On the Complexity of Computing the Justification Status of an Argument◇

TAFA 2011, Barcelona

Wolfgang Dvořák

Institute of Information Systems,
Vienna University of Technology

July 17, 2011

◇ Supported by the Vienna Science and Technology Fund (WWTF) under grant ICT08-028

Motivation

We adress the problem of:

Determining the acceptance status of an argument in abstract argumentation (Given a semantics for computing the extensions).

Motivation

We adress the problem of:

Determining the acceptance status of an argument in abstract argumentation (Given a semantics for computing the extensions).

- Traditional: Skeptical and/or Credulous Acceptance.
- Wu and Caminada recently proposed a new approach: The Justification Status of an Argument.

Motivation

We adress the problem of:

Determining the acceptance status of an argument in abstract argumentation (Given a semantics for computing the extensions).

- Traditional: Skeptical and/or Credulous Acceptance.
- Wu and Caminada recently proposed a new approach:
  The Justification Status of an Argument.
- Their original approach is stated in terms of complete semantics.
  $\hookrightarrow$ We generalize it to arbitrary semantics
- Computational issues where neglected.
  $\hookrightarrow$ We provide an comprehensive complexity analysis.

Outline

Argumentation Labelings

Let $F = (A, R)$ be an AF.

Definition

A labeling for F is a function $\mathcal{L} : A \rightarrow \{in, out, undec\}$. We denote labelings by triples $(\mathcal{L}_{in}, \mathcal{L}_{out}, \mathcal{L}_{undec})$, with $\mathcal{L}_l = \{a \in A \mid \mathcal{L}(a) = l\}$.

Argumentation Labelings

Let $F = (A, R)$ be an AF.

Definition

A labeling for F is a function $\mathcal{L} : A \rightarrow \{in, out, undec\}$. We denote labelings by triples $(\mathcal{L}_{in}, \mathcal{L}_{out}, \mathcal{L}_{undec})$, with $\mathcal{L}_l = \{a \in A \mid \mathcal{L}(a) = l\}$.

The range of a set $S \subseteq A$ is defined as $S_R^+ = S \cup \{b \mid \exists a \in S : (a, b) \in R\}$.
We define the induced labeling $\text{Ext2Lab}_F(E)$ of an extension $E \subseteq A$:

$$\text{Ext2Lab}_F(E) = (E, E_R^+ \setminus E, A \setminus E_R^+)$$

Argumentation Labelings

Let $F = (A, R)$ be an AF.

Definition

A labeling for F is a function $\mathcal{L} : A \rightarrow \{in, out, undec\}$. We denote labelings by triples $(\mathcal{L}_{in}, \mathcal{L}_{out}, \mathcal{L}_{undec})$, with $\mathcal{L}_l = \{a \in A \mid \mathcal{L}(a) = l\}$.

The range of a set $S \subseteq A$ is defined as $S_R^+ = S \cup \{b \mid \exists a \in S : (a, b) \in R\}$.
We define the induced labeling $\text{Ext2Lab}_F(E)$ of an extension $E \subseteq A$:

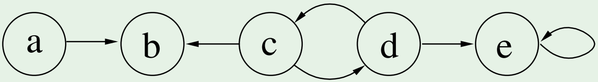
$$\text{Ext2Lab}_F(E) = (E, E_R^+ \setminus E, A \setminus E_R^+)$$

Definition

Let σ be an extension-based semantics. The corresponding labeling-based semantics $\sigma_{\mathcal{L}}$ is defined as $\sigma_{\mathcal{L}}(F) = \{\text{Ext2Lab}(E) \mid E \in \sigma(F)\}$.

Argumentation Labelings - Example

Example



$comp(F) = \{\{a\}, \{a, c\}, \{a, d\}\}$

The *complete labelings* are:

- $(\{a\}, \{b\}, \{c, d, e\})$,
- $(\{a, c\}, \{b, d\}, \{e\})$,
- $(\{a, d\}, \{b, c, e\}, \{\})$

Justification Status of an Argument

Definition

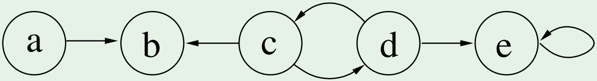
Let $F = (A, R)$ be an AF and σ a semantic. The justification status of an $a \in A$ wrt σ is defined as $\mathcal{JS}_\sigma(F, a) = \{\mathcal{L}(a) \mid \mathcal{L} \in \sigma_\mathcal{L}(F)\}$.

Justification Status of an Argument

Definition

Let $F = (A, R)$ be an AF and σ a semantic. The justification status of an $a \in A$ wrt σ is defined as $\mathcal{JS}_\sigma(F, a) = \{\mathcal{L}(a) \mid \mathcal{L} \in \sigma_\mathcal{L}(F)\}$.

Example



$comp(F) = \{\{a\}, \{a, c\}, \{a, d\}\}$

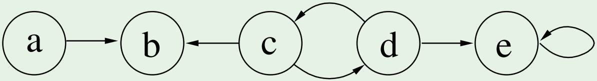
$\mathcal{JS}_{comp}(F, a) = \{in\}$,

Justification Status of an Argument

Definition

Let $F = (A, R)$ be an AF and σ a semantic. The justification status of an $a \in A$ wrt σ is defined as $\mathcal{JS}_\sigma(F, a) = \{\mathcal{L}(a) \mid \mathcal{L} \in \sigma_\mathcal{L}(F)\}$.

Example



$comp(F) = \{\{a\}, \{a, c\}, \{a, d\}\}$

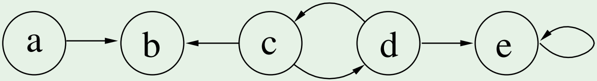
$\mathcal{JS}_{comp}(F, a) = \{in\}$, $\mathcal{JS}_{comp}(F, b) = \{out\}$,

Justification Status of an Argument

Definition

Let $F = (A, R)$ be an AF and σ a semantic. The justification status of an $a \in A$ wrt σ is defined as $\mathcal{JS}_\sigma(F, a) = \{\mathcal{L}(a) \mid \mathcal{L} \in \sigma_{\mathcal{L}}(F)\}$.

Example



$comp(F) = \{\{a\}, \{a, c\}, \{a, d\}\}$

$\mathcal{JS}_{comp}(F, a) = \{in\}$, $\mathcal{JS}_{comp}(F, b) = \{out\}$,

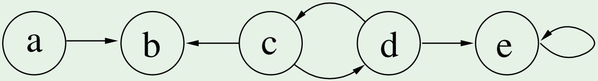
$\mathcal{JS}_{comp}(F, c) = \mathcal{JS}_{comp}(F, d) = \{in, out, undec\}$

Justification Status of an Argument

Definition

Let $F = (A, R)$ be an AF and σ a semantic. The justification status of an $a \in A$ wrt σ is defined as $\mathcal{JS}_\sigma(F, a) = \{\mathcal{L}(a) \mid \mathcal{L} \in \sigma_\mathcal{L}(F)\}$.

Example



$comp(F) = \{\{a\}, \{a, c\}, \{a, d\}\}$

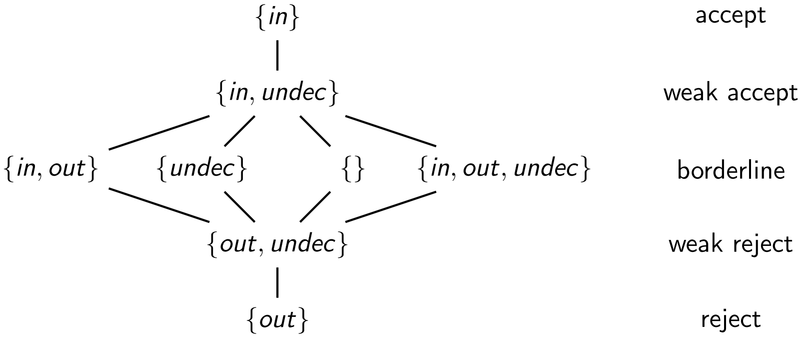
$\mathcal{JS}_{comp}(F, a) = \{in\}$, $\mathcal{JS}_{comp}(F, b) = \{out\}$,

$\mathcal{JS}_{comp}(F, c) = \mathcal{JS}_{comp}(F, d) = \{in, out, undec\}$

$\mathcal{JS}_{comp}(F, e) = \{out, undec\}$

Possible Justification Statuses

Each element of $2^{\{in, out, undec\}}$ is a justification status:



Possible Justification Statuses

Not all justification statuses are possible under each semantics:

Possible Justification Statuses

Not all justification statuses are possible under each semantics:

Theorem

Let $F = (A, R)$ be an AF and $a \in A$. Then we have that:

- $\mathcal{JS}_{ground}(F, a) \in \{\{in\}, \{out\}, \{undec\}\}$
- $\mathcal{JS}_{adm}(F, a) \in \{\{undec\}, \{in, undec\}, \{out, undec\}, \{in, out, undec\}\}$
- $\mathcal{JS}_{comp}(F, a) \in 2^{\{in,out,undec\}} \setminus \{\emptyset, \{in, out\}\}$
- $\mathcal{JS}_{stable}(F, a) \in \{\{in\}, \{out\}, \{in, out\}, \{\}\}$
- $\mathcal{JS}_{pref}(F, a) \in 2^{\{in,out,undec\}} \setminus \{\emptyset\}$
- $\mathcal{JS}_{semi}(F, a) \in 2^{\{in,out,undec\}} \setminus \{\emptyset\}$
- $\mathcal{JS}_{stage}(F, a) \in 2^{\{in,out,undec\}} \setminus \{\emptyset\}$

Computational Complexity - Problems of interest

We are interested in the following two problems:

- The justification status decision problem JS_σ
  Given: AF $F = (A, R)$, $L \subseteq \{in, out, undec\}$ and argument $a \in A$.
  Question: Does $\mathcal{JS}_\sigma(F, a) = L$ hold?
- The generalized justification status decision problem GJS_σ
  Given: AF $F = (A, R)$, $L, M \subseteq \{in, out, undec\}$ and argument $a \in A$.
  Question: Does $L \subseteq \mathcal{JS}_\sigma(F, a)$ and $\mathcal{JS}_\sigma(F, a) \cap M = \emptyset$ hold?.

Computational Complexity - Problems of interest

We are interested in the following two problems:

- The justification status decision problem JS_σ
  Given: AF $F = (A, R)$, $L \subseteq \{in, out, undec\}$ and argument $a \in A$.
  Question: Does $\mathcal{JS}_\sigma(F, a) = L$ hold?
- The generalized justification status decision problem GJS_σ
  Given: AF $F = (A, R)$, $L, M \subseteq \{in, out, undec\}$ and argument $a \in A$.
  Question: Does $L \subseteq \mathcal{JS}_\sigma(F, a)$ and $\mathcal{JS}_\sigma(F, a) \cap M = \emptyset$ hold?.

Clearly the first problem can be encoded as instance of the second one.

Computational Complexity - Problems of interest

We are interested in the following two problems:

- The justification status decision problem JS_σ
  Given: AF $F = (A, R)$, $L \subseteq \{in, out, undec\}$ and argument $a \in A$.
  Question: Does $\mathcal{JS}_\sigma(F, a) = L$ hold?
- The generalized justification status decision problem GJS_σ
  Given: AF $F = (A, R)$, $L, M \subseteq \{in, out, undec\}$ and argument $a \in A$.
  Question: Does $L \subseteq \mathcal{JS}_\sigma(F, a)$ and $\mathcal{JS}_\sigma(F, a) \cap M = \emptyset$ hold?.

Clearly the first problem can be encoded as instance of the second one.

To obtain completness for both problems we show

- membership for GJS_σ and
- hardness for JS_σ

Computational Complexity - Membership

Theorem

If the problem of verifying a σ-extension is in the complexity class $\mathcal{C}$ then the problem GJS_σ is in the complexity class $\mathsf{NP}^{\mathcal{C}} \wedge \text{co-}\mathsf{NP}^{\mathcal{C}}$.

Computational Complexity - Membership

Theorem

If the problem of verifying a σ-extension is in the complexity class $\mathcal{C}$ then the problem GJS_σ is in the complexity class $\mathrm{NP}^{\mathcal{C}} \wedge \text{co-NP}^{\mathcal{C}}$.

Proof Ideas.

We provide a $\mathrm{NP}^{\mathcal{C}}$ algorithm to decide $L \subseteq \mathcal{JS}_\sigma(F, a)$

- For each $l \in L$ guess a labeling $\mathcal{L}_l$ with $\mathcal{L}_l(a) = l$
- Test whether $\mathcal{L}_l \in \sigma(F)$ or not, using the $\mathcal{C}$-oracle.
- Accept if for each $l \in L$, $\mathcal{L}_l \in \sigma(F)$

and a $\text{co-NP}^{\mathcal{C}}$ algorithm to decide $\mathcal{JS}_\sigma(F, a) \cap M = \emptyset$,

- For each $l \in M$ guess a labeling $\mathcal{L}_l$ with $\mathcal{L}_l(a) = l$
- Test whether $\mathcal{L}_l \in \sigma(F)$ or not
- Accept if there exists an $l \in M$ such that $\mathcal{L}_l \in \sigma(F)$

□

Computational Complexity - Hardness

Theorem

The problems JS_{comp}, GJS_{comp}, JS_{adm}, GJS_{adm} *are* DP-*hard, i.e.* NP $\wedge$ co-NP-*hard.*

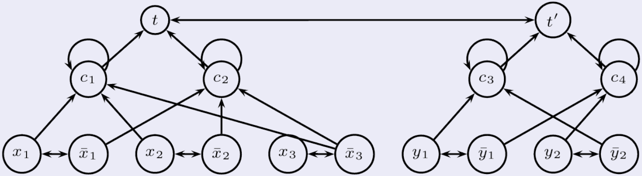
Computational Complexity - Hardness

Theorem

The problems JS_{comp}, GJS_{comp}, JS_{adm}, GJS_{adm} are DP*-hard, i.e.* NP $\wedge$ co-NP*-hard.*

Proof Idea.

We prove hardness by reducing the (DP-hard) SAT-UNSAT problem to JS_{comp} (resp. JS_{adm}).



The reduction builds on slightly modified standard translations of both formulas and adds a mutual attack between them. □

Computational Complexity

σ	*ground*	*adm*	*comp*	*stable*	*pref*	*semi*	*stage*
Cred_σ	P-c	NP-c	NP-c	NP-c	NP-c	Σ_2^p-c	Σ_2^p-c
Skept_σ	P-c	trivial	P-c	co-NP/DP-c	Π_2^p-c	Π_2^p-c	Π_2^p-c
JS_σ	P-c	DP-c	DP-c	DP-c	$\mathrm{P}^{\Sigma_2^p[1]}$-c	DP_2-c	DP_2-c
GJS_σ	P-c	DP-c	DP-c	DP-c	$\mathrm{P}^{\Sigma_2^p[1]}$-c	DP_2-c	DP_2-c

Table: Complexity Results ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$)

Relations between the above complexity classes:

$$\mathrm{P} \subseteq \begin{matrix} \mathrm{NP} \\ \text{co-NP} \end{matrix} \subseteq \mathrm{DP} \subseteq \begin{matrix} \Sigma_2^p \\ \Pi_2^p \end{matrix} \subseteq \mathrm{P}^{\Sigma_2^p[1]} \subseteq \mathrm{DP}_2$$

Conclusion

- We generalised the concept of the justification status of an argument to arbitrary semantics.
- Using the Justification Status in general increases the complexity.
- Two sources of complexity:

  We have to provide wittness for

  - some labels to be in the justification status
  - some labels not to be in the justification status
- There are several problem classes where these decision problems are easier, e.g. Skeptical Acceptance.