TECHNISCHE UNIVERSITÄT DRESDEN

Artificial Intelligence, Computational Logic

ABSTRACT ARGUMENTATION

Generalizations of Argumentation Frameworks

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

Sarah Gaggl

Dresden, ICCL Summer School 2017

DRESDEN concept Exzellenz aus Wissenschaft und Kultur

Outline

Decision Problems on AFs

Credulous Acceptance

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

Skeptical Acceptance

Skept_σ: Given AF $F = (A, R)$ and $a \in A$; is a contained in every σ-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.

Decision Problems on AFs

Credulous Acceptance

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

Skeptical Acceptance

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

[1] This is only relevant for stable semantics.

Further Decision Problems

Verifying an extension

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

Further Decision Problems

Verifying an extension

Ver_σ: Given AF $F = (A, R)$ and $S \subseteq A$; is S a σ-extension of F?

Does there exist an extension?

Exists_σ: Given AF $F = (A, R)$; Does there exist a σ-extension for F?

Further Decision Problems

Verifying an extension

Ver_σ: Given AF $F = (A, R)$ and $S \subseteq A$; is S a σ-extension of F?

Does there exist an extension?

Exists_σ: Given AF $F = (A, R)$; Does there exist a σ-extension for F?

Does there exist a nonempty extensions?

$\text{Exists}_\sigma^{\neg\emptyset}$: Does there exist a non-empty σ-extension for F?

Complexity Results (Summary)

Complexity of decision problems in AFs.

σ	Cred_σ	Skept_σ	Ver_σ	Exists_σ	$\text{Exists}_\sigma^{\neg\emptyset}$
cf	in L	trivial	in L	trivial	in L
naive	in L	in L	in L	trivial	in L
ground	P-c	P-c	P-c	trivial	P-c
stable	NP-c	co-NP-c	in L	NP-c	NP-c
adm	NP-c	trivial	in L	trivial	in L
comp	NP-c	P-c	in L	trivial	NP-c
cf2	NP-c	co-NP-c	in P	trivial	in L
ideal	Θ_2^P-c	Θ_2^P-c	Θ_2^P-c	trivial	Θ_2^P-c
pref	NP-c	Π_2^P-c	co-NP-c	trivial	NP-c
semi	Σ_2^P-c	Π_2^P-c	co-NP-c	trivial	NP-c
stage	Σ_2^P-c	Π_2^P-c	co-NP-c	trivial	in L

see [Baroni et al.2011, Coste-Marquis et al.2005, Dimopoulos and Torres1996, Dung1995, Dunne2008, Dunne and Bench-Capon2002, Dunne and Bench-Capon2004, Dunne and Caminada2008, Dvořák et al.2011, Dvořák and Woltran2010a, Dvořák and Woltran2010b]

Intractable problems in Abstract Argumentation

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 Φ
- Give a reduction that maps Φ to an Argumentation Framework F_Φ containing an argument Φ.
- Show that Φ is satisfiable iff the argument Φ is accepted.

Canonical Reduction

Definition

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

$$
\begin{aligned}
A_\Phi &= Z \cup \bar{Z} \cup \{C_1, \ldots, C_m\} \cup \{\Phi\} \\
R_\Phi &= \{(z, \bar{z}), (\bar{z}, z) \mid z \in Z\} \cup \{(C_i, \Phi) \mid i \in \{1, \ldots, m\}\} \cup \\
&\quad \{(z, C_i) \mid i \in \{1, \ldots, m\}, z \in \{l_{i1}, l_{i2}, l_{i3}\}\} \cup \\
&\quad \{(\bar{z}, C_i) \mid i \in \{1, \ldots, m\}, \neg z \in \{l_{i1}, l_{i2}, l_{i3}\}\}
\end{aligned}
$$

Canonical Reduction

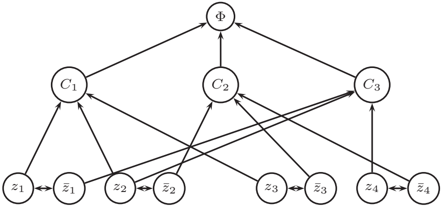
Definition

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

$$\begin{aligned} A_\Phi &= Z \cup \bar{Z} \cup \{C_1, \ldots, C_m\} \cup \{\Phi\} \\ R_\Phi &= \{(z, \bar{z}), (\bar{z}, z) \mid z \in Z\} \cup \{(C_i, \Phi) \mid i \in \{1, \ldots, m\}\} \cup \\ &\quad \{(z, C_i) \mid i \in \{1, \ldots, m\}, z \in \{l_{i1}, l_{i2}, l_{i3}\}\} \cup \\ &\quad \{(\bar{z}, C_i) \mid i \in \{1, \ldots, m\}, \neg z \in \{l_{i1}, l_{i2}, l_{i3}\}\} \end{aligned}$$

Example

Let $\Phi = (z_1 \vee z_2 \vee z_3) \wedge (\neg z_2 \vee \neg z_3 \vee \neg z_4) \wedge (\neg z_1 \vee z_2 \vee z_4)$.



Canonical Reduction: CNF $\Rightarrow$ AF (ctd.)

Theorem

The following statements are equivalent:

1. Φ is satisfiable
2. F_Φ has an admissible set containing Φ
3. F_Φ has a complete extension containing Φ
4. F_Φ has a preferred extension containing Φ
5. F_Φ has a stable extension containing Φ

Complexity Results

Theorem

1. Cred_{stable} is NP-complete
2. Cred_{adm} is NP-complete
3. Cred_{comp} is NP-complete
4. Cred_{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. □

Outline

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

ADFs

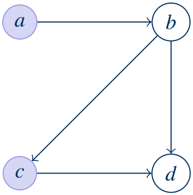
Basic Idea

Abstract Dialectical Framework

=

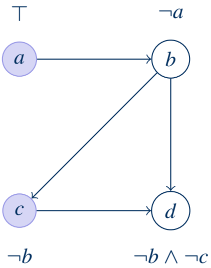
Dependency Graph + Acceptance Conditions

ADFs - Basic idea



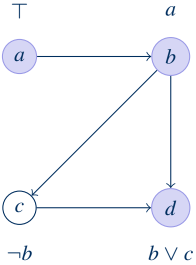
An Argumentation Framework

ADFs - Basic idea (ctd.)



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 $\mathbf{t}$(rue) or $\mathbf{f}$(alse) depending on status of parents.

ADF [Brewka and Woltran 2010]

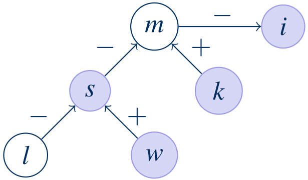
An abstract dialectical framework (ADF) is a tuple $D = (S, L, C)$ where

- 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 $C_s : 2^{par(s)} \to \{\mathbf{t}, \mathbf{f}\}$, 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:

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

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:
  $F_s = \neg r_1 \wedge \ldots \wedge \neg r_n$, 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 $v : S \to \{\mathbf{t}, \mathbf{f}\}$

Definition

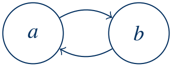
Let $D = (S, C)$ be an ADF. An interpretation v is a model of D if for all $s \in S$: $v(s) = v(C_s)$.

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

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

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.

	a	b
v_1	**t**	**f**
v_2	**f**	**t**
v_3	**f**	**f**
v_4	**t**	**t**

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: *in* or *out*
- Equivalently one can use labelings [5], which assign three values: *in* ($\mathbf{t}$), *out* ($\mathbf{f}$) and *undecided* ($\mathbf{u}$)

Definition

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

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

Example Labeling

Example

Given the following AF

Then its complete labelings are given by

	a	b	c
$\mathcal{L}_1$	**u**	**u**	**u**
$\mathcal{L}_2$	**t**	**f**	**u**
$\mathcal{L}_3$	**f**	**t**	**u**

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?

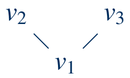
Information Ordering

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

Information Ordering

Example

	a	b	c
v_1	$\mathbf{u}$	$\mathbf{u}$	$\mathbf{u}$
v_2	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{u}$
v_3	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{u}$



A Characteristic Function for ADFs

- Our goal: define a characteristic function for ADFs [7] like for AFs
- Intuitively, $\mathbf{u}$ 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

	a
v_1	$\mathbf{u}$
v_2	$\mathbf{t}$
v_3	$\mathbf{f}$

$[v_1]_2 = \{v_2, v_3\}$, $[v_2]_2 = v_2$ and $[v_3]_2 = v_3$

A Characteristic Function for ADFs (contd)

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

Information Ordering

A Characteristic Function for ADFs (contd)

- 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

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

Example

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

	a
v	$\mathbf{u}$
v_2	$\mathbf{t}$
v_3	$\mathbf{f}$

$v_2(C_a) = \mathbf{f}$ $\quad$ $v_3(C_a) = \mathbf{t}$

$\mathbf{u}$

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

A Characteristic Function for ADFs (contd)

Example

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

	a
v	$\mathbf{u}$
v_2	$\mathbf{t}$
v_3	$\mathbf{f}$

$$v_2(C_a) = \mathbf{t} = v_3(C_a)$$

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

A Characteristic Function for ADFs (contd)

Example

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

	a	b
v	$\mathbf{t}$	$\mathbf{u}$
v_2	$\mathbf{t}$	$\mathbf{t}$
v_3	$\mathbf{t}$	$\mathbf{f}$

$$v_2(C_a) = \mathbf{t} = v_3(C_a)$$

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

- Here v incorporates already information: $v(a) = \mathbf{t}$

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 Γ_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 Γ_D wrt $<_i$
- v is complete in D if $v = \Gamma_D(v)$
- v is preferred in D if v is $<_i$-maximal admissible

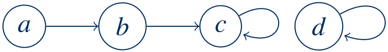
Remember for AFs we have:

- admissible if $E \subseteq \mathcal{F}_F(E)$
- grounded if E is lfp of $\mathcal{F}_F$
- complete if $E = \mathcal{F}_F(E)$
- preferred if E is $\subseteq$-maximal admissible

Example

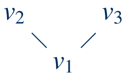
Example

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



Then the complete models are given by:

	a	b	c	d	
v_1	**t**	**t**	**u**	**u**	grd, com
v_2	**t**	**t**	**t**	**u**	com, prf
v_3	**t**	**t**	**f**	**u**	com, prf

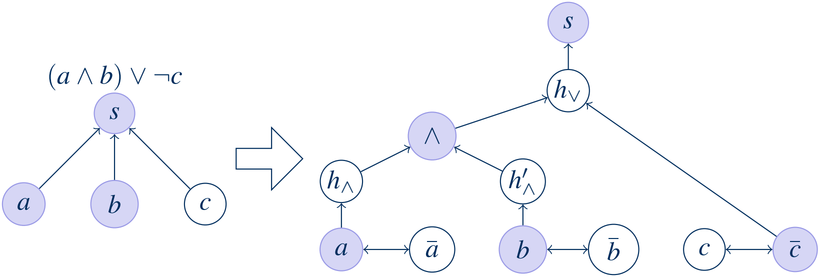


Remarks about Expressibility

- Acceptance conditions of ADFs also allow definitions of preference relations
- Argument A has a higher priority than B: $C_B = \varphi \wedge (B \rightarrow A)$
- 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:



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

Outline

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:
  - $C_s(R) = in$ 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.

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

Computational Problems

Credulous Acceptance

Cred_σ: Given ADF $D = (S, L, C)$ and $a \in S$; is there an interpretation $I \in \sigma(D)$ with $I(a) = \mathbf{t}$?

Skeptical Acceptance

Skept_σ: Given ADF $D = (S, L, C)$ and $a \in S$; is $I(a) = \mathbf{t}$ for each interpretation $I \in \sigma(D)$?

Further Computational Problems

Verification of an interpretation

Ver_σ: Given ADF $D = (S, L, C)$ and an interpretation I; is $I \in \sigma(D)$?

Existence of an interpretation

Exists_σ: Given ADF $D = (S, L, C)$; is $\sigma(D) \neq \emptyset$?

Existence of a nonempty interpretation

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

Complexity Results (Summary)

Complexity of ADFs

σ	Cred_σ	Skept_σ	Ver_σ	Exists_σ	$\text{Exists}_\sigma^{\neg\emptyset}$
ground	co-NP-c	co-NP-c	DP-c	trivial	co-NP-c
model	NP-c	co-NP-c	in P	NP-c	NP-c
adm	Σ_2^P-c	trivial	co-NP-c	trivial	Σ_2^P-c
comp	Σ_2^P-c	co-NP-c	DP-c	trivial	Σ_2^P-c
pref	Σ_2^P-c	Π_3^P-c	Π_2^P-c	trivial	Σ_2^P-c

[1] Leila Amgoud and Claudette Cayrol and Marie-Christine Lagasquie and Pierre Livet,
On Bipolarity in Argumentation Frameworks
International Journal of Intelligent Systems 23(10): 1062–1093 (2008)

[2] P. Baroni, F. Cerutti, M. Giacomin and G. Guida.
AFRA: Argumentation Framework with Recursive Attacks.
Int. J. Approx. Reasoning 52(1): 19–37 (2011).

[3] G. Brewka and S. Woltran.
Abstract Dialectical Frameworks.
Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR'10), pp. 102–111, AAAI Press, 2010.

[4] G. Brewka, P. Dunne and S. Woltran.
Relating the Semantics of Abstract Dialectical Frameworks and Standard AFs.
Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11), 2011.

[5] M. Caminada and D. M. Gabbay.
A logical account of formal argumentation.
Studia Logica 93(2): 109–145 (2009).

[6] S. Modgil.
Reasoning about Preferences in Argumentation Frameworks.
Artif. Intell. 173(9-10): 910–934 (2009).

[7] H. Strass.
Approximating Operators and Semantics for Abstract Dialectical Frameworks.
Technical Report 1, Institute of Computer Science, Leipzig University, January 2013.