Non-Probabilistic Graphical Models: Nonmonotonic Tools for Argumentation

Gerhard Brewka

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

joint work with Stefan Woltran

Outline

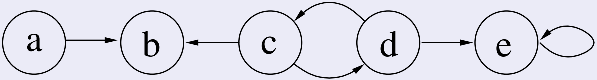
1. Background and Motivation

- Argumentation one of the highly active areas in nonmon
- Dung's abstract argumentation frameworks (AFs) a *gold standard* in argumentation
- Provide account of how to select acceptable arguments given arguments with attack relation
- Abstract away from everything but attacks: calculus of opposition
- Can be instantiated in may different ways
- Useful analytical tool and target system for translations

Argumentation frameworks: 1 slide crash course

- 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.

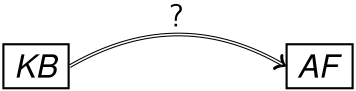
Common Use of AFs in Argumentation

- 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

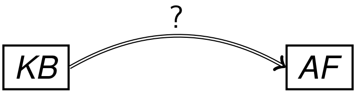
- Prototypical example: Prakken (2010)
- Given: KB consisting of defeasible rules, preferences, types of statements, proof standards etc.
- Available information compiled into adequate arguments and attacks
- Resulting AF provides system with choice of semantics



- 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

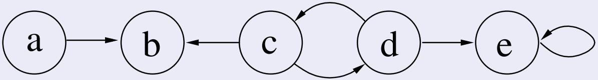
- Prototypical example: Prakken (2010)
- Given: KB consisting of defeasible rules, preferences, types of statements, proof standards etc.
- Available information compiled into adequate arguments and attacks
- Resulting AF provides system with choice of semantics



- Our goal: bring target system closer to original KB, so as to make compilation easy
- Like AFs, new target systems must come with semantics!

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
- from *calculus of opposition* to *calculus of support and opposition*

Basic research questions

- KR: quest for good combinations of expressiveness, complexity, conceptual simplicity.
- Can Dung AFs be made more expressive?
- Without increasing computational complexity?
- Such that gain in expressiveness outweighs loss of simplicity?
- Provide positive answer to 1, 2; evidence that answer to 3 is positive.
- Initial interest: proof standards; 2 steps: (1) add acceptance conditions, (2) define them in domain independent way.

Abstract Dialectical Framework

=

Dependency Graph + Acceptance Conditions

Basic research questions

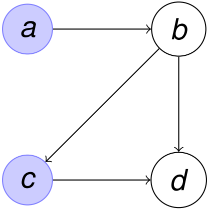
- KR: quest for good combinations of expressiveness, complexity, conceptual simplicity.
- Can Dung AFs be made more expressive?
- Without increasing computational complexity?
- Such that gain in expressiveness outweighs loss of simplicity?
- Provide positive answer to 1, 2; evidence that answer to 3 is positive.
- Initial interest: proof standards; 2 steps: (1) add acceptance conditions, (2) define them in domain independent way.

Abstract Dialectical Framework

=

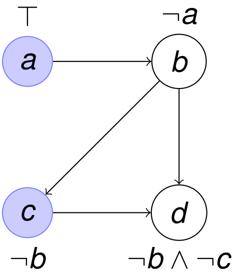
Dependency Graph + Acceptance Conditions

Basic idea



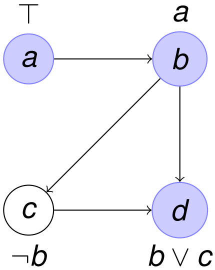
An Argumentation Framework

Basic idea



An Argumentation Framework
with explicit acceptance conditions

Basic idea



A Dialectical Framework
with flexible acceptance conditions

Remark about notation

- Acceptance conditions: Boolean functions
- Take *in*/*out* assignment to parents to generate *in*/*out* assignment of child
- Conveniently represented as propositional formulas
- Sometimes functional notation easier to handle
- Switch between the two, representing assignments by the set of their *in* nodes when using the latter

  so $C_s(R) = in/out$ will mean:
  if R are all the s-parents being *in*, then s is *in*/*out*

2. ADFs: The Formal Framework

- Like Dung, use graph to describe dependencies among nodes.
- Unlike Dung, allow individual acceptance condition for each node.
- Assigns *in* or *out* depending on status of parents.

Definition

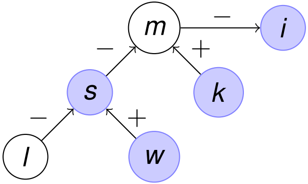
An *abstract dialectical framework* (ADF) is a tuple $D = (S, L, C)$ where

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

Propositional formula representing C_s denoted F_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 and k known (*in*), l not known (*out*)
Other nodes: *in* iff all + parents *in*, all - parents *out*.

Propositionally:

$$w : \top,\ k : \top,\ l : \bot,\ s : w \wedge \neg l,\ m : k \wedge \neg s,\ i : \neg m$$

Dung frameworks: a special case

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

Models

Definition

Let $D = (S, L, C)$ be an ADF. $M \subseteq S$ is a *model* of D if for all $s \in S$: $s \in M$ iff $C_s(M \cap par(s)) = in$.

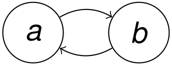
Less formally: if a node is *in* iff its acceptance condition says so.

A first result:
Let $\mathcal{A} = (AR, attacks)$ be an AF, $D_{\mathcal{A}} = (S, L, C)$ its associated dialectical framework. $E \subseteq AR$ stable extension of $\mathcal{A}$ iff E model of $D_{\mathcal{A}}$.

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

Example

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



- For $C_a(\emptyset) = C_b(\emptyset) = \mathit{in}$ and $C_a(\{b\}) = C_b(\{a\}) = \mathit{out}$ (Dung AF): two models, $M_1 = \{a\}$ and $M_2 = \{b\}$.
- For $C_a(\emptyset) = C_b(\emptyset) = \mathit{out}$ and $C_a(\{b\}) = C_b(\{a\}) = \mathit{in}$ (mutual support): $M_3 = \emptyset$ and $M_4 = \{a, b\}$.
- For $C_a(\emptyset) = C_b(\{a\}) = \mathit{out}$ and $C_a(\{b\}) = C_b(\emptyset) = \mathit{in}$ (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 $\{s \equiv F(s) \mid s \in S\}$.

Grounded semantics

Definition

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

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

Γ_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 Γ_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.

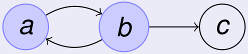
Generalizes grounded semantics, more precisely:
ultimate well-founded semantics by Denecker, Marek, Truszczyński.

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 $S = \{a, b, c\}$, $L = \{(a, b), (b, a), (b, c)\}$:



$C_a(\emptyset) = C_b(\emptyset) = out$ and $C_a(\{b\}) = C_b(\{a\}) = in$ (mutual support),
$C_c(\emptyset) = in$ and $C_c(\{b\}) = out$ (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 $R \subseteq par(s)$, $C_s(R) = in$ and $C_s(R \cup \{r\}) = out$,
2. *attacking*: for no $R \subseteq par(s)$, $C_s(R) = out$ and $C_s(R \cup \{r\}) = in$.

- 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

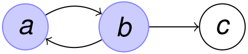
Let $D = (S, L, C)$ be a BADF. A model M of D is a *stable model* if M is the least model of the reduced ADF D^M obtained from D by

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.

Example

- 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.



- Get two models: $\{a, b\}$ and $\{c\}$. Only the latter is expected.
- The reduct of D wrt $\{a, b\}$ is $(\{a, b\}, \{(a, b), (b, a)\}, \{C_a, C_b\})$ 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 $(AR\text{-}R, att \cap (AR\text{-}R \times AR\text{-}R))$.

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.

Preferred semantics, ctd.

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.

Complexity

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. Π_2^P-complete).

Bottom line: no increase in complexity once attacking/supporting links are known.

Relationship to LPs

- Cannot represent LP rules as direct dependencies among atoms:

$$\{c \leftarrow a, not\ b; \quad c \leftarrow not\ a, b\}$$

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

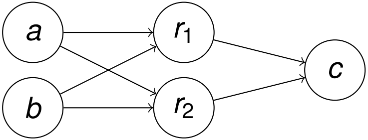
- Resulting ADFs bipolar $\Rightarrow$ any of the defined semantics works.
- Models in one-to-one correspondence (upto rule nodes).
- In principle, “bipolarization" possible for arbitrary ADFs, but exponential blowup - unlike for LP rules.

Relationship to LPs

- Cannot represent LP rules as direct dependencies among atoms:

$$\{c \leftarrow a, not\ b;\ \ c \leftarrow not\ a, b\}$$

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



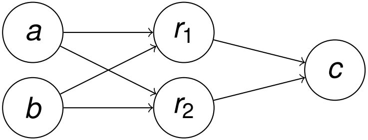
- Resulting ADFs bipolar $\Rightarrow$ any of the defined semantics works.
- Models in one-to-one correspondence (upto rule nodes).
- In principle, “bipolarization" possible for arbitrary ADFs, but exponential blowup - unlike for LP rules.

Relationship to LPs

- Cannot represent LP rules as direct dependencies among atoms:

$$\{c \leftarrow a, not\ b; \quad c \leftarrow not\ a, b\}$$

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



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

3. Weights, preferences and legal proof standards

- 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:
  - $C_s(R) = in$ iff no negative link from elements of R to s,
  - $C_s(R) = in$ iff no negative and at least one positive link from R to s,
  - $C_s(R) = in$ iff more positive than negative links from R to s.
- More fine grained distinctions if V is numerical:
  - $C_s(R) = in$ iff sum of weights of links from R to s positive,
  - $C_s(R) = in$ iff maximal positive weight higher than maximal negative weight,
  - $C_s(R) = in$ iff difference between maximal positive weight and (absolute) maximal negative weight above threshold.

Legal Proof Standards: Farley and Freeman

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.

Farley and Freeman's proof standards

- *Scintilla of Evidence*: at least one weak, defendable argument. $C_s(R) = in$ iff $\exists r \in R : w(r,s) \in \{+v, +s, +c, +w\}$.
- *Dialectical Validity*: at least one credible, defendable argument and the other side's arguments are all defeated: $C_s(R) = in$ iff
  - $\exists r \in R : w(r,s) \in \{+v, +s, +c, \}$ and
  - $w(t,s) \notin \{-v, -s, -c, -w\}$ for all $t \in R$.
- *Preponderance of Evidence*: at least one weak, defendable argument that outweighs the other side's argument: $C_s(R) = in$ iff
  - $\exists r \in R : w(r,s) \in \{+v, +s, +c, +w\}$ and
  - $\neg\exists r \in R : w(r,s) = -v$ and
  - $\exists r \in R : w(r,s) = -s$ implies $\exists r' \in R : w(r',s) = +v$ and
  - $\exists r \in R : w(r,s) = -c$ implies $\exists r' \in R : w(r',s) \in \{+v, +s\}$ and
  - $\exists r \in R : w(r,s) = -w$ implies $\exists r' \in R : w(r',s) \in \{+v, +s, +c\}$.

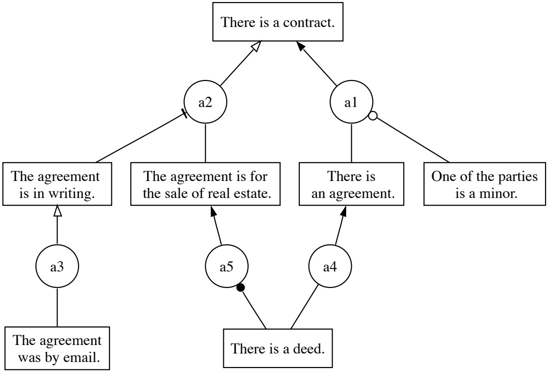
etc.

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.

4. An Application: Reconstructing Carneades

- Advanced model of argumentation (Gordon, Prakken, Walton 07)
- Proof standards: scintilla of evid., preponderance of evid., clear and convincing evid., beyond reas. doubt and dial. validity
- Some paraconsistency at work
- Major restriction: no cycles (case where Dung semantics coincide)



Carneades: Basic Definitions

- An **argument** is a tuple $\langle P, E, c\rangle$ with premises P, exceptions E ($P \cap E = \emptyset$) and conclusion c. c and elements of P, E are literals.
- An **argument evaluation structure** (CAES) is a tuple $\mathcal{S} = \langle \mathit{args}, \mathit{ass}, \mathit{weight}, \mathit{standard}\rangle$, where
  - *args* is an acyclic set of arguments,
  - *ass* is a consistent set of literals,
  - *weight* assigns a real number to each argument, and
  - *standard* maps propositions to a proof standard.
- $\langle P, E, c\rangle \in \mathit{args}$ is **applicable** in $\mathcal{S}$ iff
  - $p \in P$ implies $p \in \mathit{ass}$ or [$\overline{p} \notin \mathit{ass}$ and p acceptable in $\mathcal{S}$], and
  - $p \in E$ implies $p \notin \mathit{ass}$ and [$\overline{p} \in \mathit{ass}$ or p is not acceptable in $\mathcal{S}$].

Carneades: Acceptability

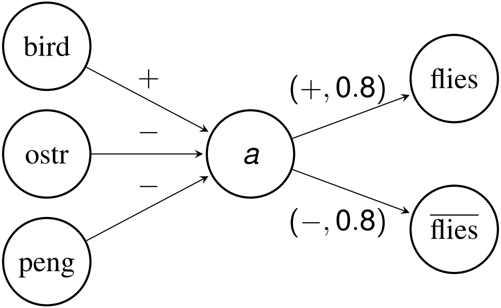
A proposition *p* is **acceptable** in $\mathcal{S}$ iff:

- *standard*(*p*) $=$ *se* and there is an applicable argument for *p*,
- *standard*(*p*) $=$ *pe*, *p* satisfies *se*, and max weight assigned to applicable argument pro *p* greater than the max weight of applicable argument con *p*,
- *standard*(*p*) $=$ *ce*, *p* satisfies *pe*, and max weight of applicable pro argument exceeds threshold α, and difference between max weight of applicable pro arguments and max weight of applicable con arguments exceeds threshold β,
- *standard*(*p*) $=$ *bd*, *p* satisfies *ce*, and max weight of the applicable con arguments less than threshold γ,
- *standard*(*p*) $=$ *dv*, and there is an applicable argument pro *p* and no applicable argument con *p*.

Translation

Example:

$a = \langle\{bird\}, \{peng, ostr\}, flies\rangle$ with $weight(a) = 0.8$ translates to:



Translation II

Acceptance condition for **argument** nodes: $C_n(R) = in$ iff

(1) for all p_i with $w(p_i, a) = +$, $p_i \in ass$ or $[\overline{p}_i \notin ass$ and $p_i \in R]$, and
(2) for all e_i with $w(e_i, a) = -$, $p_i \notin ass$ and $[p_i \notin R$ or $\overline{p}_i \in ass]$.

Acceptance conditions for **proposition** nodes: $C_m(R) = in$ iff

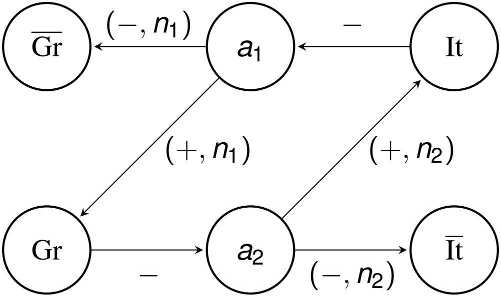
$s = se$: [1] for some $r \in R$, $w(r, m) = (+, n)$
$s = pe$: [1] and
[2] $max\{n \mid t \in R, w(t, m) = (+, n)\} > max\{n \mid t \in R, w(t, m) = (-, n)\}$
$s = ce$: [1] and [2] and
[3] $max\{n \mid t \in R, w(t, m) = (+, n)\} > \alpha$ and
[4] $max\{n \mid t \in R, w(t, m) = (+, n)\} -$
$max\{n \mid t \in R, w(t, m) = (-, n)\} > \beta$.
etc.

Theorem: arg node in iff argument applicable; prop node in iff proposition acceptable (independently of chosen semantics)

Why a Reconstruction?

- shows generality of ADFs: Dung and Carneades special cases
- puts Carneades on safe formal ground
- allows us to lift restriction of Carneades to acyclic graphs

$$a_1 = \langle \emptyset, \{It\}, Gr \rangle, a_2 = \langle \emptyset, \{Gr\}, It \rangle.$$



5. 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.
- Reconstructed Carneades, thus lifting its acyclicity restriction.

5. 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.
- Reconstructed Carneades, thus lifting its acyclicity restriction.

5. 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.
- Reconstructed Carneades, thus lifting its acyclicity restriction.

5. 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.
- Reconstructed Carneades, thus lifting its acyclicity restriction.

5. 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.
- Reconstructed Carneades, thus lifting its acyclicity restriction.

5. 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.
- Reconstructed Carneades, thus lifting its acyclicity restriction.

5. 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.
- Reconstructed Carneades, thus lifting its acyclicity restriction.

5. 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.
- Reconstructed Carneades, thus lifting its acyclicity restriction.

How about flattening?

- Can ADFs be compiled to AFs such that the semantics coincide (for old nodes)?
- Yes, in principle.
  - For node n introduce nodes representing subsets of its parents.
  - Whenever $C_n(R) = out$ add attack (R, n).
  - For parent s add node s_{out} and attack (s, s_{out}).
  - Add attacks (s, R) whenever $s \notin R$ and (s_{out}, R) whenever $s \in R$.
- Exponential blowup, thus not really interesting.
- Can we do any better? Open question.

How about flattening?

- Can ADFs be compiled to AFs such that the semantics coincide (for old nodes)?
- Yes, in principle.
  - For node n introduce nodes representing subsets of its parents.
  - Whenever $C_n(R) = out$ add attack (R, n).
  - For parent s add node s_{out} and attack (s, s_{out}).
  - Add attacks (s, R) whenever $s \notin R$ and (s_{out}, R) whenever $s \in R$.
- Exponential blowup, thus not really interesting.
- Can we do any better? Open question.

How about flattening?

- Can ADFs be compiled to AFs such that the semantics coincide (for old nodes)?
- Yes, in principle.
  - For node n introduce nodes representing subsets of its parents.
  - Whenever $C_n(R) = out$ add attack (R, n).
  - For parent s add node s_{out} and attack (s, s_{out}).
  - Add attacks (s, R) whenever $s \notin R$ and (s_{out}, R) whenever $s \in R$.
- Exponential blowup, thus not really interesting.
- Can we do any better? Open question.

How about flattening?

- Can ADFs be compiled to AFs such that the semantics coincide (for old nodes)?
- Yes, in principle.
  - For node n introduce nodes representing subsets of its parents.
  - Whenever $C_n(R) = out$ add attack (R, n).
  - For parent s add node s_{out} and attack (s, s_{out}).
  - Add attacks (s, R) whenever $s \notin R$ and (s_{out}, R) whenever $s \in R$.
- Exponential blowup, thus not really interesting.
- Can we do any better? Open question.

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!