ABSTRACT ARGUMENTATION

Generalizations of Argumentation Frameworks

* slides adapted from Stefan Woltran’s lecture on Abstract Argumentation

Sarah Gaggl

Dresden, ICCL Summer School 2017
Outline

1. Complexity of Abstract Argumentation
2. Extending Dung's Framework
3. Abstract Dialectical Frameworks
### Credulous Acceptance

\[ \text{Cred}_\sigma : \text{Given AF } F = (A, R) \text{ and } a \in A; \text{ is } a \text{ contained in at least one } \sigma \text{-extension of } F? \]

### Skeptical Acceptance

\[ \text{Skept}_\sigma : \text{Given AF } F = (A, R) \text{ and } a \in A; \text{ is } a \text{ contained in every } \sigma \text{-extension of } F? \]

If no extension exists then all arguments are skeptically accepted and no argument is credulously accepted\(^1\).

---

\(^1\) This is only relevant for stable semantics.
Credulous Acceptance

Cred$_\sigma$: Given AF $F = (A, R)$ and $a \in A$; is $a$ contained in at least one $\sigma$-extension of $F$?

Skeptical Acceptance

Skept$_\sigma$: Given AF $F = (A, R)$ and $a \in A$; is $a$ contained in every $\sigma$-extension of $F$?

If no extension exists then all arguments are skeptically accepted and no argument is credulously accepted$^1$.

Hence we are also interested in the following problem:

Skeptically and Credulously accepted

Skept'$_\sigma$: Given AF $F = (A, R)$ and $a \in A$; is $a$ contained in every and at least one $\sigma$-extension of $F$?

---

$^1$This is only relevant for stable semantics.
### Verifying an extension

**Ver}_\sigma: Given AF F = (A, R) and S \subseteq A; is S a }_\sigma\text{-extension of } F?**

---

---

---

---
### Further Decision Problems

<table>
<thead>
<tr>
<th>Verifying an extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Ver}_\sigma$: Given $AF \ F = (A, R)$ and $S \subseteq A$; is $S$ a $\sigma$-extension of $F$?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Does there exist an extension?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Exists}_\sigma$: Given $AF \ F = (A, R)$; Does there exist a $\sigma$-extension for $F$?</td>
</tr>
</tbody>
</table>

TU Dresden, ICCL Summer School 2017 Abstract Argumentation slide 6 of 50
Further Decision Problems

### Verifying an extension

$\text{Ver}_\sigma$: Given AF $F = (A, R)$ and $S \subseteq A$; is $S$ a $\sigma$-extension of $F$?

### Does there exist an extension?

$\text{Exists}_\sigma$: Given AF $F = (A, R)$; Does there exist a $\sigma$-extension for $F$?

### Does there exist a nonempty extensions?

$\text{Exists}_{\neg \emptyset}^\sigma$: Does there exist a non-empty $\sigma$-extension for $F$?
### Complexity of decision problems in AFs.

<table>
<thead>
<tr>
<th>σ</th>
<th>Cred(_{σ})</th>
<th>Skept(_{σ})</th>
<th>Ver(_{σ})</th>
<th>Exists(_{σ})</th>
<th>Exists(_{σ}^{-0})</th>
</tr>
</thead>
<tbody>
<tr>
<td>cf</td>
<td>in L</td>
<td>trivial</td>
<td>in L</td>
<td>trivial</td>
<td>in L</td>
</tr>
<tr>
<td>naive</td>
<td>in L</td>
<td>in L</td>
<td>in L</td>
<td>trivial</td>
<td>in L</td>
</tr>
<tr>
<td>ground</td>
<td>P-c</td>
<td>P-c</td>
<td>P-c</td>
<td>trivial</td>
<td>P-c</td>
</tr>
<tr>
<td>stable</td>
<td>NP-c</td>
<td>co-NP-c</td>
<td>in L</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
<tr>
<td>adm</td>
<td>NP-c</td>
<td>trivial</td>
<td>in L</td>
<td>trivial</td>
<td>in L</td>
</tr>
<tr>
<td>comp</td>
<td>NP-c</td>
<td>P-c</td>
<td>in L</td>
<td>trivial</td>
<td>NP-c</td>
</tr>
<tr>
<td>cf2</td>
<td>NP-c</td>
<td>co-NP-c</td>
<td>in P</td>
<td>trivial</td>
<td>in L</td>
</tr>
<tr>
<td>ideal</td>
<td>(Θ^P_2)-c</td>
<td>(Θ^P_2)-c</td>
<td>(Θ^P_2)-c</td>
<td>trivial</td>
<td>(Θ^P_2)-c</td>
</tr>
<tr>
<td>pref</td>
<td>NP-c</td>
<td>(Π^P_2)-c</td>
<td>co-NP-c</td>
<td>trivial</td>
<td>NP-c</td>
</tr>
<tr>
<td>semi</td>
<td>(Σ^P_2)-c</td>
<td>(Π^P_2)-c</td>
<td>co-NP-c</td>
<td>trivial</td>
<td>NP-c</td>
</tr>
<tr>
<td>stage</td>
<td>(Σ^P_2)-c</td>
<td>(Π^P_2)-c</td>
<td>co-NP-c</td>
<td>trivial</td>
<td>in L</td>
</tr>
</tbody>
</table>

Most problems in Abstract Argumentation are computationally intractable, i.e. at least NP-hard. To show intractability for a specific reasoning problem we follow the schema given below:

**Goal**: Show that a reasoning problem is NP-hard.

**Method**: Reducing the NP-hard SAT problem to the reasoning problem.

- Consider an arbitrary CNF formula $\Phi$
- Give a reduction that maps $\Phi$ to an Argumentation Framework $F_\Phi$ containing an argument $\Phi$.
- Show that $\Phi$ is satisfiable iff the argument $\Phi$ is accepted.
Canonical Reduction

Definition

For $\Phi = \bigwedge_{i=1}^{m} l_{i1} \lor l_{i2} \lor l_{i3}$ over atoms $Z$, build $F_{\Phi} = (A_{\Phi}, R_{\Phi})$ with

\[
A_{\Phi} = Z \cup \bar{Z} \cup \{C_1, \ldots, C_m\} \cup \{\Phi\}
\]

\[
R_{\Phi} = \{(z, \bar{z}), (\bar{z}, z) \mid z \in Z\} \cup \{(C_i, \Phi) \mid i \in \{1, \ldots, m\}\} \cup \{(z, C_i) \mid i \in \{1, \ldots, m\}, z \in \{l_{i1}, l_{i2}, l_{i3}\}\} \cup \{(\bar{z}, C_i) \mid i \in \{1, \ldots, m\}, \neg z \in \{l_{i1}, l_{i2}, l_{i3}\}\}
\]
Canonical Reduction

Definition

For $\Phi = \bigwedge_{i=1}^{m} l_i \lor l_i \lor l_i$ over atoms $Z$, build $F_{\Phi} = (A_{\Phi}, R_{\Phi})$ with

$$A_{\Phi} = Z \cup \bar{Z} \cup \{C_1, \ldots, C_m\} \cup \{\Phi\}$$

$$R_{\Phi} = \{(z, \bar{z}), (\bar{z}, z) \mid z \in Z\} \cup \{(C_i, \Phi) \mid i \in \{1, \ldots, m\}\} \cup \{(z, C_i) \mid i \in \{1, \ldots, m\}, z \in \{l_i, l_i, l_i\}\} \cup \{(\bar{z}, C_i) \mid i \in \{1, \ldots, m\}, \neg z \in \{l_i, l_i, l_i\}\}$$

Example

Let $\Phi = (z_1 \lor z_2 \lor z_3) \land (\neg z_2 \lor \neg z_3 \lor \neg z_4) \land (\neg z_1 \lor z_2 \lor z_4)$. 

TU Dresden, ICCL Summer School 2017 Abstract Argumentation slide 11 of 50
Theorem

The following statements are equivalent:

1. $\Phi$ is satisfiable
2. $F_\Phi$ has an admissible set containing $\Phi$
3. $F_\Phi$ has a complete extension containing $\Phi$
4. $F_\Phi$ has a preferred extension containing $\Phi$
5. $F_\Phi$ has a stable extension containing $\Phi$
Complexity Results

**Theorem**
1. Cred\(_{\text{stable}}\) is NP-complete
2. Cred\(_{\text{adm}}\) is NP-complete
3. Cred\(_{\text{comp}}\) is NP-complete
4. Cred\(_{\text{pref}}\) is NP-complete

**Proof.**
(1) The hardness is immediate by the last theorem. For the NP-membership we use the following guess & check algorithm:

- Guess a set \( E \subseteq A \)
- verify that \( E \) is stable
  - for each \( a, b \in E \) check \((a, b) \notin R\)
  - for each \( a \in A \setminus E \) check if there exists \( b \in E \) with \((b, a) \in R\)

As this algorithm is in polynomial time we obtain NP-membership. \(\square\)
Outline

1 Complexity of Abstract Argumentation

2 Extending Dung’s Framework

3 Abstract Dialectical Frameworks
Motivation

Observations

For many scenarios, limitations of abstract AFs become apparent

- “positive” (support) links between arguments
- “joint attacks”
- making attacks also subject of evaluation
- weights, priorities, etc.
Motivation

Observations
For many scenarios, limitations of abstract AFs become apparent
- “positive” (support) links between arguments
- “joint attacks”
- making attacks also subject of evaluation
- weights, priorities, etc.

In the literature
- BAFs: Bipolar Argumentation Frameworks (Attack and Support) [1]
- EAFs: Extended Argumentation Frameworks (Attack on Attacks) [6]
- AFRAs: Argumentation Frameworks with Recursive Attacks [2]
## Motivation

### Observations
For many scenarios, limitations of abstract AFs become apparent
- “positive” (support) links between arguments
- “joint attacks”
- making attacks also subject of evaluation
- weights, priorities, etc.

### In the literature
- BAFs: Bipolar Argumentation Frameworks (Attack and Support) [1]
- EAFs: Extended Argumentation Frameworks (Attack on Attacks) [6]
- AFRAs: Argumentation Frameworks with Recursive Attacks [2]

### In the lecture
- ADFs: Abstract Dialectical Frameworks [3]
Outline

1 Complexity of Abstract Argumentation
2 Extending Dung’s Framework
3 Abstract Dialectical Frameworks
Basic Idea

Abstract Dialectical Framework

= 

Dependency Graph + Acceptance Conditions
ADFs - Basic idea

An Argumentation Framework
An Argumentation Framework
with explicit acceptance conditions
ADFs - Basic idea (ctd.)

A Dialectical Framework
with flexible acceptance conditions
ADFs - The Formal Framework

- Like AFs, use graph to describe dependencies among nodes.
- Unlike AFs, allow individual acceptance condition for each node.
- Assigns \( t(\text{true}) \) or \( f(\text{false}) \) depending on status of parents.

**ADF [Brewka and Woltran 2010]**

An abstract dialectical framework (ADF) is a tuple \( D = (S, L, C) \) where

- \( S \) is a set of statements (positions, nodes),
- \( L \subseteq S \times S \) is a set of links,
- \( C = \{C_s\}_{s \in S} \) is a set of total functions \( C_s : 2^{\text{par}(s)} \rightarrow \{t, f\} \), one for each statement \( s \). \( C_s \) is called acceptance condition of \( s \).

Propositional formula representing \( C_s \) denoted \( F_s \). In the remainder: \( (S, C) \)
Example

Person innocent, unless she is a murderer.
A killer is a murderer, unless she acted in self-defense.
Evidence for self-defense needed, e.g. witness not known to be a liar.

Propositionally:
\[ w : \top, \; k : \top, \; l : \bot, \; s : w \land \neg l, \; m : k \land \neg s, \; i : \neg m \]
Argumentation frameworks: a special case

- AFs have attacking links only and a single type of nodes.
- Can easily be captured as ADFs.
- $\mathcal{A} = (AR, attacks)$. Associated ADF $D_\mathcal{A} = (AR, C)$
- $C_s$ as propositional formula:
  $F_s = \neg r_1 \land \ldots \land \neg r_n$, where $r_i$ are the attackers of $s$. 
ADF Semantics

- AF semantics specify for an AF = (A,R) subsets of A: $S \subseteq A$
- We begin with a basic semantics of ADF using interpretations $v : S \rightarrow \{t, f\}$

**Definition**

Let $D = (S, C)$ be an ADF. An interpretation $v$ is a model of $D$ if for all $s \in S$: $v(s) = v(C_s)$.

Less formally: a node is accepted (resp. true) iff its acceptance condition says so.

Notation: $v(\varphi)$ is the evaluation of $\varphi$ under $v$, i.e. $v(\varphi) = \begin{cases} t & \text{if } v \models \varphi \\ f & \text{if } v \not\models \varphi \end{cases}$
Example

Consider $D = (S, C)$ with $S = \{a, b\}$:

- For $C_a = \neg b$, $C_b = \neg a$ (AF): two models, $v_1, v_2$
- For $C_a = b$, $C_b = a$ (mutual support): two models, $v_3, v_4$
- For $C_a = b$ and $C_b = \neg a$ ($a$ attacks $b$, $b$ supports $a$): no model.

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>$v_2$</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>$v_3$</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>$v_4$</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

When $C$ is represented as set of propositional formulas, then models are just propositional models of $\{s \equiv C_s \mid s \in S\}$.
A Short Excursion to Labeling of AFs

- Classical interpretations are not suited for remaining semantics of ADFs
- Extensions of AFs inherently assign to every argument two values: \( \text{in} \) or \( \text{out} \)
- Equivalently one can use labelings [5], which assign three values: \( \text{in} \) \((t)\), \( \text{out} \) \((f)\) and undecided \((u)\)

**Definition**

Given an AF \( F = (A, R) \), a function \( \mathcal{L} : A \rightarrow \{t, f, u\} \) is a complete labeling if the following conditions hold:

- \( \mathcal{L}(a) = t \) iff for each \( b \) with \( (b, a) \in R \), \( \mathcal{L}(b) = f \)
- \( \mathcal{L}(a) = f \) iff there exists \( b \) with \( (b, a) \in R \), \( \mathcal{L}(b) = t \)
Example Labeling

Example

Given the following AF

Then its complete labelings are given by

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{L}_1$</td>
<td>$u$</td>
<td>$u$</td>
<td>$u$</td>
</tr>
<tr>
<td>$\mathcal{L}_2$</td>
<td>$t$</td>
<td>$f$</td>
<td>$u$</td>
</tr>
<tr>
<td>$\mathcal{L}_3$</td>
<td>$f$</td>
<td>$t$</td>
<td>$u$</td>
</tr>
</tbody>
</table>
Characteristic function of AFs gives easy definition of semantics via fixed points and is based on defense

**Definition**

Given an AF $F = (A, R)$. The characteristic function $\mathcal{F}_F : 2^A \rightarrow 2^A$ of $F$ is defined as

$$\mathcal{F}_F(E) = \{ x \in A \mid x \text{ is defended by } E \}$$

- For an AF $F = (A, R)$ we have a conflict-free set $E \subseteq A$ is
  - admissible if $E \subseteq \mathcal{F}_F(E)$
  - grounded if $E$ is lfp of $\mathcal{F}_F$
  - complete if $E = \mathcal{F}_F(E)$
  - preferred if $E$ is $\subseteq$-maximal admissible

- Our goal now: define char. function for ADFs with three-valued interpretations

- For three-values, what does “$\subseteq$” mean? How to compare?
Information Ordering

- In ADFs three-valued interpretations $\nu : S \rightarrow \{t, f, u\}$ are well-suited for defining semantics.
- We can define an information ordering: $u <_i t$ and $u <_i f$.

### Example

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nu_1$</td>
<td>$u$</td>
<td>$u$</td>
<td>$u$</td>
</tr>
<tr>
<td>$\nu_2$</td>
<td>$t$</td>
<td>$f$</td>
<td>$u$</td>
</tr>
<tr>
<td>$\nu_3$</td>
<td>$f$</td>
<td>$t$</td>
<td>$u$</td>
</tr>
</tbody>
</table>

TU Dresden, ICCL Summer School 2017 Abstract Argumentation
A Characteristic Function for ADFs

- Our goal: define a characteristic function for ADFs [7] like for AFs
- Intuitively, $u$ means a not yet decided value
- Let $[v]_2$ be the set of all two-valued interpretations that extend $v$, i.e., $\{v' \mid v \leq_i v', v' \text{ two-valued}\}$
- Special case: if $v$ is two-valued then $[v]_2 = v$

Example

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>$u$</td>
</tr>
<tr>
<td>$v_2$</td>
<td>$t$</td>
</tr>
<tr>
<td>$v_3$</td>
<td>$f$</td>
</tr>
</tbody>
</table>

$[v_1]_2 = \{v_2, v_3\}$, $[v_2]_2 = v_2$ and $[v_3]_2 = v_3$
[\nu]_2\text{ denotes the set of interpretations that refine } \nu, \text{ i.e. set } \nu \text{ to true or false.}

Given \nu \text{ and a boolean formula } C_s \text{ for a statement } s, \text{ we might have different outcomes for each } \nu_1, \nu_2 \in [\nu]_2.

E.g. \nu_1(C_s) \neq \nu_2(C_s), \text{ hence how to update the status of } s \text{ given } \nu?\text{ }

Idea: compute a “consensus.”

The set \{t, f, u\} forms a meet-semilattice w.r.t. \prec_i, \text{ i.e. take as consensus the meet } (\sqcap), \text{ i.e. } t \sqcap t = t, f \sqcap f = f \text{ and } u \text{ otherwise.}
For the characteristic function for ADFs we now take the consensus of $[v]_2$ applied to $C_s$:

**Definition**

$\Gamma_D(v)$ is given by

$s \mapsto \prod_{w \in [v]_2} w(C_s)$

**Example**

Let $C_a = \neg a$ and $v(a) = u$, then $[v]_2 = \{v_2, v_3\}$

$v_2(C_a) = f$

$v_3(C_a) = t$

The result is $\prod_{w \in [v]_2} w(C_a) = u$
Example

Let $C_a = \top$ and $v(a) = u$, then $[v]_2 = \{v_2, v_3\}$

\[
\begin{array}{c|c}
  v & a \\
  \hline
  v_2 & u \\
  v_3 & t \\
\end{array}
\]

$v_2(C_a) = t = v_3(C_a)$

the result is $\bigcap_{w \in [v]_2} w(C_a) = t$
Example

Let $C_a = a \lor b$ and $v(a) = t$, $v(b) = u$, then $[v]_2 = \{v_2, v_3\}$

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>$t$</td>
<td>$u$</td>
</tr>
<tr>
<td>$v_2$</td>
<td>$t$</td>
<td>$t$</td>
</tr>
<tr>
<td>$v_3$</td>
<td>$t$</td>
<td>$f$</td>
</tr>
</tbody>
</table>

$v_2(C_a) = t = v_3(C_a)$

the result is $\bigcap_{w \in [v]_2} w(C_a) = t$

- Here $v$ incorporates already information: $v(a) = t$
ADF Semantics

- Using the concept of consensus and information ordering, we can define admissible sets, grounded, complete and preferred models similarly as for AFs.

**Definition**

Let $D = (S, C)$ be an ADF and $v$ a three-valued interpretation over $S$, then

- $v$ is **admissible** in $D$ if $v \leq_i \Gamma_D(v)$
- $v$ is the **grounded model** of $D$ if $v$ is the lfp of $\Gamma_D$ wrt $<_i$
- $v$ is **complete** in $D$ if $v = \Gamma_D(v)$
- $v$ is **preferred** in $D$ if $v$ is $<_i$-maximal admissible
ADF Semantics

- Using the concept of consensus and information ordering, we can define admissible sets, grounded, complete and preferred models similarly as for AFs

**Definition**

Let $D = (S, C)$ be an ADF and $v$ a three-valued interpretation over $S$, then

- $v$ is **admissible** in $D$ if $v \leq_i \Gamma_D(v)$
- $v$ is the **grounded model** of $D$ if $v$ is the lfp of $\Gamma_D$ wrt $<_i$
- $v$ is **complete** in $D$ if $v = \Gamma_D(v)$
- $v$ is **preferred** in $D$ if $v$ is $<_i$-maximal admissible

Remember for AFs we have:

- admissible if $E \subseteq F_F(E)$
- grounded if $E$ is lfp of $F_F$
- complete if $E = F_F(E)$
- preferred if $E$ is $\subseteq$-maximal admissible
Example

Let $C_a = \top$, $C_b = a$, $C_c = c \land b$, $C_d = \neg d$

Then the complete models are given by:

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>$t$</td>
<td>$t$</td>
<td>$u$</td>
<td>$u$</td>
<td>$grd, com$</td>
</tr>
<tr>
<td>$v_2$</td>
<td>$t$</td>
<td>$t$</td>
<td>$t$</td>
<td>$u$</td>
<td>$com, prf$</td>
</tr>
<tr>
<td>$v_3$</td>
<td>$t$</td>
<td>$t$</td>
<td>$f$</td>
<td>$u$</td>
<td>$com, prf$</td>
</tr>
</tbody>
</table>
Remarks about Expressibility

• Acceptance conditions of ADFs also allow definitions of preference relations
• Argument $A$ has a higher priority than $B$: $C_B = \varphi \land (B \rightarrow A)$
• In general: given preferences can be “compiled” to an ADF
• “Joint attacks” can be modeled: set of statements $X$ attack $a$ if $C_a = \neg (\bigwedge_{x \in X} x)$
• Every ADF can be simulated by an AF such that the models of the ADF are in correspondence to the stable extensions of the AF [4].

• Idea from boolean circuits: for each statement $s$ we construct its $C_s$:

$$(a \land b) \lor \neg c$$

• The size of the resulting AF is polynomially bounded wrt to size of ADF.
1 Complexity of Abstract Argumentation

2 Extending Dung’s Framework

3 Abstract Dialectical Frameworks
   Weights and preference
   Computational Complexity of ADFs
Weights for ADFs

- So far: acceptance conditions defined via actual parents. Now: via properties of links represented as weights.

- Add function $w : L \to V$, where $V$ is some set of weights.

- Simplest case: $V = \{+,-\}$. Possible acceptance conditions:
  - $C_s(R) = \text{in}$ iff no negative link from elements of $R$ to $s$,
  - $C_s(R) = \text{in}$ iff no negative and at least one positive link from $R$ to $s$,
  - $C_s(R) = \text{in}$ iff more positive than negative links from $R$ to $s$.

- More fine grained distinctions if $V$ is numerical:
  - $C_s(R) = \text{in}$ iff sum of weights of links from $R$ to $s$ positive,
  - $C_s(R) = \text{in}$ iff maximal positive weight higher than maximal negative weight,
  - $C_s(R) = \text{in}$ iff difference between maximal positive weight and (absolute) maximal negative weight above threshold.
Prioritized ADFs

- Another way of defining acceptance: qualitative preferences among a node’s parents.

- Let $D = (S, L, C)$. Assume for each $s \in S$ strict partial order $>_s$ over parents of $s$.

- Let $C_s(R) = in$ iff for each attacking node $r \in R$ there is a supporting node $r' \in R$ such that $r' >_s r$.

- Node $out$ unless joint support more preferred than joint attack.

- Can reverse this by defining $C_s(R) = out$ iff for each supporting node $r \in R$ there is an attacking node $r' \in R$ such that $r' >_s r$.

- Now node $in$ unless its attackers are jointly preferred.

- Can have both kinds in single prioritized ADF.
Outline

1. Complexity of Abstract Argumentation
2. Extending Dung’s Framework
3. Abstract Dialectical Frameworks
   - Weights and preference
   - Computational Complexity of ADFs
## Computational Problems

### Credulous Acceptance

\[ \text{Cred}_\sigma : \text{Given } ADF \ D = (S, L, C) \text{ and } a \in S; \text{ is there an interpretation } I \in \sigma(D) \text{ with } I(a) = t? \]

### Skeptical Acceptance

\[ \text{Skept}_\sigma : \text{Given } ADF \ D = (S, L, C) \text{ and } a \in S; \text{ is } I(a) = t \text{ for each interpretation } I \in \sigma(D)? \]
Further Computational Problems

**Verification of an interpretation**

$\text{Ver}_\sigma$: Given ADF $D = (S, L, C)$ and an interpretation $I$; is $I \in \sigma(D)$?

**Existence of an interpretation**

$\text{Exists}_\sigma$: Given ADF $D = (S, L, C)$; is $\sigma(D) \neq \emptyset$?

**Existence of a nonempty interpretation**

$\text{Exists}_{\neg\emptyset}: \text{Given ADF } D = (S, L, C); \text{ does there exist an interpretation } I \in \sigma(D) \text{ with } I(s) = t \text{ for some statement } a \in S?$
### Complexity of ADFs

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$\text{Cred}_\sigma$</th>
<th>$\text{Skept}_\sigma$</th>
<th>$\text{Ver}_\sigma$</th>
<th>$\text{Exists}_\sigma$</th>
<th>$\text{Exists}_{\sigma \neg \emptyset}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ground</td>
<td>co-NP-c</td>
<td>co-NP-c</td>
<td>DP-c</td>
<td>trivial</td>
<td>co-NP-c</td>
</tr>
<tr>
<td>model</td>
<td>NP-c</td>
<td>co-NP-c</td>
<td>in P</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
<tr>
<td>adm</td>
<td>$\Sigma_2^P$-c</td>
<td>trivial</td>
<td>co-NP-c</td>
<td>trivial</td>
<td>$\Sigma_2^P$-c</td>
</tr>
<tr>
<td>comp</td>
<td>$\Sigma_2^P$-c</td>
<td>co-NP-c</td>
<td>DP-c</td>
<td>trivial</td>
<td>$\Sigma_2^P$-c</td>
</tr>
<tr>
<td>pref</td>
<td>$\Sigma_2^P$-c</td>
<td>$\Pi_3^P$-c</td>
<td>$\Pi_2^P$-c</td>
<td>trivial</td>
<td>$\Sigma_2^P$-c</td>
</tr>
</tbody>
</table>
[1] Leila Amgoud and Claudette Cayrol and Marie-Christine Lagasquie and Pierre Livet,
On Bipolarity in Argumentation Frameworks

AFRA: Argumentation Framework with Recursive Attacks.

Abstract Dialectical Frameworks.
Proceedings of the 12th International Conference on the Principles of Knowledge Representation

Relating the Semantics of Abstract Dialectical Frameworks and Standard AFs.
Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI’11), 2011.

A logical account of formal argumentation.

Reasoning about Preferences in Argumentation Frameworks.

Approximating Operators and Semantics for Abstract Dialectical Frameworks.