TLP **8** (4): 527–543, 2008. © 2008 Cambridge University Press 527
doi:10.1017/S1471068408003359 First published online 8 May 2008 Printed in the United Kingdom

TECHNICAL NOTE
*Preferred extensions as stable models**

JUAN CARLOS NIEVES and ULISES CORTÉS

Universitat Politècnica de Catalunya, Software Department (LSI), c/Jordi Girona 1-3, E08034, Barcelona, Spain
(*e-mail:* {jcnieves,ia}@lsi.upc.edu)

MAURICIO OSORIO

Universidad de las Américas – Puebla, CENTIA, Sta. Catarina Mártir, Cholula, Puebla, 72820 México
(*e-mail:* osoriomauri@googlemail.com)

submitted 6 June 2007; revised 17 December 2007; accepted 29 February 2008

Abstract

Given an argumentation framework *AF*, we introduce a mapping function that constructs a disjunctive logic program *P*, such that the preferred extensions of *AF* correspond to the stable models of *P*, after intersecting each stable model with the relevant atoms. The given mapping function is of polynomial size w.r.t. *AF*.

In particular, we identify that there is a direct relationship between the minimal models of a propositional formula and the preferred extensions of an argumentation framework by working on representing the defeated arguments. Then we show how to infer the preferred extensions of an argumentation framework by using UNSAT algorithms and disjunctive stable model solvers. The relevance of this result is that we define a direct relationship between one of the most satisfactory argumentation semantics and one of the most successful approach of nonmonotonic reasoning i.e., logic programming with the stable model semantics.

KEYWORDS: preferred semantics, abstract argumentation semantics, stable model semantics, minimal models

1 Introduction

Dung's approach, presented in Dung (1995), is a unifying framework which has played an influential role on argumentation research and artificial intelligence (AI). In fact, Dung's approach has influenced subsequent proposals for argumentation systems, e.g., (Bench-Capon 2002). Besides, Dung's approach is mainly relevant in fields where conflict management plays a central role. For instance, Dung showed that his theory naturally captures the solutions of the theory of n-person games and the well-known stable marriage problem.

* This is a revised and improved version of the paper *Inferring preferred extensions by minimal models* which appeared in Guillermo R. Simari and Paolo Torroni (Eds), proceedings of the workshop Argumentation and Non-Monotonic Reasoning (LPNMR-07 Workshop).

Dung defined four argumentation semantics: *stable semantics*, *preferred semantics*, *grounded semantics,* and *complete semantics*. The central notion of these semantics is the *acceptability of the arguments*. The main argumentation semantics for collective acceptability are the grounded semantics and the preferred semantics (Prakken and Vreeswijk 2002; ASPIC:Project 2005). The first one represents a skeptical approach and the second one represents a credulous approach.

Dung showed that argumentation can be viewed as logic programming with *negation as failure*. Specially, he showed that the grounded semantics can be characterized by the well-founded semantics (Gelder *et al.* 1991), and the stable semantics by the stable model semantics (Gelfond and Lifschitz 1991). This result is of great importance because it introduces a general method for generating metainterpreters for argumentation systems (Dung 1995). Following this issue, we will prove that it is possible to characterize the preferred semantics based on the minimal models of a propositional formula (Theorem 1). We will also show that the preferred semantics can be characterized by the stable models of a positive disjunctive logic program (Theorem 3). The importance of this characterization is that we are defining a direct relationship between one of the most satisfactory argumentation semantics and may be the most successful approach of nonmonotonic reasoning of the last two decades i.e., logic programming with the stable model semantics.

As a natural consequence of our result, we present two easy-to-use forms for inferring the preferred extensions of an argumentation framework (*AF*). The first one is based on a mapping function which is quadratic size w.r.t. the number of arguments of *AF* and UNSAT algorithms. The second one is also based on a mapping function which is quadratic size w.r.t. the number of arguments of *AF* and disjunctive stable model solvers.

It is worth mentioning that the decision problem of the preferred semantics is hard, since it is co-NP-Complete (Dunne and Bench-Capon 2004). In fact, we can find different strategies for computing the preferred semantics (Cayrol *et al.* 2003; Besnard and Doutre 2004; Dung *et al.* 2006; Dung *et al.* 2007). However, we can find really few implementations of them (ASPIC:Project 2006; Gaertner and Toni 2007). One of the relevant points of our result is that we can take advance of efficient disjunctive stable model solvers, e.g., the DLV System (DLV 1996), for inferring the preferred semantics. The DLV System is a successful stable model solver that includes deductive database optimization techniques, and nonmonotonic reasoning optimization techniques in order to improve its performance (Leone *et al.* 2002; Gebser *et al.* 2007). In fact, we can implement the preferred semantics inside object-oriented programs based on our characterization and the DLV JAVA Wrapper (Ricca 2003).

The rest of the paper is divided as follows: In Section 2, we present some basic concepts of logic programs and argumentation theory. In Section 3, we present a characterization of the preferred semantics by minimal models. In Section 4, we present how to compute the preferred semantics by using the minimal models of a positive disjunctive logic program. Finally in the last section, we present our conclusions.

2 Background

In this section, we present the syntax of a valid logic program, the definition of the stable model semantics, and the definition of the preferred semantics. We will use basic well-known definitions in complexity theory, such as that of co-NP-complete problem.

2.1 Logic programs: syntax

The language of a propositional logic has an alphabet consisting of

(i) A signature $\mathscr{L}$ that is a finite set of elements that we call atoms, denoted usually as $p_0, p_1, \ldots$;
(ii) connectives : $\vee, \wedge, \leftarrow, \neg, \bot, \top$;
(iii) auxiliary symbols : (,),

where $\vee, \wedge, \leftarrow$ are 2-place connectives, $\neg$ is 1-place connective, and $\bot, \top$ are 0-place connectives or constant symbols. A literal is an atom, a, or the negation of an atom $\neg a$. Given a set of atoms $\{a_1, \ldots, a_n\}$, we write $\neg\{a_1, \ldots, a_n\}$ to denote the set of literals $\{\neg a_1, \ldots, \neg a_n\}$. Formulae are constructed as usual in logic. A theory T is a finite set of formulae. By $\mathscr{L}_T$, we denote the signature of T, namely the set of atoms that occur in T.

A general clause, C, is denoted by $a_1 \vee \cdots \vee a_m \leftarrow l_1, \ldots, l_n$,[1] where $m \geqslant 0$, $n \geqslant 0$, $m + n > 0$, each a_i is an atom, and each l_i is a literal. When $n = 0$ and $m > 0$, the clause is an abbreviation of $a_1 \vee \cdots \vee a_m \leftarrow \top$. When $m = 0$, the clause is an abbreviation of $\bot \leftarrow l_1, \ldots, l_n$. Clauses of this form are called constraints (the rest, nonconstraint clauses). A general program, P, is a finite set of general clauses. Given a universe U, we define the *complement* of a set $S \subseteq U$ as $\widetilde{S} = U \backslash S$.

We point out that whenever we consider logic programs, our negation $\neg$ corresponds to the default negation *not* used in Logic Programming. Also, it is convenient to remark that in this paper we are not at all using the so-called *strong negation* used in ASP.

2.2 Stable model semantics

First, to define the stable model semantics, let us define some relevant concepts.

Definition 1
Let T be a theory, an interpretation I is a mapping from $\mathscr{L}_T$ to $\{0, 1\}$ meeting the following conditions:

(1) $I(a \wedge b) = min\{I(a), I(b)\}$;
(2) $I(a \vee b) = max\{I(a), I(b)\}$;
(3) $I(a \leftarrow b) = 0$ iff $I(b) = 1$ and $I(a) = 0$;

[1] $l_1, \ldots, l_n$ represents the formula $l_1 \wedge \cdots \wedge l_n$.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

(4) $I(\neg a) = 1 - I(a)$;
(5) $I(\bot) = 0$;
(6) $I(\top) = 1$.

It is standard to provide interpretations only in terms of a mapping from $\mathscr{L}_T$ to $\{0, 1\}$. Moreover, it is easy to prove that this mapping is unique by virtue of the definition by recursion (van Dalen 1994).

An interpretation I is called a model of P iff for each clause $c \in P$, $I(c) = 1$. A theory is consistent if it admits a model, otherwise it is called inconsistent. Given a theory T and a formula α, we say that α is a logical consequence of T, denoted by $T \models \alpha$, if for every model I of T it holds that $I(\alpha) = 1$. It is a well-known result that $T \models \alpha$ iff $T \cup \{\neg\alpha\}$ is inconsistent. It is possible to identify an interpretation with a subset of a given signature. For any interpretation, the corresponding subset of the signature is the set of all atoms that are true w.r.t. the interpretation. Conversely, given an arbitrary subset of the signature, there is a corresponding interpretation defined by specifying that the mapping assigned to an atom in the subset is equal to 1 and otherwise to 0. We use this view of interpretations freely in the rest of the paper.

We say that a model I of a theory T is a minimal model if there does not exist a model I' of T different from I, such that $I' \subset I$. Maximal models are defined in the analogous form.

By using logic programming with stable model semantics, it is possible to describe a computational problem as a logic program whose stable models correspond to the solutions of the given problem. The following definition of a stable model for general programs was presented in Gelfond and Lifschitz (1991).

Let P be any general program. For any set $S \subseteq \mathscr{L}_P$, let P^S be the general program obtained from P by deleting

(i) each rule that has a formula $\neg l$ in its body with $l \in S$, and then
(ii) all formulae of the form $\neg l$ in the bodies of the remaining rules.

Clearly, P^S does not contain $\neg$. Hence S is a stable model of P iff S is a minimal model of P^S.

In order to illustrate this definition, let us consider the following example.

Example 1
Let $S = \{b\}$ and P be the following logic program:

$$b \leftarrow \neg a. \qquad b \leftarrow \top.$$
$$c \leftarrow \neg b. \qquad c \leftarrow a.$$

We can see that P^S is

$$b \leftarrow \top. \qquad c \leftarrow a.$$

Notice that P^S has three models: $\{b\}, \{b, c\}$ and $\{a, b, c\}$. Since the minimal model amongst these models is $\{b\}$, we can say that S is a stable model of P.

2.3 *Argumentation theory*

Now, we define some basic concepts of Dung's argumentation approach. The first one is that of an argumentation framework. An argumentation framework captures

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

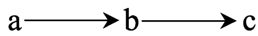

Fig. 1. Graph representation of the argumentation framework $AF = \langle\{a, b, c\}, \{(a, b), (b, c)\}\rangle$.

the relationships between the arguments. (All the definitions of this subsection were taken from the seminal paper (Dung 1995).)

Definition 2
An argumentation framework is a pair $AF = \langle AR, attacks\rangle$, where AR is a finite set of arguments, and *attacks* is a binary relation on AR, i.e., $attacks \subseteq AR \times AR$.

For two arguments a and b, we say that a *attacks* b (or b is attacked by a) if $attacks(a, b)$ holds. Notice that the relation *attacks* does not yet tell us with which arguments a dispute can be won; it only tells us the relation of two conflicting arguments.

It is worth mentioning that any argumentation framework can be regarded as a directed graph. For instance, if $AF = \langle\{a, b, c\}, \{(a, b), (b, c)\}\rangle$, then AF can be represented as shown in Figure 1.

Definition 3
A set S of arguments is said to be conflict-free if there are no arguments a, b in S such that a *attacks* b.

A central notion of Dung's framework is *acceptability*. It captures how an argument that cannot defend itself, can be protected by a set of arguments.

Definition 4
(1) An argument $a \in AR$ is *acceptable* w.r.t. a set S of arguments iff for each argument $b \in AR$: If b attacks a, then b is attacked by an argument in S. (2) A conflict-free set of arguments S is *admissible* iff each argument in S is acceptable w.r.t. S.

Let us consider the argumentation framework AF of Figure 1. We can see that AF has three admissible sets: $\{\}$, $\{a\}$, and $\{a, c\}$. Intuitively, an admissible set is a coherent point of view. Since an argumentation framework could have several coherent point of views, one can take the maximum admissible sets in order to get maximum coherent point of views of an argumentation framework. This idea is captured by Dung's framework with the concept of *preferred extension*.

Definition 5
A preferred extension of an argumentation framework AF is a maximal (w.r.t. inclusion) admissible set of AF.

Since an argumentation framework could have more than one preferred extension, the preferred semantics is called credulous. The argumentation framework of Figure 1 has just one preferred extension which is $\{a, c\}$.

Remark 1
By definition, it is clear that any argument which belongs to a preferred extension E is acceptable w.r.t. E. Hence, we will say that any argument which does not belong to some preferred extension is a *defeated argument*.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

3 Preferred extensions and UNSAT problem

In this section, we will define a mapping function that constructs a propositional formula, such that its minimal models characterize the preferred extensions of an argumentation framework. This characterization will provide a method for computing preferred extensions based on Model Checking and Unsatisfiability (UNSAT).

In order to characterize the preferred semantics in terms of minimal models, we will introduce some concepts.

Definition 6
Let T be a theory with signature $\mathscr{L}$. We say that $\mathscr{L}'$ is a copy-signature of $\mathscr{L}$ iff

- $\mathscr{L} \cap \mathscr{L}' = \emptyset$,
- the cardinality of $\mathscr{L}'$ is the same to $\mathscr{L}$ and
- there is a bijective function f from $\mathscr{L}$ to $\mathscr{L}'$.

It is well known that there exists a bijective function from one set to another if both sets have the same cardinality. Now one can establish an important relationship between maximal and minimal models.

Proposition 1
Let T be a theory with signature $\mathscr{L}_T$. Let $\mathscr{L}'$ be a copy-signature of $\mathscr{L}_T$. By $g(T)$ we denote the theory obtained from T by replacing every occurrence of an atom x in T by $\neg f(x)$. Then M is a maximal model of T iff $f(\mathscr{L}_T \backslash M)$ is a minimal model of $g(T)$.

Proof
See appendix. □

Our representations of an argumentation framework use the predicate $d(x)$, where the intended meaning of $d(x)$ is: "the argument x is defeated." By considering the predicate $d(x)$, we will define a mapping function from an argumentation framework to a propositional formula. This propositional formula captures two basic conditions which make an argument to be defeated.

Definition 7
Let $AF = \langle AR, attacks \rangle$ be an argumentation framework, then $\alpha(AF)$ is defined as follows:

$$\alpha(AF) = \bigwedge_{a \in AR} \Big(\Big(\bigwedge_{b:(b,a) \in attacks} d(a) \leftarrow \neg d(b) \Big) \wedge \Big(\bigwedge_{b:(b,a) \in attacks} d(a) \leftarrow \bigwedge_{c:(c,b) \in attacks} d(c) \Big) \Big).$$

(1) The first condition of $\alpha(AF)$ ($\bigwedge_{b:(b,a)\in attacks} d(a) \leftarrow \neg d(b)$) suggests that the argument a is defeated when any one of its adversaries is not defeated.
(2) The second condition of $\alpha(AF)$ ($\bigwedge_{b:(b,a)\in attacks} d(a) \leftarrow \bigwedge_{c:(c,b)\in attacks} d(c)$) suggests that the argument a is defeated when all the arguments that defend[2] a are defeated.

[2] We say that c defends a if b attacks a and c attacks b.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

Since $\alpha(AF)$ captures conditions that make an argument to be defeated, it is quite obvious that any argument that satisfies these conditions could not belong to an admissible set. Therefore these arguments also could not belong to a preferred extension.

Notice that $\alpha(AF)$ is a *finite grounded formula*, this means that it does not contain predicates with variables; hence, $\alpha(AF)$ is essentially a propositional formula (just considering the atoms like $d(a)$ as d_a) of propositional logic. In order to illustrate the propositional formula $\alpha(AF)$, let us consider the following example.

Example 2
Let $AF = \langle AR, attacks \rangle$ be the argumentation framework of Figure 1. We can see that $\alpha(AF)$ is

$$(d(b) \leftarrow \neg d(a)) \wedge (d(b) \leftarrow \top) \wedge (d(c) \leftarrow \neg d(b)) \wedge (d(c) \leftarrow d(a)).$$

Observe that $\alpha(AF)$ has no propositional clauses w.r.t. argument a. This is essentially because $\alpha(AF)$ is capturing the arguments which could be defeated and the argument a will be always an acceptable argument.

It is worth mentioning that given an argumentation framework AF, $\alpha(AF)$ will have at most $2n^2$ propositional clauses such that n is the number of arguments in AR and the maximum length[3] of each propositional clause is $n+1$. Hence, we can say that $\alpha(AF)$ is quadratic size w.r.t. the number of arguments of AF.

Essentially $\alpha(AF)$ is a propositional representation of the argumentation framework AF. However, $\alpha(AF)$ has the property that its minimal models characterize AF's preferred extensions. In order to formalize this property, let us consider the following proposition which was proved by Besnard and Doutre in 2004.

Proposition 2
(Besnard and Doutre 2004). Let $AF = \langle AR, attacks \rangle$ be an argumentation framework. Let $\beta(AF)$ be the formula

$$\bigwedge_{a \in AR} ((a \rightarrow \bigwedge_{b:(b,a) \in attacks} \neg b) \wedge (a \rightarrow \bigwedge_{b:(b,a) \in attacks} (\bigvee_{c:(c,b) \in attacks} c))),$$

then, a set $S \subseteq AR$ is a preferred extension iff S is a maximal model of the formula $\beta(AF)$.

In contrast with $\alpha(AF)$ which captures conditions that make an argument to be defeated, $\beta(AF)$ captures conditions that make an argument acceptable. However, we will prove that when the mapping $f(x)$ of the theory $g(\beta(AF))$ corresponds to $d(x)$ such that $x \in AF$, $\alpha(AF)$ is logically equivalent to $g(\beta(AF))$ (see the proof of Theorem 1). For instance, let us consider the argumentation framework AF of Example 2. The formula $\beta(AF)$ is

$$(\neg a \leftarrow b) \wedge (\bot \leftarrow b) \wedge (\neg b \leftarrow c) \wedge (a \leftarrow c).$$

[3] The length of our propositional clauses C is given by the number of atoms in the head of C plus the number of literals in the body of C

If we replace each atom x by the expression $\neg d(x)$, we get

$$(\neg\neg d(a) \leftarrow \neg d(b)) \wedge (\bot \leftarrow \neg d(b)) \wedge (\neg\neg d(b) \leftarrow \neg d(c)) \wedge (\neg d(a) \leftarrow \neg d(c)).$$

Now, if we apply transposition to each implication, we obtain

$$(d(b) \leftarrow \neg d(a)) \wedge (d(b) \leftarrow \top) \wedge (d(c) \leftarrow \neg d(b)) \wedge (d(c) \leftarrow d(a)).$$

The latter formula corresponds to $\alpha(AF)$. The following theorem is a straightforward consequence of Proposition 2 and Proposition 1. Given an argumentation framework $AF = \langle AR, attacks \rangle$ and $E \subseteq AR$, we define the set compl(E) as $\{d(a) | a \in AR \backslash E\}$. Essentially, compl($E$) expresses the complement of E w.r.t. AR.

Theorem 1

Let $AF = \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. When the mapping $f(x)$ of the theory $g(\beta(AF))$ corresponds to $d(x)$ such that $x \in AR$, the following condition holds: S is a preferred extension of AF iff compl(S) is a minimal model of $\alpha(AF)$.

Proof

See appendix. □

This theorem shows that it is possible to characterize the preferred extensions of an argumentation framework AF by considering the minimal models of $\alpha(AF)$. In order to illustrate Theorem 1, let us consider again $\alpha(AF)$ of Example 2. This formula has three models: $\{d(b)\}$, $\{d(b), d(c)\}$, and $\{d(a), d(b), d(c)\}$. Then the only minimal model is $\{d(b)\}$, this implies that $\{a, c\}$ is the only preferred extension of AF. In fact, each model of $\alpha(AF)$ implies an admissible set of AF, this means that $\{a, c\}$, $\{a\}$, and $\{\}$ are the admissible sets of AF.

There is a well-known relationship between minimal models and logical consequence, see Osorio *et al.* (2004). The following proposition is a direct consequence of such relationship. Let S be a set of well-formed formulae, then we define SetToFormula(S) $= \bigwedge_{c \in S} c$.

Proposition 3

Let $AF = \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. S is a preferred extension of AF iff compl(S) is a model of $\alpha(AF)$ and $\alpha(AF) \wedge$ SetToFormula($\widetilde{\neg \text{compl}(S)}$) $\models$ SetToFormula($compl(S)$).

Proof

See appendix. □

There are several well-known approaches for inferring minimal models from a propositional formula (Dimopoulos and Torres 1996; Ben-Eliyahu-Zohary 2005). For instance, it is possible to use UNSAT's algorithms for inferring minimal models. Hence, it is clear that we can use UNSAT's algorithms for computing the preferred extensions of an argumentation framework. This idea is formalized with the following proposition.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

Theorem 2
Let $AF = \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. S is a preferred extension of AF if and only if compl(S) is a model of $\alpha(AF)$ and $\alpha(AF) \wedge$ SetToFormula($\widetilde{\neg \text{compl}(S)}$) $\wedge$ ¬SetToFormula(compl(S)) is unsatisfiable.

Proof
Directly, by Proposition 3. □

In order to illustrate Theorem 2, let us consider again the argumentation framework AF of Example 2. Let $S = \{a\}$, then compl(S) $= \{d(b), d(c)\}$. We have already seen that $\{d(b), d(c)\}$ is a model of $\alpha(AF)$, hence the formula to verify its unsatisfiability is

$$(d(b) \leftarrow \neg d(a)) \wedge (d(b) \leftarrow \top) \wedge (d(c) \leftarrow \neg d(b)) \wedge (d(c) \leftarrow d(a)) \wedge \neg d(a) \wedge (\neg d(b) \vee \neg d(c)).$$

However, this formula is satisfiable by the model $\{d(b)\}$, then $\{a\}$ is not a preferred extension. Now let $S = \{a, c\}$, then compl(S) $= \{d(b)\}$. As seen before, $\{d(b)\}$ is also a model of $\alpha(AF)$, hence the formula to verify its unsatisfiability is

$$(d(b) \leftarrow \neg d(a)) \wedge (d(b) \leftarrow \top) \wedge (d(c) \leftarrow \neg d(b)) \wedge (d(c) \leftarrow d(a)) \wedge \neg d(a) \wedge \neg d(c) \wedge \neg d(b).$$

It is easy to see that this formula is unsatisfiable, therefore $\{a, c\}$ is a preferred extension.

The relevance of Theorem 2 is that UNSAT is the prototypical and best-researched co-NP-complete problem. Hence, Theorem 2 opens the possibilities for using a wide variety of algorithms for inferring the preferred semantics.

4 Preferred extensions and general programs

We have seen that the minimal models of $\alpha(AF)$ characterize the preferred extensions of AF. One interesting point of $\alpha(AF)$ is that $\alpha(AF)$ is logically equivalent to the positive disjunctive logic program Γ_{AF} (defined below). It is well known that given a positive disjunctive logic program P, all the minimal models of P correspond to the stable models of P. This property will be enough for characterizing the preferred semantics by the stable models of the positive disjunctive logic program Γ_{AF}.

We start this section by defining a mapping function which is a variation of the mapping of Definition 7.

Definition 8
Let $AF = \langle AR, attacks \rangle$ be an argumentation framework and $a \in AR$. We define the transformation function $\Gamma(a)$ as follows:

$$\Gamma(a) = \{ \bigcup_{b:(b,a) \in attacks} \{d(a) \vee d(b)\}\} \cup \{ \bigcup_{b:(b,a) \in attacks} \{d(a) \leftarrow \bigwedge_{c:(c,b) \in attacks} d(c)\}\}.$$

Now we define the function Γ in terms of an argumentation framework.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

Definition 9
Let $AF = \langle AR, attacks \rangle$ be an argumentation framework. We define its associated general program as follows:

$$\Gamma_{AF} = \bigcup_{a \in AR} \Gamma(a).$$

Remark 2
Notice that $\alpha(AF)$ (see Definition 7) is similar to Γ_{AF}. The main syntactic difference of Γ_{AF} w.r.t. $\alpha(AF)$ is the first part of Γ_{AF} which is $(\bigwedge_{b:(b,a)\in attacks}(d(a) \vee d(b)))$; however, this part is logically equivalent to the first part of $\alpha(AF)$ which is $(\bigwedge_{b:(b,a)\in attacks} d(a) \leftarrow \neg d(b))$. In fact, the main difference is their behavior w.r.t. stable model semantics. In order to illustrate this difference, let us consider the argumentation framework $AF = \langle \{a\}, \{(a, a)\} \rangle$. We can see that

$$\Gamma_{AF} = \{d(a) \vee d(a)\} \cup \{d(a) \leftarrow d(a)\}$$

and

$$\alpha(AF) = (d(a) \leftarrow \neg d(a)) \wedge (d(a) \leftarrow d(a)).$$

It is clear that both formulae have a minimal model which is $\{d(a)\}$[4]; however $\alpha(AF)$ has no stable models. This suggests that $\alpha(AF)$ is not a suitable representation for characterizing preferred extensions by using stable models. Nonetheless, we will see that the stable models of Γ_{AF} characterize the preferred extensions of AF.

Even though, in this paper we are only interested in the preferred semantics, it is worth mentioning that the stable models of the first part of the formula $\alpha(AF)$ i.e., $(\bigwedge_{b:(b,a)\in attacks} d(a) \leftarrow \neg d(b))$, characterize the so-called stable semantics in argumentation theory (Dung 1995). It is also important to point out that $\alpha(AF)$ and Γ_{AF} have different use. On the one hand, we will see that Γ_{AF} is a suitable mapping for inferring preferred extensions by using stable model solvers. On the other hand, $\alpha(AF)$ has shown to be most suitable for studying abstract argumentation semantics. For example, in Nieves *et al.* (2006), $\alpha(AF)$ was used for defining an extension of the preferred semantics. Also, since the well-founded model of $\alpha(AF)$ characterizes the grounded semantics of AF, $\alpha(AF)$ was used for defining extensions of the grounded semantics and to describe the interaction of arguments based on reasoning under the grounded semantics (Nieves *et al.* 2008).

In the following theorem, we formalize a characterization of the preferred semantics in terms of positive disjunctive logic programs and stable model semantics.

Theorem 3
Let $AF = \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. S is a preferred extension of AF iff compl(S) is a stable model of Γ_{AF}.

Proof
See appendix. □

Let us consider the following example.

[4] Notice that $\{d(a)\}$ suggests that AF has a preferred extensions, which is $\{\}$.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

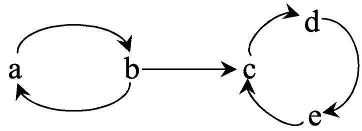

Fig. 2. Graph representation of the argumentation framework $AF = \langle\{a, b, c, d, e\}, \{(a, b), (b, a), (b, c), (c, d), (d, e), (e, c)\}$.

.

Example 3
Let AF be the argumentation framework of Figure 2. We can see that Γ_{AF} is

$$
\begin{array}{ll}
d(a) \vee d(b). & d(a) \leftarrow d(a). \\
d(b) \vee d(a). & d(b) \leftarrow d(b). \\
d(c) \vee d(b). & d(c) \vee d(e). \\
d(c) \leftarrow d(a). & d(c) \leftarrow d(d). \\
d(d) \vee d(c). & d(d) \leftarrow d(b), d(e). \\
d(e) \vee d(d). & d(e) \leftarrow d(c).
\end{array}
$$

Γ_{AF} has two stable models which are $\{d(a), d(c), d(e)\}$ and $\{d(b), d(c), d(e), d(d))\}$, therefore $\{b, d\}$ and $\{a\}$ are the preferred extensions of AF.

4.1 Default negation

As we have commented in whole paper, ours mappings are inspired by two basic conditions that make an argument to be defeated. One of the advantages of characterizing the preferred semantics by using a logic programming semantics with *default negation*, is that we can infer the acceptable arguments from the stable models of Γ_{AF} in a straightforward form. For instance, let Λ_{AF} be the disjunctive logic program Γ_{AF} of Example 3 plus the following clauses:

$$
\begin{array}{ll}
a \leftarrow \neg d(a). & b \leftarrow \neg d(b). \\
c \leftarrow \neg d(c). & d \leftarrow \neg d(d). \\
e \leftarrow \neg d(e), &
\end{array}
$$

such that the intended meaning of each clause is: the argument x is acceptable if it is not defeated. Λ_{AF} has two stable models which are $\{d(a), d(c), d(e), b, d\}$ and $\{d(b), d(c), d(e), d(d), a\}$. By taking the intersection of each model of Λ_{AF} with AR (the set of arguments of AF), we can see that $\{b, d\}$ and $\{a\}$ are the preferred extensions of AF. This idea is formalized by Proposition 4 below.

Definition 10
Let $AF = \langle AR, attacks\rangle$ be an argumentation framework. We define its associated general program as follows:

$$\Lambda_{AF} = \bigcup_{a \in AR} \{\Gamma(a) \cup \{a \leftarrow \neg d(a)\}\}.$$

Notice that $\Gamma(a)$ and $\Lambda(a)$ are equivalent, the main difference between Γ_{AF} and Λ_{AF} is the rule $a \leftarrow \neg d(a)$ for each argument.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

Proposition 4
Let $AF = \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. S is a preferred extension of AF iff there is a stable model M of Λ_{AF} such that $S = M \cap AR$.

Proof
The proof is straightforward from Theorem 3 and the semantics of default negation.
□

It is worth mentioning that by using the disjunctive logic program Λ_{AF} and the DLV System, we can perform any query w.r.t. *sceptical and credulous reasoning*. For instance let `gamma-AF` be the file that contains Λ_{AF} such that AF is the argumentation framework of Figure 2. Let us suppose we want to know if the argument a belongs to some preferred extension of AF. Hence, let `query-1` be the file:
a?
Let us call DLV with the *brave/credulous reasoning* front-end and `query-1`:
`$ dlv -brave gamma-AF query-1`
a `is bravely true, evidenced by` $\{d(b), d(c), d(e), d(d), a\}$
This means that it is true that the argument a belongs to a preferred extension and even more, we have a preferred extension which contains the argument a. Now let us suppose that we want to know if the argument a belongs to all the preferred extensions of AF. Let us call DLV with the *cautious/sceptical reasoning* front-end and `query-1`:
`$ dlv -cautious gamma-AF query-1`
a `is cautiously false, evidenced by` $\{d(a), d(c), d(e), b, d\}$
This means that it is false that the argument a belongs to all the preferred extensions of AF. In fact, we have a counterexample.

5 Conclusions

Since Dung introduced his abstract argumentation approach, he proved that his approach can be regarded as a special form of logic programming with *negation as failure*. In fact, he showed that the grounded and stable semantics can be characterized by the well-founded and stable models semantics, respectively. This result is important because it defined a general method for generating metainterpreters for argumentation systems (Dung 1995). Concerning this issue, Dung did not give any characterization of the preferred semantics in terms of logic programming semantics. It is worth mentioning that according to the literature (Dung 1995; Pollock 1995; Bondarenko *et al.* 1997; Prakken and Vreeswijk 2002; ASPIC:Project 2005), the preferred semantics is regarded as one of the most satisfactory argumentation semantics of Dung's argumentation approach.

In this paper, we characterize the preferred semantics in terms of minimal models (see Theorem 1) and stable model semantics (see Theorem 3). These characterizations are based on two mapping functions that construct a propositional formula and a disjunctive logic program, respectively. These characterizations have as main result the definition of a direct relationship between one of the most satisfactory argumentation semantics and may be the most successful approach of nonmonotonic reasoning of the last two decades i.e., logic programming with the stable model semantics. Based on this fact, we introduce a novel and easy-to-use method for implementing argumentation systems that are based on the preferred semantics. It is quite obvious that our method will take advantage of the platform that has been developed under stable model semantics for generating argumentation systems. For instance, we can implement the preferred semantics inside object-oriented programs based on our characterization (Theorem 3, Proposition 4) and the DLV JAVA Wrapper (Ricca 2003).

We can see that our approach falls in the family of the model-checking methods for inferring the preferred semantics. In fact, our approach is closely related to the methods suggested in Besnard and Doutre (2004) and Egly and Woltran (2006). As seen in Theorem 1, our propositional formula $\alpha(AF)$ is closely related to one of the propositional formulae (see Proposition 2) which were suggested in Besnard and Doutre (2004). It is worth mentioning that the propositional formula suggested by Egly and Woltran (2006) for inferring the admissible sets of an argumentation framework is the same as the propositional formula of Proposition 2. The main difference between the approaches suggested by Besnard and Doutre (2004) and Egly and Woltran (2006) and our approach is the strategy for inferring the models of a propositional formula. Instead of using *maximal models* for characterizing the preferred semantics as it is done by (Besnard and Doutre 2004), we are using *minimal models/stable models*. Hence, we can use any system which could compute minimal models/stable models of a propositional formula. Maximality in Egly and Woltran's approach is checked on the object level, i.e. within the resulting quantified Boolean formula (QBF).

An interesting property of our approach is that whenever we use stable model solvers for computing the preferred extensions of an argumentation framework, we can compute all the preferred extensions in full. In decision-making systems, it is not strange to require all the possible coherent points of view (preferred extensions) in a dispute between arguments. For instance, in the medical domain when a doctor has to give a diagnosis under incomplete information, he has to consider all the possible alternatives in his decisions (Cortés *et al.* 2005; Tolchinsky *et al.* 2005).

Acknowledgement

We are grateful to anonymous referees for their useful comments. J.C. Nieves thanks to CONACyT for his PhD Grant. J.C. Nieves and U. Cortés would like to acknowledge support from the EC-funded project SHARE-it: Supported Human Autonomy for Recovery and Enhancement of cognitive and motor abilities using information technologies (FP6-IST-045088). The views expressed in this paper are not necessarily those of the SHARE-it consortium.

References

ASPIC: Project. 2005. *Deliverable D2.2:Formal semantics for inference and decision-making*. Argumentation Service Plarform with Integrated Components.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

ASPIC: Project. 2006. ASPIC: Argumentation engine demo. URL: http://aspic.acl.icnet.uk/.

Ben-Eliyahu-Zohary, R. 2005. An incremental algorithm for generating all minimal models. *Artificial Intelligence 169,* 1, 1–22.

Bench-Capon, T. 2002. Value-based argumentation frameworks. In *Proceedings of Non Monotonic Reasoning*, 444–453.

Besnard, P. and Doutre, S. 2004. Checking the acceptability of a set of arguments. In *Tenth International Workshop on Non-Monotonic Reasoning (NMR 2004),* 59–64.

Bondarenko, A., Dung, P. M., Kowalski, R. A. and Toni, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence 93*, 63–101.

Cayrol, C., Doutre, S. and Mengin, J. 2003. On decision problems related to the preferred semantics for argumentation frameworks. *Journal of Logic and Computation 13,* 3, 377–403.

Cortés, U., Tolchinsky, P., Nieves, J. C., López-Navidad, A. and Caballero, F. 2005. Arguing the discard of organs for transplantation in CARREL. In *CATAI 2005*, 93–105.

Dimopoulos, Y. and Torres, A. 1996. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science 170*, 1–2, 209–244.

DLV, S. 1996. Vienna University of Technology. URL: http://www.dbai.tuwien.ac.at/proj/dlv/.

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

Dung, P. M., Kowalski, R. A. and Toni, F. 2006. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence 170,* 2, 114–159.

Dung, P. M., Mancarella, P. and Toni, F. 2007. Computing ideal sceptical argumentation. *Artificial Intelligence 171,* 10–15, 642–674.

Dunne, P. E. and Bench-Capon, T. J. M. 2004. Complexity in value-based argument systems. In *JELIA*. Lecture Notes in Computer Science, vol. 3229. Springer, Berlin, 360–371.

Egly, U. and Woltran, S. 2006. Reasoning in argumentation frameworks using quantified Boolean formulas. In *Proceedings of COMMA*, P. E. Dunne and T. J. Bench-Capon, Eds. Frontiers in Artificial Intelligence and Applications, vol. 144. IOS Press, Amsterdam, 133–144.

Gaertner, D. and Toni, F. 2007. CaSAPI: A system for credulous and sceptical argumentation. In *Argumentation and Non-Monotonic Reasoning (LPNMR-07 Workshop)*, G. Simari and P. Torroni, Eds. Arizona, USA, 80–95.

Gebser, M., Liu, L., Namasivayam, G., Neumann, A., Schaub, T. and Truszczynski, M. 2007. The first answer set programming system competition. In *Ninth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-07)*, G. B. Chitta Baral and J. Schlipf, Eds. Lecture Notes in Artificial Intelligence, vol. 4483. Springer-Verlag, Berlin, 3–17.

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

Gelfond, M. and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing 9*, 365–385.

Leone, N., Pfeifer, G., Faber, W., Calimeri, F., Dell'Armi, T., Eiter, T., Gottlob, G., Ianni, G., Ielpa, G., Koch, C., Perri, S. and Polleres, A. 2002. The dlv system. In *JELIA,* 537–540.

Nieves, J. C., Osorio, M., Cortés, U., Olmos, I. and Gonzalez, J. A. 2006. Defining new argumentation-based semantics by minimal models. In *Seventh Mexican International Conference on Computer Science (ENC 2006)*. IEEE Computer Science Press, Los Alamitos, California, 210–220.

Nieves, J. C., Osorio, M., and Cortés, U. 2008. *Studying the Grounded Semantics by Using a Suitable Codification*. Research report LSI-08-6-R, Universitat Politècnica de Catalunya, Software Department (LSI), Barcelona, Spain.

Osorio, M., Navarro, J. A. and Arrazola, J. 2004. Applications of intuitionistic logic in answer set programming. *Theory and Practice of Logic Programming (TPLP) 4,* 3, 225–354.

Pollock, J. L. 1995. *Cognitive Carpentry: A Blueprint for How to Build a Person*. MIT Press, Cambridge, MA.

Prakken, H. and Vreeswijk, G. A. W. 2002. Logics for defeasible argumentation. In *Handbook of Philosophical Logic*, 2nd ed., Vol. 4, D. Gabbay and F. Günthner, Eds. Kluwer Academic Publishers, Dordrecht–Boston–London, 219–318.

Ricca, F. 2003. The dlv java wrapper. In *2003 Joint Conference on Declarative Programming, AGP-2003*, Reggio Calabria, Italy, 3–5 September 2003, 263–274.

Tolchinsky, P., Cortés, U., Nieves, J. C., López-Navidad, A. and Caballero, F. 2005. Using arguing agents to increase the human organ pool for transplantation. In *Proceedings of the Third Workshop on Agents Applied in Health Care (IJCAI 2005)*.

van Dalen, D. 1994. *Logic and Structure*, 3rd augmented ed. Springer-Verlag, Berlin.

Appendix

Proof of Proposition 1

Proof

First of all, we present the following two observations.

(1) Given $M_1, M_2 \subseteq \mathscr{L}_T$, it is true that $M_1 \subset M_2$ iff $f(\mathscr{L}_T \backslash M_2) \subset f(\mathscr{L}_T \backslash M_1)$.

(2) Given a propositional formula A, an interpretation M from $\mathscr{L}_T$ to $\{0, 1\}$ and $x \in \{0, 1\}$. Then it is not difficult to prove by induction on A's length[5] that $M(A) = x$ iff $f(\mathscr{L}_T \backslash M)(g(A)) = x$.

=> To prove that if M is a maximal model of T, then $f(\mathscr{L}_T \backslash M)$ is a minimal model of $g(T)$. The proof is by contradiction. Let us suppose that M is a maximal model of T, but $f(\mathscr{L}_T \backslash M)$ is a model of $g(T)$ and is not minimal. Then if $f(\mathscr{L}_T \backslash M)$ is not minimal, then there exists M_2 such that $f(\mathscr{L}_T \backslash M_2)$ is a model of $g(T)$ and $f(\mathscr{L}_T \backslash M_2) \subset f(\mathscr{L}_T \backslash M)$. Then by the second observation, if $f(\mathscr{L}_T \backslash M_2)$ is a model of $g(T)$, then M_2 is a model of T. By the first observation, if $f(\mathscr{L}_T \backslash M_2) \subset f(\mathscr{L}_T \backslash M)$ then $M \subset M_2$. But this is a contradiction because M is a maximal model of T.

<= To prove that if $f(\mathscr{L}_T \backslash M)$ is a minimal model of $g(T)$, then M is a maximal model of T. The proof is also by contradiction. Let us suppose that $f(\mathscr{L}_T \backslash M)$ is a minimal model of $g(T)$, but M is model of T and is not maximal. If M is not maximal, then there exists a model M_2 of T such that $M \subset M_2$. Then by the second observation, if M_2 is a model of T then $f(\mathscr{L}_T \backslash M_2)$ is a model of $g(T)$. By the first observation, if $M \subset M_2$ then $f(\mathscr{L}_T \backslash M_2) \subset f(\mathscr{L}_T \backslash M)$. But this is a contradiction because $f(\mathscr{L}_T \backslash M)$ is a minimal model of $g(T)$. □

[5] Since A is a disjunctive clause, the length of A is given by the number of atoms in the head of A plus the number of literals in the body of A.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

Proof of Theorem 1

Proof

We present the following two observations.

(1) Since the mapping $f(x)$ corresponds to $d(x)$, then compl$(S) = f(AR \backslash S)$ because compl$(S) = \{d(a)|a \in AR \backslash S\}$ and $f(AR \backslash S) = \{f(a)|a \in AR \backslash S\}$.

(2) $\alpha(AF)$ is logically equivalent to $g(\beta(AF))$,

$g(\beta(AF)) =$

$$\bigwedge_{a \in AR} ((\neg d(a) \rightarrow \bigwedge_{b:(b,a) \in attacks} d(b)) \wedge (\neg d(a) \rightarrow \bigwedge_{b:(b,a) \in attacks} (\bigvee_{c:(c,b) \in attacks} \neg d(c)))).$$

Since $a \rightarrow \bigwedge_{b \in S} b \equiv \bigwedge_{b \in S} (a \rightarrow b)$, we get

$$\bigwedge_{a \in AR} (\bigwedge_{b:(b,a) \in attacks} (\neg d(a) \rightarrow d(b)) \wedge (\bigwedge_{b:(b,a) \in attacks} (\neg d(a) \rightarrow \bigvee_{c:(c,b) \in attacks} \neg d(c)))).$$

By applying transposition and cancelation of double negation in both implications, we get

$$\bigwedge_{a \in AR} (\bigwedge_{b:(b,a) \in attacks} (\neg d(b) \rightarrow d(a)) \wedge (\bigwedge_{b:(b,a) \in attacks} (\neg \bigvee_{c:(c,b) \in attacks} \neg d(c) \rightarrow d(a)))).$$

Now for the right-hand side of the formula, we need to apply Morgan laws,

$$\bigwedge_{a \in AR} (\bigwedge_{b:(b,a) \in attacks} (\neg d(b) \rightarrow d(a)) \wedge (\bigwedge_{b:(b,a) \in attacks} (\bigwedge_{c:(c,b) \in attacks} d(c) \rightarrow d(a)))).$$

Finally by changing $\rightarrow$ by $\leftarrow$, we get $\alpha(AF)$,

$$\bigwedge_{a \in AR} (\bigwedge_{b:(b,a) \in attacks} (d(a) \leftarrow \neg d(b)) \wedge (\bigwedge_{b:(b,a) \in attacks} (d(a) \leftarrow \bigwedge_{c:(c,b) \in attacks} d(c)))) =$$

$\alpha(AF)$.

Now the main proof: S is a preferred extension of AF iff (by Proposition 2) S is a maximal model of $\beta(AF)$ iff (by Proposition 1) $f(AR \backslash S)$ is a minimal model of $g(\beta(AF))$ iff (by observations 1 and 2) compl(S) is a minimal model of $\alpha(AF)$. □

Proof of Proposition 3

First of all, let us introduce the following relationship between minimal models and logic consequence.

Lemma 1

(Osorio *et al.* 2004). For a given general program P, M is a model of P and $P \cup \neg\widetilde{M}) \models M$ iff M is a minimal model of P.

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press

This lemma was introduced in terms of augmented programs. Since a general program is a particular case of an augmented program, we write the lemma in terms of general programs (see Osorio *et al.* (2004) for more details about augmented programs).

Proof
S is a preferred extension of AF iff (by Theorem 1) compl(S) is a minimal model of $\alpha(AF)$ iff (by lemma 1) compl(S) is a model of $\alpha(AF)$ and $\alpha(AF) \wedge$ SetToFormula($\widehat{\neg \mathrm{compl}(S)}$) $\models$ SetToFormula(compl(S)). □

Proof of Theorem 3

Proof
S is a preferred extension of AF iff compl(S) is a minimal model of $\alpha(AF)$ (by Theorem 1) iff compl(S) is a minimal model of Γ_{AF} (since Γ_{AF} is logically equivalent to $\alpha(AF)$ in classical logic) iff compl(S) is a stable model of Γ_{AF} (since Γ_{AF} is a positive disjunctive logic program and for every positive disjunctive logic program P, M is a stable model of P iff M is a minimal model of P). □

https://doi.org/10.1017/S1471068408003359 Published online by Cambridge University Press