Synthesizing Argumentation Frameworks from Examples

Andreas Niskanen   Johannes P. Wallner   Matti Järvisalo

HIIT, Department of Computer Science
University of Helsinki
Finland

August 31, 2016 @ ECAI 2016, The Hague

Motivation

Argumentation

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

Dung's argumentation frameworks (AFs)

- Central KR formalism in **abstract argumentation**
- Interested in jointly acceptable sets of arguments
  - Defined via **semantics**, e.g. conflict-free, admissible, stable

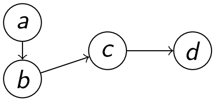
Motivation

Argumentation

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

Dung's argumentation frameworks (AFs)

- Central KR formalism in **abstract argumentation**
- Interested in jointly acceptable sets of arguments
  - Defined via **semantics**, e.g. conflict-free, admissible, stable



Contributions

AF Synthesis

- **Introduce the problem** as a natural generalization of realizability
  - Allowing incomplete information and noisy settings
- **Complexity analysis** for multiple AF semantics
  - NP-complete in the general case
- **Algorithms** based on constraint optimization
  - Encoding the problem as an instance of MaxSAT
- **Implementation** and empirical evaluation

Argumentation Frameworks

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

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

- a function σ which maps an AF $F = (A, R)$ to a set $\sigma(F) \subseteq 2^A$

e.g. **conflict-free:** $E \in cf(F)$ if E is an independent set

admissible: $E \in adm(F)$ if $E \in cf(F)$ and every $a \in E$ is defended

Argumentation Frameworks

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

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

- a function σ which maps an AF $F = (A, R)$ to a set $\sigma(F) \subseteq 2^A$

e.g. **conflict-free:** $E \in cf(F)$ if E is an independent set

admissible: $E \in adm(F)$ if $E \in cf(F)$ and every $a \in E$ is defended

AF Reasoning Tasks

Reasoning about extensions from AFs

- Given: AF F, semantics σ
- Tasks include:
  - credulous and skeptical acceptance of an argument
  - extension enumeration

The inverse problem: Realizability

- Given: Collection $\mathcal{E}$ of extensions, semantics σ
- Task: is there an AF F with $\sigma(F) = \mathcal{E}$?

Note: $\sigma(F) = \mathcal{E}$ implies **all** other subsets are **not** extensions!

AF Reasoning Tasks

Reasoning about extensions from AFs

- Given: AF F, semantics σ
- Tasks include:
  - credulous and skeptical acceptance of an argument
  - extension enumeration

The inverse problem: Realizability

- Given: Collection $\mathcal{E}$ of extensions, semantics σ
- Task: is there an AF F with $\sigma(F) = \mathcal{E}$?

Note: $\sigma(F) = \mathcal{E}$ implies **all** other subsets are **not** extensions!

AF Reasoning Tasks

Reasoning about extensions from AFs

- Given: AF F, semantics σ
- Tasks include:
  - credulous and skeptical acceptance of an argument
  - extension enumeration

The inverse problem: Realizability

- Given: Collection $\mathcal{E}$ of extensions, semantics σ
- Task: is there an AF F with $\sigma(F) = \mathcal{E}$?

Note: $\sigma(F) = \mathcal{E}$ implies **all** other subsets are **not** extensions!

AF Synthesis

Problem instance

(A, E^+, E^-, σ), where

- A is a non-empty set of **arguments**
- E^+, E^- are sets of **positive** and **negative examples**
  - pairs (S, w), where $S \subseteq 2^A$ and $w > 0$ is the example's **weight**
- σ is an AF **semantics**

positive E^+	negative E^-
$e_1 = (\{a, b\}, 1)$	$e_4 = (\{a\}, 1)$
$e_2 = (\{a, c\}, 1)$	$e_5 = (\{a, b, c\}, 5)$
$e_3 = (\{b, c\}, 5)$	

AF Synthesis

Problem instance

(A, E^+, E^-, σ), where

- A is a non-empty set of **arguments**
- E^+, E^- are sets of **positive** and **negative examples**
  - pairs (S, w), where $S \subseteq 2^A$ and $w > 0$ is the example's **weight**
- σ is an AF **semantics**

positive E^+	negative E^-
$e_1 = (\{a, b\}, 1)$	$e_4 = (\{a\}, 1)$
$e_2 = (\{a, c\}, 1)$	$e_5 = (\{a, b, c\}, 5)$
$e_3 = (\{b, c\}, 5)$	

AF Synthesis

AF F satisfies

- a positive example $e = (S, w)$ if $S \in \sigma(F)$
- a negative example $e = (S, w)$ if $S \notin \sigma(F)$

Cost of F: the sum of the weights of examples not satisfied by F

The AF synthesis problem

- **Input**: (A, E^+, E^-, σ)
- **Task**: Find the closest AF in terms of minimizing the cost

AF Synthesis

AF F satisfies

- a positive example $e = (S, w)$ if $S \in \sigma(F)$
- a negative example $e = (S, w)$ if $S \notin \sigma(F)$

Cost of F: the sum of the weights of examples not satisfied by F

The AF synthesis problem

- **Input**: (A, E^+, E^-, σ)
- **Task**: Find the closest AF in terms of minimizing the cost

AF Synthesis

AF F satisfies

- a positive example $e = (S, w)$ if $S \in \sigma(F)$
- a negative example $e = (S, w)$ if $S \notin \sigma(F)$

Cost of F: the sum of the weights of examples not satisfied by F

The AF synthesis problem

- **Input**: (A, E^+, E^-, σ)
- **Task**: Find the closest AF in terms of minimizing the cost

AF Synthesis

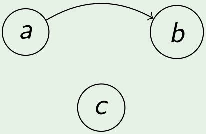
Example

positive E^+	negative E^-
$e_1 = (\{a, b\}, 1)$	$e_4 = (\{a\}, 1)$
$e_2 = (\{a, c\}, 1)$	$e_5 = (\{a, b, c\}, 5)$
$e_3 = (\{b, c\}, 5)$	

AF Synthesis

Example

positive E^+	negative E^-
$e_1 = (\{a, b\}, 1)$	$e_4 = (\{a\}, 1)$
$e_2 = (\{a, c\}, 1)$	$e_5 = (\{a, b, c\}, 5)$
$e_3 = (\{b, c\}, 5)$	



$\sigma = cf$

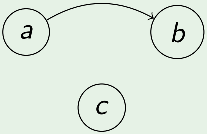
$\sigma = adm$

$\sigma = stb$

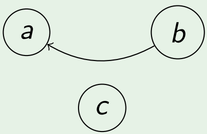
AF Synthesis

Example

positive E^+	negative E^-
$e_1 = (\{a, b\}, 1)$	$e_4 = (\{a\}, 1)$
$e_2 = (\{a, c\}, 1)$	$e_5 = (\{a, b, c\}, 5)$
$e_3 = (\{b, c\}, 5)$	



$\sigma = cf$



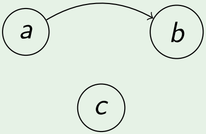
$\sigma = adm$

$\sigma = stb$

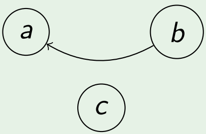
AF Synthesis

Example

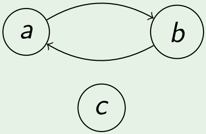
positive E^+	negative E^-
$e_1 = (\{a, b\}, 1)$	$e_4 = (\{a\}, 1)$
$e_2 = (\{a, c\}, 1)$	$e_5 = (\{a, b, c\}, 5)$
$e_3 = (\{b, c\}, 5)$	



$\sigma = cf$



$\sigma = adm$



$\sigma = stb$

Computational Complexity

Complexity of AF synthesis in different settings

	no restrictions	$E^+ = \emptyset$	$E^- = \emptyset$
Conflict-free	NP-complete	trivial	trivial
Admissible	NP-complete	trivial	trivial
Stable	NP-complete	trivial	NP-complete

- Complete digraph satisfies all negative examples $\Rightarrow E^+ = \emptyset$ trivial
- Empty digraph satisfies all positive examples under $\sigma \in \{cf, adm\}$ $\Rightarrow E^- = \emptyset$ trivial
- NP-hardness by a reduction from the Boolean satisfiability problem
- Note: Under stable semantics even $E^- = \emptyset$ is NP-complete!

Computational Complexity

Complexity of AF synthesis in different settings

	no restrictions	$E^+ = \emptyset$	$E^- = \emptyset$
Conflict-free	NP-complete	trivial	trivial
Admissible	NP-complete	trivial	trivial
Stable	NP-complete	trivial	NP-complete

- Complete digraph satisfies all negative examples $\Rightarrow E^+ = \emptyset$ trivial
- Empty digraph satisfies all positive examples under $\sigma \in \{cf, adm\}$ $\Rightarrow E^- = \emptyset$ trivial
- NP-hardness by a reduction from the Boolean satisfiability problem
- Note: Under stable semantics even $E^- = \emptyset$ is NP-complete!

Computational Complexity

Complexity of AF synthesis in different settings

	no restrictions	$E^+ = \emptyset$	$E^- = \emptyset$
Conflict-free	NP-complete	**trivial**	trivial
Admissible	NP-complete	**trivial**	trivial
Stable	NP-complete	**trivial**	NP-complete

- Complete digraph satisfies all negative examples
  $\Rightarrow E^+ = \emptyset$ trivial
- Empty digraph satisfies all positive examples under $\sigma \in \{cf, adm\}$
  $\Rightarrow E^- = \emptyset$ trivial
- NP-hardness by a reduction from the Boolean satisfiability problem
- Note: Under stable semantics even $E^- = \emptyset$ is NP-complete!

Computational Complexity

Complexity of AF synthesis in different settings

	no restrictions	$E^+ = \emptyset$	$E^- = \emptyset$
Conflict-free	NP-complete	trivial	**trivial**
Admissible	NP-complete	trivial	**trivial**
Stable	NP-complete	trivial	NP-complete

- Complete digraph satisfies all negative examples
  $\Rightarrow E^+ = \emptyset$ trivial
- Empty digraph satisfies all positive examples under $\sigma \in \{cf, adm\}$
  $\Rightarrow E^- = \emptyset$ trivial
- NP-hardness by a reduction from the Boolean satisfiability problem
- Note: Under stable semantics even $E^- = \emptyset$ is NP-complete!

Computational Complexity

Complexity of AF synthesis in different settings

	no restrictions	$E^+ = \emptyset$	$E^- = \emptyset$
Conflict-free	**NP-complete**	trivial	trivial
Admissible	**NP-complete**	trivial	trivial
Stable	**NP-complete**	trivial	**NP-complete**

- Complete digraph satisfies all negative examples
  $\Rightarrow E^+ = \emptyset$ trivial
- Empty digraph satisfies all positive examples under $\sigma \in \{cf, adm\}$
  $\Rightarrow E^- = \emptyset$ trivial
- NP-hardness by a reduction from the Boolean satisfiability problem
- Note: Under stable semantics even $E^- = \emptyset$ is NP-complete!

Computational Complexity

Complexity of AF synthesis in different settings

	no restrictions	$E^+ = \emptyset$	$E^- = \emptyset$
Conflict-free	NP-complete	trivial	trivial
Admissible	NP-complete	trivial	trivial
Stable	NP-complete	trivial	**NP-complete**

- Complete digraph satisfies all negative examples
  $\Rightarrow E^+ = \emptyset$ trivial
- Empty digraph satisfies all positive examples under $\sigma \in \{cf, adm\}$
  $\Rightarrow E^- = \emptyset$ trivial
- NP-hardness by a reduction from the Boolean satisfiability problem
- Note: Under stable semantics even $E^- = \emptyset$ is NP-complete!

Maximum Satisfiability (MaxSAT)

Problem instance

A (weighted partial) MaxSAT instance consists of

- hard clauses φ_h and soft clauses φ_s
- weight $w(C) > 0$ for each soft clause $C \in \varphi_s$

Cost of a truth assignment: sum of weights of the soft clauses not satisfied

The maximum satisfiability problem

- **Input**: $(\varphi_h, \varphi_s, w)$
- **Task**: Find a truth assignment that minimizes the cost

Maximum Satisfiability (MaxSAT)

Problem instance

A (weighted partial) MaxSAT instance consists of

- hard clauses φ_h and soft clauses φ_s
- weight $w(C) > 0$ for each soft clause $C \in \varphi_s$

Cost of a truth assignment: sum of weights of the soft clauses not satisfied

The maximum satisfiability problem

- **Input**: $(\varphi_h, \varphi_s, w)$
- **Task**: Find a truth assignment that minimizes the cost

MaxSAT Encoding of AF Synthesis

Let (A, E^+, E^-, σ) be an instance of AF synthesis.

Variables

- $r_{a,b}$ = "attack (a, b) included" for all $a, b \in A$
- Ext^e_σ = "e is a σ-extension" for all $e \in E^+ \cup E^-$

Clauses

- **Problem structure:** For each $e \in E^+ \cup E^-$ a hard clause $\text{Ext}^e_\sigma \leftrightarrow \varphi_\sigma(e)$ where $\varphi_\sigma(e)$ encodes that e is a σ-extension
- **Objective function:** For each $e \in E^+$ a soft clause Ext^e_σ, for each $e \in E^-$ a soft clause $\neg\text{Ext}^e_\sigma$
  - Weights of soft clauses according to weights of examples

MaxSAT Encoding of AF Synthesis

Let (A, E^+, E^-, σ) be an instance of AF synthesis.

Variables

- $r_{a,b}$ = "attack (a, b) included" — for all $a, b \in A$
- Ext_σ^e = "e is a σ-extension" — for all $e \in E^+ \cup E^-$

Clauses

- **Problem structure:** For each $e \in E^+ \cup E^-$ a hard clause $\text{Ext}_\sigma^e \leftrightarrow \varphi_\sigma(e)$ where $\varphi_\sigma(e)$ encodes that e is a σ-extension
- **Objective function:** For each $e \in E^+$ a soft clause Ext_σ^e, for each $e \in E^-$ a soft clause $\neg\text{Ext}_\sigma^e$
  - Weights of soft clauses according to weights of examples

MaxSAT Encoding of AF Synthesis

Let (A, E^+, E^-, σ) be an instance of AF synthesis.

Variables

- $r_{a,b}$ = "attack (a, b) included" for all $a, b \in A$
- Ext_σ^e = "e is a σ-extension" for all $e \in E^+ \cup E^-$

Clauses

- **Problem structure:** For each $e \in E^+ \cup E^-$ a hard clause $\mathrm{Ext}_\sigma^e \leftrightarrow \varphi_\sigma(e)$ where $\varphi_\sigma(e)$ encodes that e is a σ-extension
- **Objective function:** For each $e \in E^+$ a soft clause Ext_σ^e, for each $e \in E^-$ a soft clause $\neg\mathrm{Ext}_\sigma^e$
  - Weights of soft clauses according to weights of examples

$\mathrm{Ext}_\sigma^e \leftrightarrow \varphi_\sigma(e)$

Example

Conflict-free sets:

$$\varphi_{cf}(e) = \bigwedge_{a,b \in e} \neg r_{a,b}$$

Admissible sets:

$$\varphi_{adm}(e) = \varphi_{cf}(e) \wedge \bigwedge_{a \in e} \bigwedge_{b \in A \setminus e} \left(r_{b,a} \rightarrow \bigvee_{c \in e} r_{c,b} \right)$$

Implementation and Benchmarks

AFSynth: System for AF synthesis

- MaxSAT encodings for conflict-free, admissible and stable semantics
- Available at

  http://www.cs.helsinki.fi/group/coreo/afsynth/

Benchmark setup

AF synthesis instances generated from

- ICCMA'15 competition AFs (see paper!)
- a random model with 100 arguments
  - parameters: number of positive and number of negative examples

Experiment setup

- Timeout limit: 900 seconds
- MaxSAT solver: MSCG

Implementation and Benchmarks

AFSynth: System for AF synthesis

- MaxSAT encodings for conflict-free, admissible and stable semantics
- Available at

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

Benchmark setup

AF synthesis instances generated from

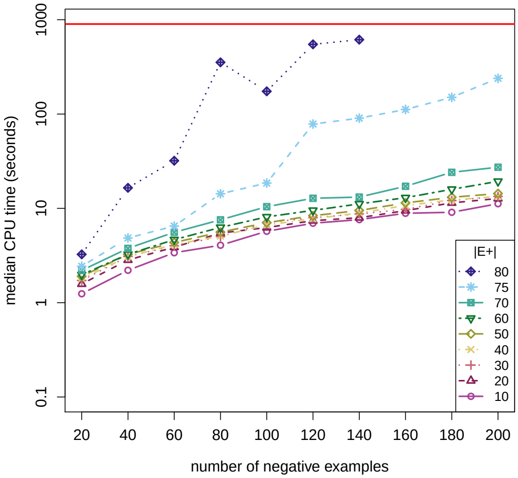
- ICCMA'15 competition AFs (see paper!)
- a random model with 100 arguments
  - parameters: number of positive and number of negative examples

Experiment setup

- Timeout limit: 900 seconds
- MaxSAT solver: MSCG

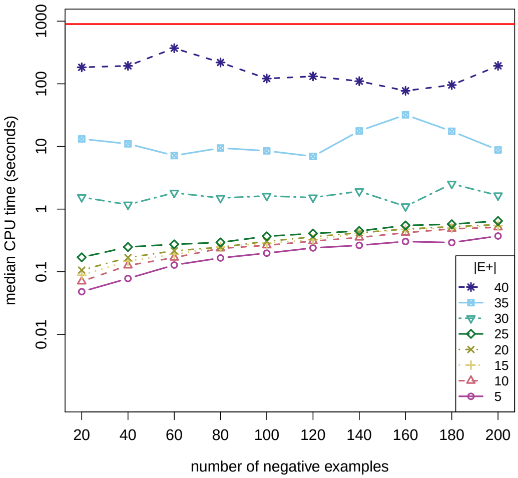
Empirical Evaluation

Random instances under admissible:



Empirical Evaluation

Random instances under stable:



Paper Summary

AF Synthesis

- Generalization of realizability in abstract argumentation
  - Incomplete information—only partial knowledge of (non-)extensions as positive and negative examples
  - Noisy settings—relative trust in the examples expressed via weights

Contributions

- Complexity analysis for three key AF semantics and different settings
- Solution algorithm based on Boolean optimization
- Empirical evaluation of the system AFSynth, available at

  `http://www.cs.helsinki.fi/group/coreo/afsynth/`
- More in paper: analysis of relationship to realizability, problem variants (e.g. symbolic representation of examples)
- Future work: extend approach to cover further AF semantics

Paper Summary

AF Synthesis

- Generalization of realizability in abstract argumentation
  - Incomplete information—only partial knowledge of (non-)extensions as positive and negative examples
  - Noisy settings—relative trust in the examples expressed via weights

Contributions

- Complexity analysis for three key AF semantics and different settings
- Solution algorithm based on Boolean optimization
- Empirical evaluation of the system AFSynth, available at

  `http://www.cs.helsinki.fi/group/coreo/afsynth/`
- More in paper: analysis of relationship to realizability, problem variants (e.g. symbolic representation of examples)
- Future work: extend approach to cover further AF semantics