentation frameworks (AFs)

Computational Models of Argumentation

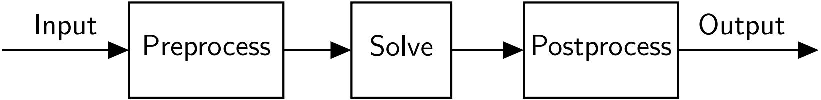
- Multiple practical AF reasoning systems (AF solvers) available
  - argument acceptance, extension enumeration
- Biennial AF solver competition: ICCMA
- Less attention on preprocessing and simplification techniques

Contributions

Solver-independent Preprocessing for AFs

- Introduce the notion of **replacement patterns**
  - polynomial-time applicable simplification rules
  - preserving a general form of equivalence
- Provide a suite of concrete replacement patterns
  - for stable, preferred, and complete semantics
- Empirically evaluate the impact of preprocessing
  - task: extension enumeration
  - especially native AF solvers affected

Preprocessing

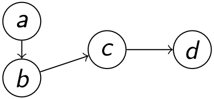


Abstract Argumentation: Syntax and Semantics

Argumentation Framework (AF)

A directed graph $F = (A, R)$, where

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



Semantics

- Functions σ mapping an AF $F = (A, R)$ to a set $\sigma(F) \subseteq 2^A$
- Define sets of jointly accepted arguments or **extensions**
  - Required to be **conflict-free** (independent sets)

Example (Stable semantics)

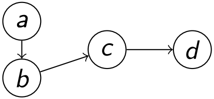
A conflict-free set $S \subseteq A$ is a **stable** extension, $S \in stb(F)$, if S attacks every argument outside S.

Abstract Argumentation: Syntax and Semantics

Argumentation Framework (AF)

A directed graph $F = (A, R)$, where

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



Semantics

- Functions σ mapping an AF $F = (A, R)$ to a set $\sigma(F) \subseteq 2^A$
- Define sets of jointly accepted arguments or **extensions**
  - Required to be **conflict-free** (independent sets)

Example (Stable semantics)

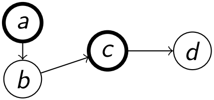
A conflict-free set $S \subseteq A$ is a **stable** extension, $S \in stb(F)$, if S attacks every argument outside S.

Abstract Argumentation: Syntax and Semantics

Argumentation Framework (AF)

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

- Functions σ mapping an AF $F = (A, R)$ to a set $\sigma(F) \subseteq 2^A$
- Define sets of jointly accepted arguments or **extensions**
  - Required to be **conflict-free** (independent sets)

Example (Stable semantics)

A conflict-free set $S \subseteq A$ is a **stable** extension, $S \in stb(F)$, if S attacks every argument outside S.

Notions of Equivalence in Abstract Argumentation

Let F and G be AFs and σ an AF semantics.

Standard equivalence

$F \equiv^\sigma G$ iff $\sigma(F) = \sigma(G)$.

Let U be a countably infinite domain of arguments, and $C \subseteq U$ a core.

C-relativized equivalence [Baumann et al. 2017]

$F \equiv^\sigma_C G$ iff for each AF H over $U \setminus C$, $F \cup H \equiv^\sigma G \cup H$.

Notions of Equivalence in Abstract Argumentation

Let F and G be AFs and σ an AF semantics.

Standard equivalence

$F \equiv^{\sigma} G$ iff $\sigma(F) = \sigma(G)$.

Let U be a countably infinite domain of arguments, and $C \subseteq U$ a core.

C-relativized equivalence [Baumann et al. 2017]

$F \equiv^{\sigma}_{C} G$ iff for each AF H over $U \setminus C$, $F \cup H \equiv^{\sigma} G \cup H$.

Merging and Unpacking Arguments

Goal: merge arguments $S \subseteq A$ resulting in an argument m_S.
Let $U_m = \{m_S \mid S \subseteq U, S \text{ is finite}\}$.

Definition

Let $F = (A, R)$ be an AF and $a, b \in A$.
The merge $\mathrm{M}(F, a, b)$ of a, b in F is the AF obtained via

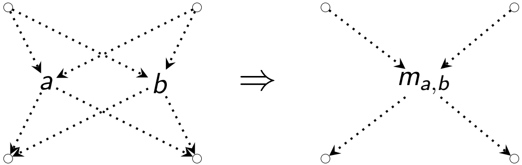
Unpacking functions $\mathrm{U}(\cdot)$ map a set of arguments over $U \cup U_m$ to the corresponding set of arguments in U.

Merging and Unpacking Arguments

Goal: merge arguments $S \subseteq A$ resulting in an argument m_S.
Let $U_m = \{m_S \mid S \subseteq U, S \text{ is finite}\}$.

Definition

Let $F = (A, R)$ be an AF and $a, b \in A$.
The merge $\mathrm{M}(F, a, b)$ of a, b in F is the AF obtained via



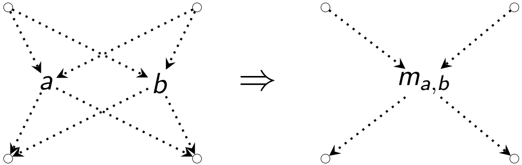
Unpacking functions $\mathrm{U}(\cdot)$ map a set of arguments over $U \cup U_m$ to the corresponding set of arguments in U.

Merging and Unpacking Arguments

Goal: merge arguments $S \subseteq A$ resulting in an argument m_S.
Let $U_m = \{m_S \mid S \subseteq U, S \text{ is finite}\}$.

Definition

Let $F = (A, R)$ be an AF and $a, b \in A$.
The merge $\mathrm{M}(F, a, b)$ of a, b in F is the AF obtained via



Unpacking functions $\mathrm{U}(\cdot)$ map a set of arguments over $U \cup U_m$ to the corresponding set of arguments in U.

Replacement Pattern

Definition

A replacement pattern P_C for a core $C \subseteq U$ is a set of pairs (F, F') of AFs F, F' such that

- $A_F \subseteq U$,
- $A_{F'} \subseteq U \cup U_m$,
- F and F' coincide on the arguments not in $C \cup \{m_S \mid S \subseteq C\}$.

Replacement Pattern

Definition

A replacement pattern P_C for a core $C \subseteq U$ is a set of pairs (F, F') of AFs F, F' such that

- $A_F \subseteq U$,
- $A_{F'} \subseteq U \cup U_m$,
- F and F' coincide on the arguments not in $C \cup \{m_S \mid S \subseteq C\}$.

Replacement Pattern

Definition

A replacement pattern P_C for a core $C \subseteq U$ is a set of pairs (F, F') of AFs F, F' such that

- $A_F \subseteq U$,
- $A_{F'} \subseteq U \cup U_m$,
- F and F' coincide on the arguments not in $C \cup \{m_S \mid S \subseteq C\}$.

Replacement Pattern

Definition

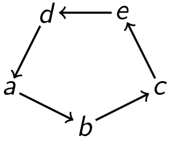
A replacement pattern P_C for a core $C \subseteq U$
is a set of pairs (F, F') of AFs F, F' such that

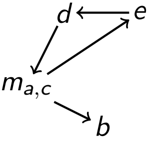
- $A_F \subseteq U$,
- $A_{F'} \subseteq U \cup U_m$,
- F and F' coincide on the arguments not in $C \cup \{m_S \mid S \subseteq C\}$.

Applying a Replacement Pattern

Example

Consider the pattern P_C with $C = \{a, b, c\}$ containing (F, F') with

$F =$ 

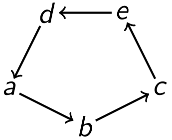
$F' =$ 

Applying a Replacement Pattern

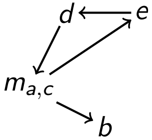
Example

Consider the pattern P_C with $C = \{a, b, c\}$ containing (F, F') with

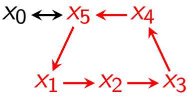
$F =$



$F' =$

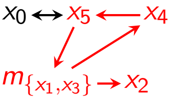


G



apply P_C $\Longrightarrow$

$P_C[G]$



Faithfulness of a Replacement Pattern

Definition

A replacement pattern P_C is σ-faithful if for all AFs G over $U \cup U_m$

$$P_C[G] \equiv^{\sigma} G.$$

Theorem

For semantics $\sigma \in \{stb, prf, com\}$ and replacement pattern P_C such that for each $(F, F') \in P_C$,

- $A_{F'} \cap S = \emptyset$ *for* $m_S \in A_{F'}$,
- $S \cap S' = \emptyset$ *for* $m_S, m_{S'} \in A_{F'}$,

we have

$$P_C \text{ is } \sigma\text{-faithful} \Leftrightarrow \text{ for each } (F, F') \in P_C, \quad F \equiv^{\sigma}_{C} \mathsf{U}(F').$$

Faithfulness of a Replacement Pattern

Definition

A replacement pattern P_C is σ-faithful if for all AFs G over $U \cup U_m$

$$P_C[G] \equiv^{\sigma} G.$$

Theorem

For semantics $\sigma \in \{stb, prf, com\}$ and replacement pattern P_C such that for each $(F, F') \in P_C$,

- $A_{F'} \cap S = \emptyset$ *for* $m_S \in A_{F'}$,
- $S \cap S' = \emptyset$ *for* $m_S, m_{S'} \in A_{F'}$,

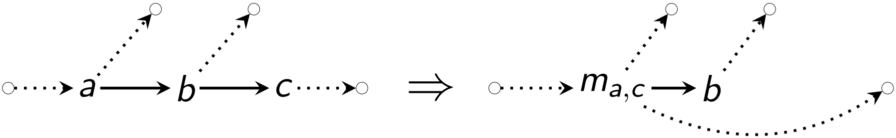
we have

$$P_C \textit{ is } \sigma\textit{-faithful} \;\Leftrightarrow\; \textit{for each } (F, F') \in P_C, \quad F \equiv^{\sigma}_{C} \mathrm{U}(F').$$

Concrete Patterns: 3-Path

Consider the directed path $a \to b \to c$.

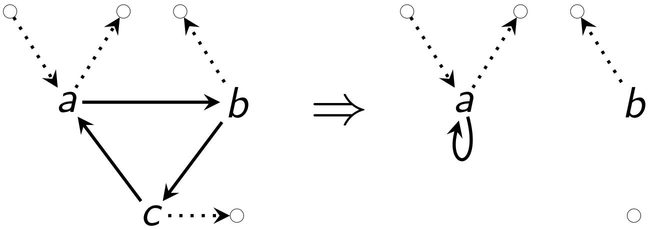
- If b and c are otherwise unattacked,
  - merge arguments a and c.



Concrete Patterns: 3-Loop

Consider the directed cycle $a \to b \to c \to a$.

- If only a is attacked from the outside,
  - remove c and the attack (a, b),
  - add a self-loop to a.



Overview of Faithfulness

Table: σ-faithfulness of replacement patterns.

	3-path	3-loop	3-cone	2to1	4-path	4-cone	3to2
stb	✓	✓	✓	✓	✓	✓	✓
prf	✓	(✓)	(✓)	✓	✓	(✓)	✓
com	✓	(✓)	×	✓	✓	×	✓

Empirical Evaluation

Experimental Setup

- Task: extension enumeration
- Semantics: stable and preferred
- Solvers: ArgTools, Heureka, CEGARTIX
- Benchmark instances: 440 AFs generated using AFBenchGen2
- Per-instance timeout: 1800 seconds

Implementation

- Encode the search of a set of arguments to which a replacement pattern is applicable using Answer Set Programming (ASP)
- Iterate through all patterns one-by-one until no such set exists
- 5 second time limit for each ASP solver call

Empirical Evaluation

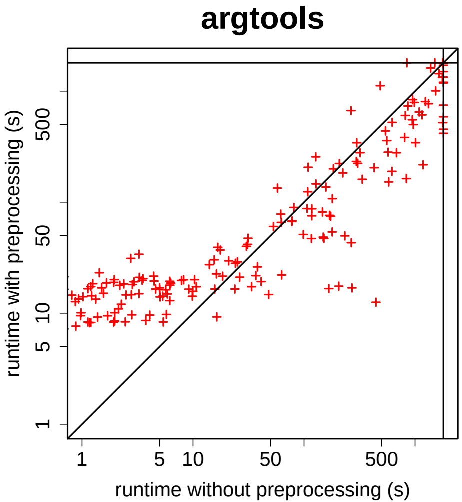
Experimental Setup

- Task: extension enumeration
- Semantics: stable and preferred
- Solvers: ArgTools, Heureka, CEGARTIX
- Benchmark instances: 440 AFs generated using AFBenchGen2
- Per-instance timeout: 1800 seconds

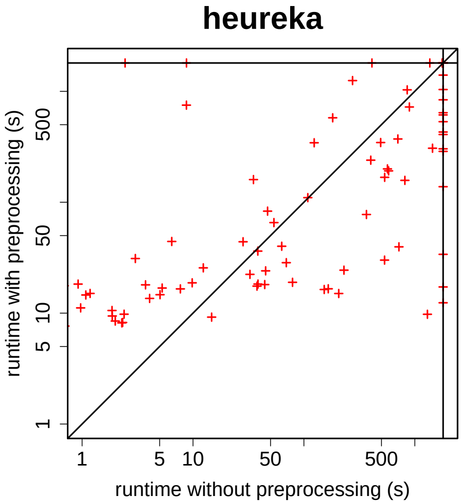
Implementation

- Encode the search of a set of arguments to which a replacement pattern is applicable using Answer Set Programming (ASP)
- Iterate through all patterns one-by-one until no such set exists
- 5 second time limit for each ASP solver call

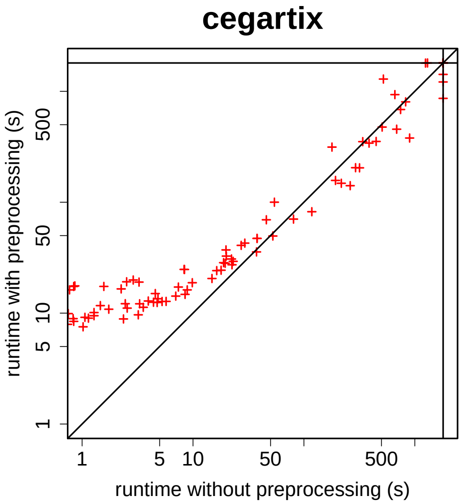
Results for Stable Semantics



Results for Stable Semantics



Results for Stable Semantics



Paper Summary

Contributions

- First steps towards solver-independent AF preprocessing
- Replacement patterns for identification of local simplifications
  - faithful w.r.t. standard AF semantics
- Suite of concrete replacement patterns
  - 3-path, 3-loop, 3-cone, 2to1, 4-path, 4-cone, 3to2
- Empirical evaluation: promising results for native AF solvers

Future Work

- Preprocessing for acceptance problems
  - faithful w.r.t. query argument
- Implementation of an optimized stand-alone AF preprocessor

Paper Summary

Contributions

- First steps towards solver-independent AF preprocessing
- Replacement patterns for identification of local simplifications
  - faithful w.r.t. standard AF semantics
- Suite of concrete replacement patterns
  - 3-path, 3-loop, 3-cone, 2to1, 4-path, 4-cone, 3to2
- Empirical evaluation: promising results for native AF solvers

Future Work

- Preprocessing for acceptance problems
  - faithful w.r.t. query argument
- Implementation of an optimized stand-alone AF preprocessor