Dialectical Frameworks: Argumentation Beyond Dung

Gerhard Brewka

Computer Science Institute
University of Leipzig
brewka@informatik.uni-leipzig.de

joint work with Stefan Woltran
1. Introduction

- Argumentation a hot topic in logic based AI
- Highly successful: Dung’s abstract argumentation frameworks
- AFs provide account of how to select acceptable arguments given arguments with attack relation
- Abstract away from everything but attacks
- Can be instantiated in many different ways
- Useful analytical tool and target system for translations
Common Use of AFs in Argumentation

- Prototypical example: Prakken (2010)
- Given: KB consisting of defeasible rules, preferences, types of statements, proof standards etc.
- Available information compiled into adequate arguments and attacks
- Resulting AF provides system with choice of semantics

Our goal: bring target system closer to original KB, so as to make compilation easy

Like AFs, new target systems must come with semantics!
Common Use of AFs in Argumentation

- Prototypical example: Prakken (2010)
- Given: KB consisting of defeasible rules, preferences, types of statements, proof standards etc.
- Available information compiled into adequate arguments and attacks
- Resulting AF provides system with choice of semantics

Our goal: bring target system closer to original KB, so as to make compilation easy

Like AFs, new target systems must come with semantics!
Common Use of AFs in Argumentation

- Prototypical example: Prakken (2010)
- Given: KB consisting of defeasible rules, preferences, types of statements, proof standards etc.
- Available information compiled into adequate arguments and attacks
- Resulting AF provides system with choice of semantics

```
KB  AF
```

- Our goal: bring target system closer to original KB, so as to make compilation easy
- Like AFs, new target systems must come with semantics!
• Our initial interest: proof standards
• Introduced in 2 steps: (1) add acceptance conditions, (2) define them in domain independent way
• Leads to surprisingly powerful generalization
• Dung’s semantics can be generalized accordingly
• Shares motivation with bipolar AFs (Cayrol, Lagasquie-Schiex, Amgoud) yet more general and flexible
• Our initial interest: proof standards
• Introduced in 2 steps: (1) add acceptance conditions, (2) define them in domain independent way
• Leads to surprisingly powerful generalization
• Dung’s semantics can be generalized accordingly
• Shares motivation with bipolar AFs (Cayrol, Lagasquie-Schiex, Amgoud) yet more general and flexible

Abstract Dialectical Framework

= Dependency Graph + Acceptance Conditions
2. Background: Dung argumentation frameworks

- Graph, nodes are arguments, links represent attack
- Intuition: node accepted unless attacked
- Arguments not further analyzed

Example

- Semantics select acceptable sets $E$ of arguments (extensions):
  - grounded: (1) accept unattacked args, (2) delete args attacked by accepted args, (3) goto 1, stop when fixpoint reached.
  - preferred: maximal conflict-free sets attacking all their attackers
  - stable: conflict free sets attacking all unaccepted args.
Restrictions of AFs

Example

- fixed meaning of links: attack
- fixed acceptance condition for args: no parent accepted
- want more flexibility:
  1. links supporting arguments/positions
  2. nodes not accepted unless supported
  3. flexible means of combining attack and support

Observation: Dung’s acceptance condition for node $n$ boolean function in parents of $n$, e.g. $b \equiv \neg a \land \neg b$. 
Like Dung, use graph to describe dependencies among nodes
Unlike Dung, allow individual acceptance condition for each node
Assigns \textit{in} or \textit{out} depending on status of parents

**Definition**

An \textit{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 $C_s : 2^{\text{par}(s)} \rightarrow \{\text{in}, \text{out}\}$, one for each statement $s$. $C_s$ is called acceptance condition of $s$. 

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.

\[ w \text{ and } k \text{ known (}\text{in}\text{), } l \text{ not known (}\text{out}\text{)} \]
Other nodes: \text{in} iff all + parents \text{in}, all - parents \text{out}.
Dung frameworks: a special case

- AFs have attacking links only and a single type of nodes.
- Can easily be captured as ADFs.
- \( \mathcal{A} = (AR, \text{attacks}) \). Associated ADF \( D_\mathcal{A} = (AR, \text{attacks}, C) \):
  for all \( s \in AR \), \( C_s(R) = \text{in} \) iff \( R = \emptyset \).
- Acceptance conditions boolean functions, \( C_s \) conveniently represented as propositional formula \( F_s \).
  for AFs: \( F_s = \neg r_1 \land \ldots \land \neg r_n \), where \( r_i \) are the attackers of \( s \).
Models

Definition

Let $D = (S, L, C)$ be an ADF.

- $M \subseteq S$ is called conflict-free (in $D$) if for all $s \in M$ we have $C_s(M \cap \text{par}(s)) = \text{in}$.
- $M \subseteq S$ is a model of $D$ if $M$ is conflict-free and for each $s \in S$, $C_s(M \cap \text{par}(s)) = \text{in}$ implies $s \in M$.

In other words, $M \subseteq S$ is a model of $D = (S, L, C)$ if for all $s \in S$ we have $s \in M$ iff $C_s(M \cap \text{par}(s)) = \text{in}$.
Consider $D = (S, L, C)$ with $S = \{a, b\}$, $L = \{(a, b), (b, a)\}$:

\[
\begin{array}{ccc}
  a & \leftrightarrow & b
\end{array}
\]

- For $C_a(\emptyset) = C_b(\emptyset) = \text{in}$ and $C_a(\{b\}) = C_b(\{a\}) = \text{out}$ (Dung AF): two models, $M_1 = \{a\}$ and $M_2 = \{b\}$.
- For $C_a(\emptyset) = C_b(\emptyset) = \text{out}$ and $C_a(\{b\}) = C_b(\{a\}) = \text{in}$ (mutual support): $M_3 = \emptyset$ and $M_4 = \{a, b\}$.
- For $C_a(\emptyset) = C_b(\{a\}) = \text{out}$ and $C_a(\{b\}) = C_b(\emptyset) = \text{in}$ ($a$ attacks $b$, $b$ supports $a$): no model.

When $C$ is represented as set of propositional formulas $F(s)$, then models are just propositional models of $\{s \equiv F(s) \mid s \in S\}$.
Let $\mathcal{A} = (AR, \text{attacks})$ be an AF, $D_A = (S, L, C)$ its associated dialectical framework, and $E \subseteq AR$.

1. $E$ is conflict-free in $\mathcal{A}$ iff $E$ is conflict-free in $D_A$;
2. $E$ is a stable extension of $\mathcal{A}$ iff $E$ is a model of $D_A$.

For more general ADFs, models and stable models will be different.
Grounded semantics

Definition

For $D = (S, L, C)$, let $\Gamma_D(A, R) = (\text{acc}(A, R), \text{reb}(A, R))$ where

$$\text{acc}(A, R) = \left\{ r \in S | A \subseteq S' \subseteq (S \setminus R) \Rightarrow C_r(S' \cap \text{par}(r)) = \text{in} \right\}$$

$$\text{reb}(A, R) = \left\{ r \in S | A \subseteq S' \subseteq (S \setminus R) \Rightarrow C_r(S' \cap \text{par}(r)) = \text{out} \right\}.$$ 

$\Gamma_D$ monotonic in both arguments, thus has least fixpoint. $E$ is the well-founded model of $D$ iff for some $E' \subseteq S$, $(E, E')$ least fixpoint of $\Gamma_D$.

First (second) argument collects nodes known to be in (out). Starting with $(\emptyset, \emptyset)$, iterations add $r$ to first (second) argument whenever status of $r$ must be in (out) whatever the status of undecided nodes.

Generalizes grounded semantics, more precisely: ultimate well-founded semantics by Denecker, Marek, Truszczyński.
4. Stable models and bipolar ADFs

- Stable models in LP exclude *self-supporting cycles*
- May appear in ADF models, not captured by minimality.

**Example**

Let $D = (S, L, P)$ with $S = \{a, b, c\}$, $L = \{(a, b), (b, a), (b, c)\}$:

\[ C_a(\emptyset) = C_b(\emptyset) = \text{out and } C_a(\{b\}) = C_b(\{a\}) = \text{in} \text{ (mutual support)}, \]
\[ C_c(\emptyset) = \text{in and } C_c(\{b\}) = \text{out} \text{ (attack)}. \]

$M = \{a, b\}$ model, however $a$ in because $b$ is, $b$ in because $a$ is.

- Need notion of *supporting link*
- Apply construction similar to Gelfond/Lifschitz reduct.
Bipolar ADFs

Definition

Let $D = (S, L, C)$ be an ADF. A link $(r, s) \in L$ is

1. **supporting**: for no $R \subseteq \text{par}(s)$, $C_s(R) = \text{in}$ and $C_s(R \cup \{r\}) = \text{out}$,
2. **attacking**: for no $R \subseteq \text{par}(s)$, $C_s(R) = \text{out}$ and $C_s(R \cup \{r\}) = \text{in}$.

- $D$ is called *bipolar* if all of its links are supporting or attacking.
- $D$ is called *monotonic* if all of its links are supporting.
- If $D$ is monotonic, then it has a unique least model.
Stable models

Definition

Let $D = (S, L, C)$ be a BADF. A model $M$ of $D$ is a stable model if $M$ is the least model of the reduced ADF $D^M$ obtained from $D$ by

1. eliminating all nodes not contained in $M$ together with all links in which any of these nodes appear,
2. eliminating all attacking links,
3. restricting the acceptance condition $C_s$ for each remaining node $s$ to the remaining parents of $s$.

Remark: for BADFs representing Dung AFs, models and stable models coincide.
Consider $D$ where $a$ supports $b$, $b$ supports $a$, and $b$ attacks $c$: $a$ is $in$ iff $b$ is and vice versa. Moreover, $c$ is $in$ unless $b$ is.

\begin{center}
\begin{tikzpicture}[node distance=1.5cm,auto]
  
  \node (a) {$a$};
  \node (b) [right of=a] {$b$};
  \node (c) [right of=b] {$c$};

  \path[->] (a) edge (b);
  \path[->] (b) edge (a);
  \path[->] (b) edge (c);

\end{tikzpicture}
\end{center}

Get two models: $\{a, b\}$ and $\{c\}$. Only the latter is expected.

The reduct of $D$ wrt $\{a, b\}$ is $\left(\{a, b\}, \{(a, b), (b, a)\}, \{C_a, C_b\}\right)$ where $C_a, C_b$ are as described above. Reduct has $\emptyset$ as least model. $\{a, b\}$ thus not stable.

On the other hand, the reduct $D^{\{c\}}$ has no link at all. According to its acceptance condition $c$ is $in$; we thus have a stable model.
Preferred semantics

- Dung: preferred extension = maximal admissible set.
- Admissible set: conflict-free, defends itself against attackers.
- Can show: $E$ admissible in $\mathcal{A} = (AR, att)$ iff for some $R \subseteq AR$
  - $R$ does not attack $E$, and
  - $E$ stable extension of $(AR-R, att \cap (AR-R \times AR-R))$.

Definition

Let $D = (S, L, C)$, $R \subseteq S$. $D-R$ is the BADF obtained from $D$ by

1. deleting all nodes in $R$ together with their proof standards and links they are contained in.
2. restricting proof standards of remaining nodes to remaining parents.
Definition

Let $D = (S, L, C)$ be a BADF. $M \subseteq S$ admissible in $D$ iff there is $R \subseteq S$ such that

1. no element in $R$ attacks an element in $M$, and
2. $M$ is a stable model of $D-R$.

$M$ is a preferred model of $D$ iff $M$ is (inclusion) maximal among the sets admissible in $D$.

Results

- BADFs have at least one preferred model.
- Each stable model is a preferred model.
- Generalize preferred extensions of AFs adequately.
5. Complexity results

$D$ is ADF, acceptance conditions given as propositional formulas:
- Deciding whether $M$ is well-founded model of $D$ coNP-hard.
- Deciding whether $D$ is bipolar coNP-hard.

$D$ is BADF with supporting links $L^+$ and attacking links $L^-$:
- Deciding whether $M$ is well-founded model of $D$ polynomial.
- Deciding whether $s$ is contained in some (resp. all) stable models of $D$ NP-complete (resp. coNP-complete).
- Deciding whether $s$ is contained in some (resp. all) preferred models of $D$ NP-complete (resp. $\Pi_2^P$-complete).

Bottom line: no increase in complexity once attacking/supporting links are known.
6. Relationship to LPs

- Cannot represent LP rules as direct dependencies among atoms:
  \[
  \{ c \leftarrow a, \text{not } b; \ c \leftarrow \text{not } a, b \}\]

- Links \((a, c)\) and \((b, c)\) neither supporting nor attacking, no BADF.

- Get BADF if rule explicitly represented as additional node:

  ![Diagram](attachment:diagram.png)

  \[
  a \rightarrow r_1 \rightarrow c \rightarrow r_2 \rightarrow b
  \]

- Resulting ADFs bipolar → any of the defined semantics work.
- Models in one-to-one correspondence (upto rule nodes).
- In principle, “bipolarization” possible for arbitrary ADFs, but exponential blowup - unlike for LP rules.
6. Relationship to LPs

- Cannot represent LP rules as direct dependencies among atoms:
  \[
  \{ c \leftarrow a, \text{not } b; \ c \leftarrow \text{not } a, b \} 
  \]

- Links \((a, c)\) and \((b, c)\) neither supporting nor attacking, no BADF.
- Get BADF if rule explicitly represented as additional node:

  ![Diagram](image)

  - Resulting ADFs bipolar \(\Rightarrow\) any of the defined semantics work.
  - Models in one-to-one correspondence (upto rule nodes).
  - In principle, "bipolarization" possible for arbitrary ADFs, but exponential blowup - unlike for LP rules.
6. Relationship to LPs

- Cannot represent LP rules as direct dependencies among atoms:
  \[ \{ c \leftarrow a, \textit{not } b; \ c \leftarrow \textit{not } a, b \} \]

- Links \((a, c)\) and \((b, c)\) neither supporting nor attacking, no BADF.
- Get BADF if rule explicitly represented as additional node:

```
  a---r1
   |
   |
  b--r2
```

- Resulting ADFs bipolar ⇒ any of the defined semantics work.
- Models in one-to-one correspondence (upto rule nodes).
- In principle, "bipolarization" possible for arbitrary ADFs, but exponential blowup - unlike for LP rules.
The other way around

- Can also represent ADFs as LPs: for each $M = \{v_1, \ldots, v_n\}$ such that $C_s(M) = in$, add the rule

$$s \leftarrow v_1, \ldots, v_n, \text{not } w_1, \ldots, \text{not } w_k$$

where the $w_i$ are the parents of $s$ not in $M$.

Mirek’s question: why not use LPs right away?

- ANS1: is the same as asking LP people: why not default logic?
- ANS2: small is beautiful, may lead to new ideas - see Dung, preferred semantics
- ANS3: graphs are beautiful - see Dung, widespread use of AFs
- ANS4: believe gain in expressiveness achieved by giving up some of AFs simplicity is worthwhile.
• Acceptance conditions conveniently defined through weights.

• Add function $w : L \rightarrow V$, 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.
Introduced (1995) model of legal argumentation which distinguishes 4 types of arguments:

- **valid** arguments based on deductive inference,
- **strong** arguments based on inference with defeasible rules,
- **credible** arguments where premises give some evidence,
- **weak** arguments based on abductive reasoning.

By using values $V = \{+v, +s, +c, +w, -v, -s, -c, -w\}$ we can distinguish pro and con links of corresponding types.
• **Scintilla of Evidence**: at least one weak, defendable argument. 
\[ C_s(R) = \text{in} \iff \exists r \in R : w(r, s) \in \{+v, +s, +c, +w\} \].

• **Preponderance of Evidence**: at least one weak, defendable argument that outweighs the other side’s argument: 
\[ C_s(R) = \text{in} \iff \begin{align*} &\exists r \in R : w(r, s) \in \{+v, +s, +c, +w\} \text{ and} \\ &\neg \exists r \in R : w(r, s) = -v \text{ and} \\ &\exists r \in R : w(r, s) = -s \text{ implies } \exists r' \in R : w(r', s) = +v \text{ and} \\ &\exists r \in R : w(r, s) = -c \text{ implies } \exists r' \in R : w(r', s) \in \{+v, +s\} \text{ and} \\ &\exists r \in R : w(r, s) = -w \text{ implies } \exists r' \in R : w(r', s) \in \{+v, +s, +c\}. \end{align*} \]

• **Dialectical Validity**: at least one credible, defendable argument and the other side’s arguments are all defeated: 
\[ C_s(R) = \text{in} \iff \begin{align*} &\exists r \in R : w(r, s) \in \{+v, +s, +c, +w\} \text{ and} \\ &w(t, s) \not\in \{-v, -s, -c, -w\} \text{ for all } t \in R. \end{align*} \]
• **Beyond Reasonable Doubt**: at least one strong, defendable argument and the other side’s arguments all defeated: $C_s(R) = \text{in}$ iff
  - $\exists r \in R : w(r, s) \in \{+v, +s\}$ and
  - $w(t, s) \not\in \{-v, -s, -c, -w\}$ for all $t \in R$.

• **Beyond Doubt**: at least one valid argument and the other side’s arguments all defeated: $C_s(R) = \text{in}$ iff
  - $\exists r \in R : w(r, s) = +v$ and
  - $w(t, s) \not\in \{-v, -s, -c, -w\}$ for all $t \in R$. 

G. Brewka (Leipzig)
Prioritized ADFs

- Another option: qualitative preferences.
- 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 BADF.
8. An Application: Reconstructing Carneades

- Advanced model of argumentation (Gordon, Prakken, Walton 07)
- Graphical representation of statements and their relationships
- Statements have proof standard, some paraconsistency at work
- Major restriction: no cycles (case where Dung semantics coincide)
Why a Reconstruction?

- Goals:
  - putting Carneades on safe formal ground
  - lifting its restriction to acyclic graphs; GPM07: *leave an extension to graphs that allow for cycles ... for future work*
  - providing it with any of the Dung semantics

- Result: reconstruction successful (not written up yet)

- Benefits
  - for Carneades: can now have arbitrary cycles, choice of semantics
  - for ADFs: demonstration of suitability as analytical and semantical tool in argumentation
9. Conclusions

- Presented ADFs, a powerful generalization of Dung frameworks.
- Flexible acceptance conditions for nodes model variety of link and node types.
- Grounded semantics extended to arbitrary ADFs.
- Stable and preferred semantics need restriction to bipolar ADFs.
- Encouraging complexity results.
- Weighted ADFs allow for convenient definition of domain independent proof standards.
- Easy to integrate qualitative preferences.
- Discussed relationship to LPs and application to Carneades.
9. Conclusions

- Presented ADFs, a powerful generalization of Dung frameworks.
- Flexible acceptance conditions for nodes model variety of link and node types.
  - Grounded semantics extended to arbitrary ADFs.
  - Stable and preferred semantics need restriction to bipolar ADFs.
  - Encouraging complexity results.
- Weighted ADFs allow for convenient definition of domain independent proof standards.
- Easy to integrate qualitative preferences.
- Discussed relationship to LPs and application to Carneades.
9. Conclusions

• Presented ADFs, a powerful generalization of Dung frameworks.

• Flexible acceptance conditions for nodes model variety of link and node types.

• Grounded semantics extended to arbitrary ADFs.

• Stable and preferred semantics need restriction to bipolar ADFs.

• Encouraging complexity results.

• Weighted ADFs allow for convenient definition of domain independent proof standards.

• Easy to integrate qualitative preferences.

• Discussed relationship to LPs and application to Carneades.
9. Conclusions

- Presented ADFs, a powerful generalization of Dung frameworks.
- Flexible acceptance conditions for nodes model variety of link and node types.
- Grounded semantics extended to arbitrary ADFs.
- Stable and preferred semantics need restriction to bipolar ADFs.
- Encouraging complexity results.
- Weighted ADFs allow for convenient definition of domain independent proof standards.
- Easy to integrate qualitative preferences.
- Discussed relationship to LPs and application to Carneades.
9. Conclusions

- Presented ADFs, a powerful generalization of Dung frameworks.
- Flexible acceptance conditions for nodes model variety of link and node types.
- Grounded semantics extended to arbitrary ADFs.
- Stable and preferred semantics need restriction to bipolar ADFs.
- Encouraging complexity results.
- Weighted ADFs allow for convenient definition of domain independent proof standards.
- Easy to integrate qualitative preferences.
- Discussed relationship to LPs and application to Carneades.
9. Conclusions

- Presented ADFs, a powerful generalization of Dung frameworks.
- Flexible acceptance conditions for nodes model variety of link and node types.
- Grounded semantics extended to arbitrary ADFs.
- Stable and preferred semantics need restriction to bipolar ADFs.
- Encouraging complexity results.
- Weighted ADFs allow for convenient definition of domain independent proof standards.
- Easy to integrate qualitative preferences.
- Discussed relationship to LPs and application to Carneades.
9. Conclusions

- Presented ADFs, a powerful generalization of Dung frameworks.
- Flexible acceptance conditions for nodes model variety of link and node types.
- Grounded semantics extended to arbitrary ADFs.
- Stable and preferred semantics need restriction to bipolar ADFs.
- Encouraging complexity results.
- Weighted ADFs allow for convenient definition of domain independent proof standards.
- Easy to integrate qualitative preferences.
- Discussed relationship to LPs and application to Carneades.
Future Work

• Generalize other semantics for Dung frameworks, e.g. semi-stable or ideal semantics.

• Investigate computational methods for ADFs
  • can available AF labeling methods be adjusted?
  • splitting results for ADFs?

• Demonstrate suitability of BADFs as analytical and semantical tools in argumentation.

THANK YOU!
Future Work

• Generalize other semantics for Dung frameworks, e.g. semi-stable or ideal semantics.

• Investigate computational methods for ADFs
  • can available AF labeling methods be adjusted?
  • splitting results for ADFs?

• Demonstrate suitability of BADFs as analytical and semantical tools in argumentation.

THANK YOU!