SEMINARY ABSTRACT
ARGUMENTATION

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

Sarah Gaggl

Dresden, 15th November 2013
Motivation

<table>
<thead>
<tr>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>For many scenarios, limitations of abstract AFs become apparent</td>
</tr>
<tr>
<td>● “positive” (support) links between arguments</td>
</tr>
<tr>
<td>● “joint attacks”</td>
</tr>
<tr>
<td>● making attacks also subject of evaluation</td>
</tr>
<tr>
<td>● weights, priorities, etc.</td>
</tr>
</tbody>
</table>
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]
Basic Idea

Abstract Dialectical Framework

= 

Dependency Graph + Acceptance Conditions
ADFs - Basic idea

An Argumentation Framework
An Argumentation Framework with explicit acceptance conditions
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 $t$(rue) or $f$(alse) depending on status of parents.

**Definition**

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

\[ \nu : S \rightarrow \{ t, f \} \]

**Definition**

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

\[ \nu(s) = \nu(C_s). \]

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

Notation: \( \nu(\varphi) \) is the evaluation of \( \varphi \) under \( \nu \), i.e.

\[
\nu(\varphi) = \begin{cases} 
  t & \text{if } \nu \models \varphi \\
  f & \text{if } \nu \not\models \varphi 
\end{cases}
\]
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: \textit{in} or \textit{out}
- Equivalently one can use labelings [5], which assign three values: \textit{in} (t), \textit{out} (f) and \textit{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 $(b, a) \in R$, $\mathcal{L}(b) = f$
- $\mathcal{L}(b) = f$ 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 AF Semantics

- 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>( \nu_1 )</th>
<th>( a )</th>
<th>( b )</th>
<th>( c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_1 )</td>
<td>( u )</td>
<td>( u )</td>
<td>( u )</td>
</tr>
<tr>
<td>( v_2 )</td>
<td>( t )</td>
<td>( f )</td>
<td>( u )</td>
</tr>
<tr>
<td>( v_3 )</td>
<td>( f )</td>
<td>( t )</td>
<td>( u )</td>
</tr>
</tbody>
</table>

\( v_2 \) \( v_1 \) \( v_3 \)
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 \( \{ v' \mid v \leq_i v', v' \text{ two-valued} \} \)
- Special case: if \( v \) is two-valued then \([v]_2 = v\)

**Example**

<table>
<thead>
<tr>
<th>( v )</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>

\([v_1]_2 = \{ v_2, v_3 \}, [v_2]_2 = v_2 \text{ and } [v_3]_2 = v_3\)
• $[\nu]_2$ denotes the set of interpretations that refine $\nu$, i.e. set $u$ to true or false
• Given $\nu$ and a boolean formula $C_s$ for a statement $s$, we might have different outcomes for each $\nu_1, \nu_2 \in [\nu]_2$
• E.g. $\nu_1(C_s) \neq \nu_2(C_s)$, hence how to update the status of $s$ given $\nu$?
• Idea: compute a “consensus”
• The set $\{t, f, u\}$ forms a meet-semilattice w.r.t. $<$, i.e. take as consensus the meet ($\sqcap$)
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 $[v]_2 = \{v_2, v_3\}$

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</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>

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</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>

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

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>

$\forall w \in [v]_2, w(C_a) = t$

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

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

Then the complete models are given by:

<table>
<thead>
<tr>
<th>$v_1$</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 $C_a = \neg(\bigwedge_{x \in X} x)$
ADF Simulation via AF

- 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 $C_s$:

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

- The size of the resulting AF is polynomially bounded wrt to size of ADF.
[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.