Abstract Dialectical Frameworks

Gerhard Brewka
University of Leipzig
Dept. of Computer Science
Augustusplatz 10-11, D-04109 Leipzig
brewka@informatik.uni-leipzig.de

Stefan Woltran
Vienna University of Technology
Institute of Information Systems
Favoritenstraße 9-11, A-1040 Vienna, Austria
woltran@dbai.tuwien.ac.at

Abstract

In this paper we introduce dialectical frameworks, a powerful generalization of Dung-style argumentation frameworks where each node comes with an associated acceptance condition. This allows us to model different types of dependencies, e.g. support and attack, as well as different types of nodes within a single framework. We show that Dung's standard semantics can be generalized to dialectical frameworks, in case of stable and preferred semantics to a slightly restricted class which we call bipolar frameworks. We show how acceptance conditions can be conveniently represented using weights respectively priorities on the links and demonstrate how some of the legal proof standards can be modeled based on this idea.

Motivation

Formal models of argumentation have recently received considerable interest across different AI communities, like nonmonotonic reasoning, multi-agent systems and legal reasoning. Argumentation frameworks provide a particular way of defining nonmonotonic consequences by constructing and comparing the arguments pro and con a certain position. Particularly successful and widely used are the abstract argumentation frameworks introduced by Dung (1995). These frameworks basically are graphs whose nodes represent abstract arguments. The content of these arguments is not further analyzed. The links in the graphs represent attack relations. Intuitively, an argument is accepted unless it is attacked by another accepted argument.

Dung defined several semantics formalizing different intuitions about which arguments to accept on the basis of a given framework, most notably the *grounded*, *preferred* and *stable* semantics.

Dung frameworks are very useful as analytical tools for comparing various forms of argumentation. They are also commonly used as target systems for translations from less abstract formalisms. These formalisms then inherit, via the translation to Dung frameworks, a semantics. A number of researchers in argumentation, like Wyner, Bench-Capon, and Dunne (2009) or Prakken (2009) followed this approach and built specific argumentation systems based on some kind of defeasible and undefeasible rules. Rather than directly defining a semantics for their systems, they translated them to Dung frameworks.

In spite of their success, it is obvious that Dung frameworks lack certain features which are common in almost every form of argumentation to be found in practice. One such feature we focus on in this paper are proof standards. The legal reasoning literature is full of discussions of standards like *scintilla of evidence*, *preponderance of evidence*, *beyond reasonable doubt* etc. Also in everyday reasoning proof standards play an essential role: in situations involving risk we obviously apply higher standards than in cases where there is not much to loose. Excellent overviews of some recent formal treatments can be found in (Atkinson and Bench-Capon 2007) and (Gordon and Walton 2009).

The work presented here started as an attempt to add proof standards to Dung frameworks. Proof standards in legal reasoning are domain independent. They are defined based on certain domain independent properties of the arguments pro and con a certain position. In this paper we will first introduce the more abstract notion of an acceptance condition and add it to Dung frameworks. Acceptance conditions cover any function determining the status of a node based on the status of its parent nodes. This includes domain dependent conditions and thus goes beyond the handful of well-known legal standards. The latter will then be introduced based on certain properties/types of the links in our graphs. Acceptance conditions allow us to introduce different node and link types. The influence a node may have on another node is entirely specified through the acceptance condition. So, in contrast to Dung frameworks where links represent one particular type of relationship, namely attack, and where nodes are accepted unless attacked, our frameworks allow a variety of different dependencies to be represented in a flexible way[1]: we can have nodes which are not accepted unless supported, and corresponding supporting links.

[1] For a node with n incoming links the number of possible acceptance conditions will be 2^{2^n}.

We also can have links of different strength, and even links which support/attack a node sometimes, that is, depending on the context. The following slogan characterizes our approach:

abstract dialectical frameworks =
dependency graphs + acceptance conditions

An important aspect of the work we are going to present here is the generalization of the standard Dung semantics. It turns out that grounded semantics can indeed be generalized to arbitrary dialectical frameworks. For stable and preferred semantics a notion of support and attack is needed which is present in a somewhat restricted, but still powerful and flexible class of frameworks which we call bipolar.

Although we started from argumentation frameworks, we decided to change terminology. In Dung frameworks an argument is accepted unless it is attacked by an accepted node. As mentioned above, in our frameworks it will depend entirely on the acceptance condition whether a node is accepted or not. We still can have nodes which behave like arguments – Dung frameworks, after all, are special cases of our frameworks – but we also may have nodes which are rejected unless they are supported by some accepted node.

To reflect this higher generality we chose the term *dialectical frameworks*. In classical philosophy, dialectic is a form of reasoning based on the exchange of arguments and counter-arguments, advocating propositions (theses) and counter-propositions (antitheses). This describes nicely what our frameworks can be used for. Also, following the Carneades terminology (Gordon, Prakken, and Walton 2007) we call the nodes in our graphs statements (or positions) rather than arguments.

The outline of the paper is as follows. We first give the relevant background on Dung systems. We then go on to present the general approach and define well-founded (alias grounded) semantics for it. The subsequent section introduces bipolar frameworks, that is, frameworks whose links either support or attack a node, and presents stable and preferred semantics for these. We then show how acceptance conditions can conveniently be represented using weights. We apply this idea and formalize several legal proof standards which go back to the work by Farley and Freeman (1995) and are well known in legal reasoning. We finally discuss proof standards based on qualitative preferences among links and the complexity of major reasoning tasks. We conclude with a discussion of related work which in particular includes bipolar argumentation frameworks (Cayrol and Lagasquie-Schiex 2009).

Background

We assume some familiarity with Dung-style abstract argumentation (Dung 1995) and just recall the essential definitions. An argumentation framework (AF, for short) is a pair $\mathcal{A} = (AR, attacks)$ where AR is a set of arguments, and *attacks* is a binary relation on AR (used in infix in prose). An argument $a \in AR$ is *acceptable with respect to a set S of arguments*, if each argument $b \in AR$ that attacks a is attacked by some $b' \in S$. A set S of arguments is *conflict-free*, if there are no arguments $a, b \in S$ such that a attacks b, and S is *admissible*, if in addition each argument in S is acceptable wrt. S.

Dung defined (among others) the following three semantics for an AF $\mathcal{A} = (AR, attacks)$:

- The *grounded extension of* $\mathcal{A}$ is the least fixpoint of the operator $F_\mathcal{A} : 2^{AR} \rightarrow 2^{AR}$ where
$$F_\mathcal{A}(S) = \{a \in AR \mid a \text{ is acceptable wrt. } S\}.$$
- A *preferred extension* of $\mathcal{A}$ is a maximal (wrt. $\subseteq$) admissible set of $\mathcal{A}$.
- A *stable extension of* $\mathcal{A}$ is a conflict-free set of arguments S which attacks each argument not belonging to S.

The unique grounded extension is a subset of the intersection of all preferred extensions, and each stable extension is a preferred extension, but not vice versa. While the grounded and some preferred extension are guaranteed to exist (the latter in the finite case), $\mathcal{A}$ may have no stable extension.

The general framework

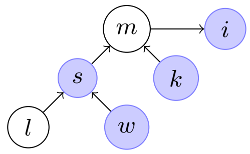
We now define abstract dialectical frameworks (ADFs) formally. An ADF is a directed graph whose nodes represent statements or positions which can be accepted or not. The links represent dependencies: the status of a node s only depends on the status of its parents (denoted $par(s)$), that is, the nodes with a direct link to s. In addition, each node s has an associated acceptance condition C_s specifying the exact conditions under which s is accepted. C_s will be a function assigning to each subset of $par(s)$ one of the values *in*, *out*. Intuitively, if for some $R \subseteq par(s)$ we have $C_s(R) = in$, then s will be accepted provided the nodes in R are accepted and those in $par(s) \setminus R$ are not accepted.

Definition 1 *An* abstract dialectical framework *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.*

Example 1 *This is a variant of a similar example by Gordon, Prakken, and Walton (2007). A person is innocent, unless she is a murderer. A killer is a murderer, unless she acted in self-defense. There must be evidence for self-defense, for instance a witness who is not known to be a liar.*

The dependency structure of the example can be represented as follows:



Let's assume w and k are known and l is not known, that is C_w and C_k are (constant functions) in, C_l is out. The acceptance conditions for the remaining nodes are: $C_s(R) = in$ iff $w \in R$ and $l \notin R$; $C_m(R) = in$ iff $k \in R$ and $s \notin R$; $C_i(R) = in$ iff $R = \emptyset$. The shaded nodes represent the nodes which are in when values are propagated according to the chosen acceptance conditions.

Let us first check how Dung argumentation frameworks fit into the picture. AFs have attacking links only and a single type of nodes. This can easily be captured in an ADF. Let $\mathcal{A} = (AR, attacks)$ be an argumentation framework. The associated dialectical framework $D_\mathcal{A} = (AR, attacks, C)$ uses, for all nodes $s \in AR$, the following acceptance condition: $C_s(R) = in$ iff $R = \emptyset$, $C_s(R) = out$ otherwise.

Interestingly, we can also represent a normal logic program G as an associated ADF D_G. The nodes of D_G are the atoms in the program. There is a link from atom a to atom b whenever b appears, positive or negated, in the body of some rule with head a. Finally, for an atom a, C_a is defined as follows (for $R \subseteq par(a)$): $C_a(R) = in$ if there exists a rule

$$a \leftarrow b_1, \ldots, b_k, not\ c_1 \ldots, not\ c_m$$

in G, such that $\{b_1, \ldots, b_k\} \subseteq R$ and $\{c_1 \ldots, c_m\} \cap R = \emptyset$; and $C_a(R) = out$, otherwise.

The reader will have noticed that our acceptance conditions are boolean functions (we chose *in* and *out* as values rather than *true* and *false* because we make no claims about the truth of the involved statements[2]). Boolean functions can conveniently be represented using propositional formulas. An ADF can thus also be viewed as a graph such that each node s is annotated with a propositional formula $F(s)$ built from parent nodes of s. For instance, the formulas representing the acceptance condition of a node s in a Dung-style ADF are of the form $\neg r_1 \wedge \ldots \wedge \neg r_n$, where $r_1, \ldots, r_n$ are all attackers of s. Let us mention at this point, that the acceptance conditions are not intended to tell us anything about the contents of the statements[3] which remain fully abstract. Instead, the acceptance conditions just explain the relationship between statements.

We now turn to the semantics of dialectical frameworks. We start with the notion of a model. Intuitively, a model M is a set of statements satisfying the acceptance condition of each node:

[2]Three-valued labellings with *in*, *out* and *undec*, the latter representing undecided, have been used to compute extensions of argumentation frameworks, see (Modgil and Caminada 2009) for an overview. Although we use two of these labels, we use them for entirely different purposes, namely to define the notion of a model.

[3]For recent work in this direction, see e.g. (Amgoud and Besnard 2009; Wooldridge, Dunne, and Parsons 2006).

Definition 2 *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 $C_s(M \cap par(s)) = in$. Moreover, $M \subseteq S$ is a* model *of D if M is conflict-free and for each $s \in S$, $C_s(M \cap par(s)) = in$ 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 $C_s(M \cap par(s)) = in$.

We say M is a minimal model if there is no model M' which is a proper subset of M.

Definition 3 *Let $D = (S, L, C)$ be an ADF. $s \in S$ is a (minimal) consequence of D iff s is contained in all (minimal) models of D.*

We will denote the set of consequences of D as $Cn(D)$.

Example 2 *Consider the ADF $D = (S, L, C)$ with $S = \{a, b\}$ and $L = \{(a, b), (b, a)\}$. If the acceptance conditions are $C_a(\emptyset) = C_b(\emptyset) = in$ and $C_a(\{b\}) = C_b(\{a\}) = out$ (that is, we have a Dung framework), then we have two models, $M_1 = \{a\}$ and $M_2 = \{b\}$.*

For $C_a(\emptyset) = C_b(\emptyset) = out$ and $C_a(\{b\}) = C_b(\{a\}) = in$ (that is, a and b support each other), we get the two models, $M_3 = \emptyset$ and $M_4 = \{a, b\}$. Only the former is minimal.

For $C_a(\emptyset) = C_b(\{a\}) = out$ and $C_a(\{b\}) = C_b(\emptyset) = in$ (that is, a attacks b, b is a necessary support for a), we get no model at all. Assume a is in, then b is out, so a has no support and must be out. Assume to the contrary that a is out. Then b is in, so a has support and must be in. In both cases we do not have a model.

It is not difficult to verify that, when the acceptance condition of each node s is represented as a propositional formula $F(s)$, a model is just a propositional model of the set of formulas

$$\{s \equiv F(s) \mid s \in S\}. \quad (1)$$

For ADFs corresponding to Dung argumentation frameworks we obtain the following relationships:

Proposition 1 *Let $\mathcal{A} = (AR, attacks)$ be an argumentation framework, $D_\mathcal{A} = (AR, attacks, C)$ its associated dialectical framework, and $E \subseteq AR$. (1) E is a conflict-free set in $\mathcal{A}$ iff E is conflict-free in $D_\mathcal{A}$; (2) E is a stable extension of $\mathcal{A}$ iff E is a model of $D_\mathcal{A}$.*

Proof: (1) E is conflict-free in $\mathcal{A}$ iff for no $s, s' \in E$, s' attacks s iff $E \cap par(s) = \emptyset$ for all $s \in E$ iff $C_s(E \cap par(s)) = in$ for all $s \in E$. (2) E is a stable extension of $\mathcal{A}$ iff E is conflict free in $\mathcal{A}$ and defeats each $s \in AR \setminus E$ iff E is conflict free in $D_\mathcal{A}$ and for each $s \in AR \setminus E$, $C_s(E \cap par(s)) \neq in$ iff E is a model of $D_\mathcal{A}$. □

For more general ADFs, models and stable models will be different. This is the topic of the next section.

Here we first want to introduce a generalization of grounded semantics. Following common terminology in logic programming we will speak of the well-founded model rather than the grounded extension.

Definition 4 *Let $D = (S, L, C)$ be an ADF. Consider the operator*

$$\Gamma_D(A, R) = (acc(A, R), reb(A, R))$$

where $acc(A, R) =$

$$\{r \in S \mid A \subseteq S' \subseteq (S \setminus R) \Rightarrow C_r(S' \cap par(r)) = in\}$$

and $reb(A, R) =$

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

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

The intuition behind this operator is as follows: in the first argument we collect those statements which are already known to be *in*, in the second those which are known to be *out*. The least fixpoint can be reached by iterating Γ_D, starting with $(\emptyset, \emptyset)$. In each iteration, statement r is added to the first (respectively second) argument, if whatever the status of the still undecided statements is (that is, those neither in the first nor in the second argument), the status of r must be *in* (respectively *out*).

We first show that the well-founded model of an ADF is contained in each of its models.

Proposition 2 *Let* D *be an ADF,* W *the well-founded model of* D *and* M *an arbitrary model of* D*. Then* $W \subseteq M$*.*

Proof: We can show by induction on the number i of iterations of Γ_D that, for all i, $\Gamma^i_D(\emptyset, \emptyset) = (A, R)$ implies $A \subseteq M$ and $R \cap M = \emptyset$ for each model M of D. This follows directly from the construction of $acc(A, R)$, respectively $reb(A, R)$. □

We next show that the well-founded model generalizes grounded semantics adequately.

Proposition 3 *Let* $\mathcal{A}$ *be an argumentation framework,* $D_{\mathcal{A}}$ *its associated ADF. The grounded extension of* $\mathcal{A}$ *coincides with the well-founded model of* $D_{\mathcal{A}}$*.*

Proof: We can show by induction that, for all $i > 0$, $\Gamma^i_{D_{\mathcal{A}}}(\emptyset, \emptyset) = (F^i_{\mathcal{A}}(\emptyset), R_i)$, where $F_{\mathcal{A}}$ is the fixpoint operator used in the definition of the grounded extension (see Background section), and R_i is the set of arguments of $\mathcal{A}$ defeated by $F^{i-1}_{\mathcal{A}}(\emptyset)$. From this the result follows. □

We finally establish the relationship between logic programs under the well-founded semantics (Gelder, Ross, and Schlipf 1991) and their associated ADFs.

Proposition 4 *Let* G *be a normal logic program,* $D_G = (S, L, C)$ *its associated ADF. The well-founded model of* G *is a subset of the well-founded model of* D_G*.*

Proof: The well-founded model of G can be characterized as the least fixpoint of the monotone operator γ^2_G obtained by applying the antimonotone operator γ_G twice. Here $\gamma_G(S) = Cn(G^S)$ is the set of consequences of the Gelfond/Lifschitz reduct G^S. The reduct is obtained from G by deleting rules with a literal *not* a in the body, for some $a \in S$, and by deleting *not* literals from the remaining rules. Applying γ_G once to a set of atoms already known to be in the well-founded model yields a set of potential conclusions, that is, atoms which are still possibly in this model (the complement of this set cannot be in the model). Applying γ_G to a set of potential conclusions yields a set of atoms which must be in the well-founded model.

The twofold application of γ_G is mirrored in a single application of Γ_{D_G} where the complement of the result of the first application of γ_G is kept in the second argument. We can show that, for $i > 0$, $\Gamma^i_{D_G}(\emptyset, \emptyset) = (A_i, R_i)$ implies $\gamma^{2i-2}_G(\emptyset) \subseteq A_i$ and $S \setminus \gamma^{2i-1}_G(\emptyset) \subseteq R_i$. From this the result follows. □

To see that the well-founded model of D_G may contain more elements than the well-founded model of G consider the program G consisting of the four rules

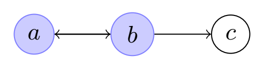
$$a \leftarrow not\ b;\ a \leftarrow b;\ b \leftarrow not\ a;\ b \leftarrow a$$

The well-founded model of G is empty, but the well-founded model of D_G is $\{a, b\}$.[4]

Stable and preferred models for BADFs

Stable models in logic programming (Gelfond and Lifschitz 1988) exclude *self-supporting cycles* - which can be viewed as their distinguishing feature. Such cycles may appear in models of ADFs and cannot be captured by minimality alone. An example is the following:

Example 3 *Consider the following ADF* $D = (S, L, C)$ *with* $S = \{a, b, c\}$*,* $L = \{(a, b), (b, a), (b, c)\}$*, i.e. we have the following structure*



and for the acceptance conditions, we want (a, b) *and* (b, a) *to be supporting links, that is we have* $C_a(\emptyset) = C_b(\emptyset) = out$ *and* $C_a(\{b\}) = C_b(\{a\}) = in$*, while* (b, c) *should be an attacking link, that is,* $C_c(\emptyset) = in$ *and* $C_c(\{b\}) = out$*.*

D *has a model* $M = \{a, b\}$*.* c *is excluded because* b *is in* M*. However,* a *is in the model because* b *is, and* b *is in the model because* a *is. There is mutual cyclic support and the model appears unjustified. Note that* M *is a minimal model, so minimality is insufficient to exclude this type of unwanted models.*

In logic programming the Gelfond/Lifschitz reduction (Gelfond and Lifschitz 1988) used to define stable models for normal logic programs guarantees that there are no such cycles. We will apply a related construction here. Note that in AFs there are no supporting links and consequently self-supporting cycles do not play a role. This is what makes the definition of stable extensions of AFs rather simple.

[4] As pointed out to us by Mirek Truszczyński in personal communication, our well-founded models for ADFs appear to coincide with the ultimate well-founded models developed in (Denecker, Marek, and Truszczynski 2004). This article contains a discussion of further semantics which may be of interest to ADFs. An in depth study of these semantics is left as future work.

As we saw, the well-founded model can be defined for arbitrary ADFs. For the definition of stable models a minimal requirement is the ability to detect self-supporting cycles, and thus the notion of supporting versus attacking links.

Definition 5 *Let $D = (S, L, C)$ be an ADF. A link $(r, s) \in L$ is*

1. supporting *iff for no $R \subseteq par(s)$ we have that $C_s(R) = in$ and $C_s(R \cup \{r\}) = out$,*
2. attacking *if for no $R \subseteq par(s)$ we have that $C_s(R) = out$ and $C_s(R \cup \{r\}) = in$.*

If (r, s) is a supporting (attacking) link to s we will call r supporting (attacking) node for s. Note that a link (r, s) can be both supporting and attacking. However, in this case r does not have any influence on s whatsoever. We call (r, s) *redundant* in this case. Redundant links can simply be deleted from dialectical frameworks as they represent no real dependencies.

For links which are neither attacking nor supporting, the effect of a parent node is dependent on the status of other parent nodes.

Example 4 *Let $D = (\{a, b, c\}), \{(a, c), (b, c)\}, P)$ with $C_c(R) = in$ iff exactly one of $\{a, b\}$ is in. Now the two links are neither supporting nor attacking. For instance, if b is out then a supports c, if b is in then a attacks c.*

We call an ADF *monotonic* if all of its links are supporting. Indeed, if $D_1 = (S_1, L_1, C_1)$ and $D_2 = (S_2, L_2, C_2)$ are monotonic ADFs such that $S_1 \subseteq S_2$, $L_1 \subseteq L_2$ and C_1 is the restriction of C_2 to the nodes in S_1, then $Cn(D_1) \subseteq Cn(D_2)$. The following result will be relevant later.

Proposition 5 *Let $D = (S, L, C)$ be a monotonic ADF. Then D has a unique least model.*

Proof: Consider the operator $Th_D : 2^S \rightarrow 2^S$ defined as

$$Th_D(S) = \{r \in S \mid C_r(S) = in\}.$$

The operator is monotonic and thus possesses a least fixpoint lfp_D which can be reached by iterating Th_D on the empty set. It can be shown by induction on the number of iterations that all elements of the fixpoint must be contained in each model of D. Moreover, lfp_D clearly is a model of D. Hence, lfp_D is the least model of D. □

Now consider the subclass of ADFs all of whose links are either supporting or attacking (this will include in particular the weighted ADFs with positive and negative weights to be defined later). This class allows the most natural forms of argumentation based on statements pro or con a particular position to be modeled. We call such frameworks *bipolar* ADFs, or BADFs for short.

Our definition of stable models for this class is based on a slight reformulation of the definition of stable models for logic programs.

Proposition 6 *Let G be a (normal) logic program. M is a stable model of G iff M is (1) a model of G and (2) the least model of the* model-reduct $G^{\underline{M}}$ *of G obtained from G by*

1. *deleting all rules whose body is false in M,*
2. *eliminating negated literals from the remaining rules.*

Proof: First, observe that $G^{\underline{M}}$ is always a subset of G^M (the Gelfond/Lifschitz reduct of G). Thus N is a model of $G^{\underline{M}}$, if N is a model of G^M. Let M be a stable model of G. Then M is a model of G, and since $G^{\underline{M}} \subseteq G^M$, M is a model of $G^{\underline{M}}$. Moreover, all rules which are applied in the construction of the least model of G^M are contained in $G^{\underline{M}}$. Thus M is the least model of $G^{\underline{M}}$. Vice versa, let M be a model of G and M be the least model of $G^{\underline{M}}$ but not the least model of G^M. Since $G^{\underline{M}} \subseteq G^M$, M cannot be a model of G^M, i.e. there exists a rule r in G^M, such that body of r is true in M yet the head of r is not in M. Since $r \in G^M$, there exists a rule r' in G of the same form as r but with additional negative literals in the body, which are however not contained in M. Then, M is not a model of r' and thus not a model of G, contrary to our assumption. □

The major difference between the original definition and this reformulation is that the latter uses models and eliminates *all* rules whose bodies are false in the model, including positive ones.[5] Based on this idea stable models for BADFs can be defined as follows:

Definition 6 *Let $D = (S, L, C)$ be a bipolar ADF. 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 conditions C_s for each remaining node s to the remaining parents of s.*

Note that D^M has supporting links only, so according to Proposition 5 the least model exists.

The analogy with the Gelfond/Lifschitz reduct and, more precisely, the model-reduct introduced above should be obvious. We want to emphasize again that, contrary to the Gelfond/Lifschitz reduct, our reduct has to be applied to sets of statements which are already known to be models.

Let us see how our construction works for ADFs representing Dung systems. Here the proof standards for all nodes assign *in* if and only if no parent node is *in*. We consider only models when testing stability. If M is a model, then we know already that nodes not in M must be attacked by nodes in M. We also know that M is conflict free, otherwise it would not be a model. To check whether a model M is stable we eliminate nodes

[5] In this respect our reduct is closer to the FLP-reduct (Faber, Leone, and Pfeifer 2004). However, it is not identical to the FLP-reduct as we also eliminate negated literals from non-eliminated rules.

not in M and all related links. This leaves us with the set of nodes in M. Since there is no attack, all of them are trivially in the least model of the reduced ADF. Thus, for Dung-style ADFs, models and stable models coincide.

Proposition 7 *Let $\mathcal{A}$ be an argumentation framework, $D_{\mathcal{A}}$ its associated dialectical framework. E is a stable extension of $\mathcal{A}$ iff E is a stable model of $D_{\mathcal{A}}$.*

Proof: Follows from Proposition 1 (2) and the fact that models and stable models coincide for Dung-style ADFs. □

Example 5 *Consider ADF D discussed in Example 3 where a supports b, b supports a, and b attacks c. As before we have the following acceptance conditions: a is in iff b is and vice versa. Moreover, c is in unless b is.*

There are two models: $\{a, b\}$ and $\{c\}$. Only the latter is the expected one. Indeed, 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. The reduced framework has the empty set as least model. The set $\{a, b\}$ thus is not stable.

On the other hand, the reduct $D^{\{c\}}$ has no link at all. According to its acceptance condition c is in, that is, the least model of the reduct gives us back the original model; we thus have a stable model as intended.

It is worth mentioning that the ADFs we associated with normal logic programs earlier are in general not bipolar. For instance, the ADF for the program

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

has links from a and b to c which are neither supporting nor attacking. This shows that our definition of stable models for ADFs does not immediately capture the particular type of reduct needed for logic programs. However, if we choose a slightly more sophisticated representation using additional nodes for rules we can still obtain an interesting connection. Let $r = c \leftarrow a_1 \dots a_n, not\ b_1, \dots not\ b_m$ be a rule. We represent this rule by connecting the a_is and the b_js with node r, where connections from the former are supporting, from the latter attacking. It is easy to define the acceptance condition for r in such a way that r is *in* iff the body of r is satisfied. We then connect r and c with a supporting link. The acceptance condition for c is such that c is *in* whenever at least one rule with head c has its body satisfied. The resulting ADF is bipolar (unless the same atom appears positively and negatively in the body of the same rule, in which case the rule can be deleted without changing the stable models). We then can prove that stable models of the ADF constructed this way correspond to stable models of the program - modulo the atoms representing applicable rules.

Proposition 8 *Let G be a logic program, D the bipolar ADF constructed from G using rule names N (disjoint from the atoms in G) as described above. If M is a stable model of G and $N[M]$ are the names of rules with bodies true in M, then $M \cup N[M]$ is a stable model of D. Vice versa, if M is a stable model of D, then $M \setminus N$ is a stable model of G.*

Proof: (sketch) (1) Let G^+ be the program obtained from G by replacing each rule $head \leftarrow body$ with name n by the two rules $head \leftarrow n$ and $n \leftarrow body$. It can be shown that M is a stable model of G iff $M \cup N[M]$ is a stable model of G^+, where $N[M]$ are the names of G's rules with bodies true in M. (2) The ADF associated with the model-reduct of G^+ wrt. a model M (see Proposition 6) is the reduct of D wrt. model M. Moreover, the least model of a definite logic program G_d is identical to the least model of its associated ADF D_d. Together with Proposition 6 this implies the result. □

It even turns out that the addition of nodes allows us to transform each ADF $D = (S, L, C)$ into a BADF D' such that

1. each model of D is contained in a model of D', and
2. for each model M of D', $M \cap S$ is a model of D.

The additional nodes needed in D' represent arbitrary subsets S' of S. The node representing S' has supporting links from the elements of S', and attacking links from nodes in the complement of S'. The node is *in* iff all supporting nodes are *in* and no attacking node is *in*. A "regular" node r then has a supporting link from the node representing S' whenever $C_r(S' \cap par(r)) = in$.

This construction obviously is of limited practical relevance as it may lead to an exponential blowup - unlike the construction for logic programs described above.

Let us now turn to preferred semantics. According to Dung, a preferred extension of an AF $\mathcal{A}$ is a maximal admissible set of arguments. An admissible set R is conflict-free and defends itself against potential attackers, that is, each argument attacking an element in R is attacked by some element of R. This guarantees that elements of R continue to be defended even if R is "reasonably" extended (see also Dung's (1995) fundamental lemma).

Before defining preferred models for BADFs we give an alternative characterization of admissible sets for AFs.

Proposition 9 *Let $\mathcal{A} = (AR, att)$ be an argumentation framework. $E \subseteq AR$ is* admissible *in $\mathcal{A}$ iff there is some $R \subseteq (AR \setminus E)$ such that*

1. *no element in R attacks an element in E, and*
2. *E is a stable extension of the reduced argumentation framework*

$$\mathcal{A}\text{-}R = (AR \setminus R, \{(a, b) \in att \mid a, b \in AR \setminus R\}).$$

Proof: Let E be admissible in $\mathcal{A}$. Then E is conflict-free and attacks each $r \in (AR \setminus E)$ which attacks an element in E. Let

$$R = \{s \in (AR \setminus E) \mid (s, t) \in att \text{ implies } t \notin E\}.$$

E is a stable extension of $\mathcal{A}$-R.

Vice versa, if E is a stable extension of $\mathcal{A}$-R for some R consisting only of arguments not attacking E, then E is conflict-free and attacks each $r \in AR$ attacking some element of E (since such arguments cannot be contained in R). Thus E is admissible. □

Our definition of preferred models of bipolar ADFs will be based on a generalization of this characterization. For an ADF $D = (S, L, C)$ and $R \subseteq S$ we use D-R to denote the bipolar ADF obtained from D by deleting all nodes in R together with their proof standards and links they are contained in. Moreover, proof standards of the remaining nodes are restricted to the remaining parents.

Definition 7 *Let $D = (S, L, C)$ be a bipolar ADF. $M \subseteq S$ is* 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.*

Proposition 10 *Let $D = (S, L, C)$ be a finite bipolar ADF. D possesses at least one preferred model.*

Proof: Follows from the facts that (1) S - and thus the number of subsets of S - is finite, and (2) the empty set is trivially admissible. □

We next show that the same relationship between stable and preferred models as in Dung frameworks holds.

Proposition 11 *Let D be a bipolar ADF, M a stable model of D. Then M is a preferred model of D.*

Proof: If M is a stable model of D, then M is admissible since M is a stable model of D-$\emptyset$. It remains to be shown that M is maximal among the admissible sets.

Assume there is a proper superset M' of M which is admissible in D. Then there is an R such that M' is a stable model of D-R. Since $\emptyset \subseteq R$ and $M \subseteq M'$, the monotonic ADF $(D\text{-}R)^{M'}$ cannot have more nodes and links than the monotonic ADF D^M. Thus, the least model of $(D\text{-}R)^{M'}$ cannot contain more elements than the least model M of D^M. Since M' is a proper superset of M, it is not a stable model of D-R, contrary to our assumption. □

We finally show that Dung frameworks under preferred semantics are a special case of our approach.

Proposition 12 *Let $\mathcal{A}$ be an argumentation framework, $D_{\mathcal{A}}$ its associated dialectical framework. E is a preferred extension of $\mathcal{A}$ iff E is a preferred model of $D_{\mathcal{A}}$.*

Proof: It suffices to show that E is admissible in $\mathcal{A}$ iff E is admissible in $D_{\mathcal{A}}$. This follows from the facts that (1) R satisfies condition 1 in Proposition 9 iff it satisfies condition 1 in Definition 7, and (2) E is a stable extension of $\mathcal{A}$-R iff it is a stable model of $D_{\mathcal{A}}$-R (Proposition 7). □

We thus have successfully generalized the three most important semantics introduced by Dung for AFs to ADFs, respectively BADFs.

Weighted ADFs and proof standards

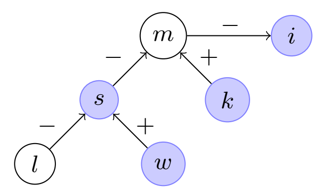
Acceptance conditions can be defined in a particularly convenient way through (positive and negative) weights of links. For this reason, we will introduce weighted ADFs in this section, and then show how some of the domain independent proof standards which have been discussed in the literature on legal argumentation can be represented. We start with qualitative weights and discuss numerical weights later on.

A weighted ADF is an ADF whose acceptance conditions are defined in terms of weights of links. For a given ADF $D = (S, L, C)$ we thus need an additional function $w : L \to V$, where V is some set of weights. In order to simplify notation, when applying w to $(r, s) \in L$ we will write $w(r, s)$ rather than $w((r, s))$.

In the simplest case we can use $V = \{+, -\}$ where $+$ represents a supporting (positive) link, $-$ an attacking (negative) link. The resulting ADFs clearly are bipolar, and so we can immediately apply all semantics. In this setting the following domain independent acceptance conditions immediately come to mind:

1. $C_s(R) = in$ iff R contains no node attacking s,
2. $C_s(R) = in$ iff R contains no node attacking s and at least one node supporting s,
3. $C_s(R) = in$ iff R contains more nodes supporting s than nodes attacking s
4. $C_s(R) = in$ iff R contains all nodes supporting s and no node attacking s.

The weights also help us to make the dependency graphs more readable. Recall Example 1 where w and k are *in*, l is *out*, and the acceptance condition of the 3 other nodes corresponds to 4. in the list above. Using weights, the dependency graph - and the single model - can be represented as follows:



Following Prakken (2009) we can also introduce “necessary axioms” by defining $C_s(R) = in$ for all $R \subseteq par(s)$. This means that such axioms are always accepted and can never be successfully attacked. Prakken’s “issues” are acceptable only in case they are “backed with a further argument”. This can obviously be modeled by choosing an acceptance condition which yields *out* in case no supporting node is in R.

Farley and Freeman (1995) introduced a model of legal argumentation which distinguishes four types of arguments: (1) *valid* arguments based on deductive inference, (2) *strong* arguments based on inference with defeasible rules, (3) *credible* arguments where premises give some evidence, (4) *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 distinguished 5 different standards of proof which can be modeled in ADFs as follows:

1. *Scintilla of Evidence*: at least one weak, defendable argument.
   $C_s(R) = in$ iff $\exists r \in R : w(r, s) \in \{+v, +s, +c, +w\}$.
2. *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 $\exists r' \in R : w(r', s) \in \{+v, +s, +c\}$.
3. *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 $t \in R$.
4. *Beyond Reasonable Doubt*: at least one strong, defendable argument and the other side's arguments all defeated: $C_s(R) = in$ iff
   - $\exists r \in R : w(r, s) \in \{+v, +s\}$ and
   - $w(t, s) \notin \{-v, -s, -c, -w\}$ for all $t \in R$.
5. *Beyond Doubt*: at least one valid argument and the other side's arguments all defeated: $C_s(R) = in$ iff
   - $\exists r \in R : w(r, s) = +v$ and
   - $w(t, s) \notin \{-v, -s, -c, -w\}$ for all $t \in R$.

More fine grained distinctions can be made if V is a set of numerical weights. We can then define several other natural proof standards. Examples are:

- $C_s(R) = in$ iff the sum of all weights of links from elements of R to s is positive,
- $C_s(R) = in$ iff the maximal positive weight of incoming links is higher than the maximal negative weight,
- $C_s(R) = in$ iff the difference between the maximal positive weight and the absolute value of the maximal negative weight is above a certain threshold.

We emphasize that weighted ADFs allow for a simpler and domain-independent formulation of acceptance conditions, but are not a proper generalization of standard ADFs (i.e. ADF without weighted links). In fact, the weight-based evaluation of a proof standard can always be compiled into a regular boolean expression.

Prioritized BADFs

We have seen in the last section how weights on links can be used to define domain independent proof standards. Another useful option is to use qualitative preferences. Let $D = (S, L, C)$ be a bipolar ADF. Moreover, assume for each node $s \in S$ we are given a strict partial order $>_s$ on the links in L leading to s. We can now take the preference information into account by defining $C_s(R) = in$ iff for each attacking link $(r, s) \in L$ such that $r \in R$ there is a supporting link $(r', s) \in L$ with $r' \in R$ such that $(r', s) >_s (r, s)$.

This definition assumes that a node is *out* unless its joint support is more preferred than its joint attack. We can also reverse this by defining $C_s(R) = out$ iff for each supporting link $(r, s) \in L$ such that $r \in R$ there is an attacking link $(r', s) \in L$ with $r' \in R$ such that $(r', s) >_s (r, s)$. The default case is now that a node is *in* unless its attackers are jointly preferred. Of course, nothing prevents us from having nodes of both kinds in a single prioritized BADF.

This treatment of preferences is very different from the one proposed in (Amgoud and Cayrol 1998). Rather than preferences on nodes, we use preferences on links in the graph. For some difficulties of the former approach see (Dimopoulos, Moraitis, and Amgoud 2009).

Complexity of ADFs

In this section, we assume that acceptance conditions of ADFs are given via propositional formulas. We start with the case of well-founded semantics.

Proposition 13 *Deciding whether a given set of statements is the well-founded model of a given ADF is* coNP*-hard.*

Proof: The following simple reduction from the coNP-hard problem of validity of propositional logic to the considered problem shows the claim:

For a formula ϕ over atoms A, construct

$$D_\phi = (A \cup \bar{A} \cup \{t\}, \{(a, \bar{a}), (\bar{a}, a), (a, t) \mid a \in A\}, C)$$

where $\bar{A} = \{\bar{a} \mid a \in A\}$ and t are fresh statements (disjoint from A), and C is given as $C_a = \neg\bar{a}$ and $C_{\bar{a}} = \neg a$ for each $a \in A$, and $C_t = \phi$.

We show that $\{t\}$ is the well-founded model of D_ϕ iff ϕ is valid. First note that a's and $\bar{a}$'s mutually attack each other and thus are not in the well-founded model. The following obvious observation shows the claim: $t \in acc(\emptyset, \emptyset)$ iff $C_t(S) = in$ for all $S \subseteq A$ iff ϕ is true under all assignments S iff ϕ is valid. □

We next turn to BADFs. Deciding whether an ADF is bipolar is also intractable (and so is deciding whether a link is supporting or attacking), but the well-founded semantics for BADFs becomes tractable once the attacking and supporting links are known.

Proposition 14 *Deciding whether a given ADF is bipolar is* coNP*-hard.*

Proof: We can use the construction from the proof of Proposition 13. Obviously links $(a, \bar{a})$ and $(\bar{a}, a)$ are attacking, and links (a, t) are supporting, in case ϕ is valid ($C_t(S)$ always yields *in* then). In case, ϕ is not valid, it is however not necessarily the case that links (a, t) are not supporting. But we can guarantee this property, if we restrict ourselves to formulas ϕ which evaluate to true under the assignment where all atoms are assigned false. Obviously, the validity problem remains coNP-hard for such formulas. □

Proposition 15 *Deciding whether a given set of statements is the well-founded model of a given BADF with supporting links L^+ and attacking links L^- can be done in polynomial time.*

Proof: It is sufficient to show that deciding $r \in acc(A, R)$, resp. $r \in reb(A, R)$, can be done in polynomial time. Let A' be the union of sets $A \cap par(r)$ and the supporting nodes for r. To decide $r \in acc(A, R)$ amounts in case of BADFs to check whether $C_r(A') = in$. This can be seen as follows: Let S be any set of statements such that $A \cap par(r) \subseteq S \subseteq par(r)$. Then, each statement in $T = A' \setminus S$ is attacking and each statement in $U = S \setminus A'$ is supporting. By the definition of attacking statements, we get from $C_r(A') = in$ that also $C_r(A' \setminus T) = in$, and further, by the definition of supporting statements, that also $C_r((A' \setminus T) \cup U) = in$. Note that $(A' \setminus T) \cup U = S$. Thus, $C_r(S) = in$ holds for all relevant sets S.

Similarly, one can show that deciding $r \in reb(A, R)$ amounts in case of BADFs to check whether $C_r(A'') = out$, where A'' is the union of sets $A \cap par(r)$ and the attacking nodes for r. □

Proposition 16 *Deciding whether a statement s is contained in some (resp. all) stable models of a BADF with supporting links L^+ and attacking links L^- is* NP-*complete (resp.* coNP-*complete).*

Proof: We only have to show the membership parts, since known hardness results (Dimopoulos and Torres 1996) about Dung frameworks carry over to ADFs. For the NP-result, we provide the following algorithm which decides, given a BADF $D = (S, L, C)$ and a statement s, whether s is contained in a stable model of D: Guess a set $M \subseteq S$ with $s \in M$ and check whether (i) M is a model of D and (ii) M is the least of model of D^M. Both tasks can be done in polynomial time (wrt. the size of D) ad (i): this corresponds to the problem of model checking for a propositional formula. ad (ii): Constructing D^M is easy. Computing the (unique) least model of a monotonic ADF (D^M is such an ADF), takes at most quadratic time by using the fixpoint construction in the proof of Proposition 5.

This shows the NP-result. The coNP-result is by a similar argumentation using the complementary problem (use M as above but with $s \notin M$). □

Proposition 17 *Deciding whether a statement s is contained in some (resp. all) preferred models of a BADF with supporting links L^+ and attacking links L^- is* NP-*complete (resp.* Π_2^P-*complete).*

Proof: Again, we only have to show the membership parts, since known hardness results (Dimopoulos and Torres 1996; Dunne and Bench-Capon 2002) about Dung frameworks carry over to ADFs. For the NP-result, we provide the following algorithm which decides, given a BADF $D = (S, L, C)$ and a statement s, whether s is contained in a set admissible in D (and thus whether s is contained in some preferred extension of D). We guess sets $M, R \subseteq S$, such that $s \in M$ and such that there is no attacking link from any element of R to any element of M, and then check whether M is a stable model of D-R. To check whether a set of statements is a stable model of a BADF has been shown to be in P already in the proof of Proposition 16.

For the Π_2^P-result, we show Σ_2^P-membership for the complementary problem. To this end, consider the following algorithm: guess a set M with $s \notin M$, check whether (i) M satisfies conditions 1 and 2 of Definition 7, and (ii) whether no $N \supset M$ satisfies conditions 1 and 2 of Definition 7. Using a similar argumentation as above, it is clear that (i) can be done in polynomial time and (ii) can be done via an NP-oracle. □

Thus, with the exception of well-founded semantics for non-bipolar ADFs, our generalization of Dung frameworks does not increase the complexity of important reasoning tasks given the link types are already known.

Discussion and related work

In this paper we introduced abstract dialectical frameworks, a powerful generalization of Dung frameworks where various forms of dependencies among statements can be represented by using acceptance conditions. We introduced grounded, stable and preferred semantics for our frameworks, the latter for a slightly restricted class called bipolar. We also presented various ways of defining acceptance conditions based on weights on links, respectively qualitative preferences. Moreover, we showed how to represent legal proof standards.

Our approach shares some motivation with bipolar argumentation frameworks (Cayrol and Lagasquie-Schiex 2009), an extension of argumentation frameworks that includes a second relation expressing support. However, our proposal is more flexible and goes further in several respects. In particular, rather than adding a second type of links we allow for a whole variety of link and node types. Moreover, there is a fundamental difference between what is considered a conflict in bipolar AFs and ADFs.

Example 6 *Assume you plan to go swimming in the afternoon. There are clouds indicating it might rain. However, the (reliable) weather report says that winds will blow away the clouds so that there will be no rain. Now c supports r, r attacks s and w attacks r. Using appropriate acceptance conditions, assuming w's attack on r is stronger than c's support, we get $\{c, w, s\}$ as the single well-founded, stable and preferred model which makes perfect sense. However, this set is not +conflict-free in the sense of (Cayrol and Lagasquie-Schiex 2009).*

To model the stronger notion of +conflict-freeness in ADFs one has to add an attack link from a to c whenever c is attacked by a node b and a, directly or indirectly, supports b. The example suggests that this may not always be desired.

Oren and Norman (2008) propose an extension of AFs where an argument needs evidential support in order to attack other arguments. This support is ultimately based on a special argument representing "support from the environment". Again, our framework is more general and provides additional flexibility, e.g. in using weights for determining the status of arguments.

Weighted argumentation frameworks were discussed in a recent paper by Dunne et al. (2009). However, the weights there are used for inconsistency handling, that is, for determining which links to disregard in case of non-existence of stable extensions, not for determining the status of arguments as in our proposal.

We have restricted our discussion in this paper to Dung's standard semantics. Several alternative proposals exist like semi-stable (Caminada 2006) and ideal (Dung, Mancarella, and Toni 2007) semantics. In future work we plan to generalize those as well.

We also plan to reconstruct the Carneades framework (Gordon, Prakken, and Walton 2007) as an ADF. This will provide Carneades with the different semantics we developed for ADFs. It will also demonstrate one of the common uses we foresee for our ADFs, namely as target systems for translations from less abstract systems. Due to their greater expressiveness we expect translations to ADFs to be easier than those to AFs.

Acknowledgments

This work was supported by the Vienna Science and Technology Fund (WWTF) under grant ICT08-028 and by Deutsche Forschungsgemeinschaft (DFG) under grant Br 1817/3. We would like to thank Tom Gordon, Henry Prakken, Mirek Truszczyński and the anonymous reviewers for helpful comments.

References

Amgoud, L., and Besnard, P. 2009. Bridging the gap between abstract argumentation systems and logic. In *Proc. SUM'09*, 12–27.

Amgoud, L., and Cayrol, C. 1998. On the acceptability of arguments in preference-based argumentation. In *Proc. UAI'98*, 1–7.

Atkinson, K., and Bench-Capon, T. J. M. 2007. Argumentation and standards of proof. In *Proc. ICAIL'07*, 107–116.

Caminada, M. 2006. Semi-stable semantics. In *Proc. COMMA'06*, 121–130.

Cayrol, C., and Lagasquie-Schiex, M.-C. 2009. Bipolar abstract argumentation systems. In Rahwan, I., and Simari, G., eds., *Argumentation in Artificial Intelligence*. Springer. 65–84.

Denecker, M.; Marek, V. W.; and Truszczynski, M. 2004. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Inf. Comput.* 192(1):84–121.

Dimopoulos, Y., and Torres, A. 1996. Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.* 170(1-2):209–244.

Dimopoulos, Y.; Moraitis, P.; and Amgoud, L. 2009. Extending argumentation to make good decisions. In *Proc. ADT'09*, 225–236.

Dung, P. M.; Mancarella, P.; and Toni, F. 2007. Computing ideal sceptical argumentation. *Artif. Intell.* 171(10-15):642–674.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2):321–358.

Dunne, P. E., and Bench-Capon, T. J. M. 2002. Coherence in finite argument systems. *Artif. Intell.* 141(1/2):187–203.

Dunne, P. E.; Hunter, A.; McBurney, P.; Parsons, S.; and Wooldridge, M. 2009. Inconsistency tolerance in weighted argument systems. In *Proc. AAMAS'09*, 851–858.

Faber, W.; Leone, N.; and Pfeifer, G. 2004. Recursive aggregates in disjunctive logic programs: Semantics and complexity. In *Proc. JELIA'04*, 200–212.

Farley, A. M., and Freeman, K. 1995. Burden of proof in legal argumentation. In *Proc. ICAIL'95*, 156–164.

Gelder, A. V.; Ross, K. A.; and Schlipf, J. S. 1991. The well-founded semantics for general logic programs. *J. ACM* 38(3):620–650.

Gelfond, M., and Lifschitz, V. 1988. The Stable Model Semantics for Logic Programming. In *Proc. ICLP'88*, 1070–1080.

Gordon, T. F., and Walton, D. 2009. Proof burdens and standards. In Rahwan, I., and Simari, G., eds., *Argumentation in Artificial Intelligence*. 239–258.

Gordon, T. F.; Prakken, H.; and Walton, D. 2007. The Carneades model of argument and burden of proof. *Artif. Intell.* 171(10-15):875–896.

Modgil, S., and Caminada, M. 2009. Proof theories and algorithms for abstract argumentation frameworks. In Rahwan, I., and Simari, G., eds., *Argumentation in Artificial Intelligence*. Springer. 105–129.

Oren, N., and Norman, T. J. 2008. Semantics for evidence-based argumentation. In *Proc. COMMA'08*, 276–284.

Prakken, H. 2009. An abstract framework for argumentation with structured arguments. Technical Report UU-CS-2009-019, Utrecht University.

Wooldridge, M.; Dunne, P. E.; and Parsons, S. 2006. On the complexity of linking deductive and abstract argument systems. In *Proc. AAAI'06*, 299–304.

Wyner, A.; Bench-Capon, T.; and Dunne, P. E. 2009. Instantiating knowledge bases in abstract argumentation frameworks. In *Proc. AAAI Fall Symposium - The Uses of Computational Argumentation*.