Fundamenta Informaticae 1–15
IOS Press

Semi-Stable and Stage Extensions as 2-valued Models

Mauricio Osorio
Universidad de las Américas - Puebla
Depto. de Actuaría, Física y Matemáticas
Sta. Catarina Mártir, Cholula, Puebla, 72820 México
osoriomauri@gmail.com

Juan Carlos Nieves
Department of Computing Science
Umeå University
SE-901 87, Umeå, Sweden
juan.carlos.nieves@umu.se

Abstract. This paper introduces the characterizations of both semi-stable and stage extensions in terms of 2-valued logical models. To this end, the so-called GL-supported and GL-stage models are defined. These two classes of logical models are logic programming counterparts of the notion of *range* which is a coined concept in argumentation semantics.

1. Introduction

Theoretical argumentation research has been strongly influenced by the abstract argumentation theory of Dung [7]. This approach is mainly oriented towards managing the interaction between arguments.

Argumentation has been regarded as a non-monotonic reasoning approach since it was suggested as an inference reasoning approach [20]. Dung showed that argumentation inference can be regarded as a logic programming inference with *negation as failure* [7]. In his seminal paper [7], Dung introduced four argumentation semantics: *grounded, stable, preferred* and *complete* semantics; moreover, he showed that both the grounded and stable semantics can be regarded as logic programming inferences by considering the well-founded [9] and stable model [10] semantics, respectively. Currently, it is known that the four argumentation semantics introduced in [7] can be regarded as logic programming inferences by using different mappings, from argumentation frameworks into logic programs and different logic programming semantics (see Section 4).

Following Dung's argumentation style, several new argumentation semantics have been proposed. Among them, ideal, semi-stable, stage and CF2 have been deeply explored [2]. Currently we already know that both ideal and CF2 semantics can also be regarded as logic programming inferences by using different mappings from argumentation frameworks into logic programs and different logic programming semantics (see Section 4).

Semi-stable and stable semantics were introduced from different point of views; however, both semi-stable and stage semantics have been defined in terms of the so-called *ranges* of complete extensions and conflict free sets, respectively. Given that the concept of range seems as a fundamental feature for both semi-stable and stage semantics, the following question arises:

[Q1] *How the concept of range can be captured from the point of view of logic programming?*

This question takes relevance in the understanding of argumentation as logic programming.

Against this background, we will extend the results around the characterization of argumentation semantics in terms of mappings from argumentation frameworks into logic programs and logic programming semantics In particular, we will give answer to Q1 by introducing the so-called *GL-supported* and *GL-stage* models which argue for an interpretation of range from a logic programming point of view. We argue that for capturing the idea of range from the logic programming point view, logic programming reductions which have been used for defining logic programming semantics such as stable model [10] and p-stable [17] semantics are important.

We will show that semi-stable semantics can be characterized by GL-supported models and a mapping which has been used for characterizing the grounded, stable, preferred, complete and ideal semantics (see Section 4). The stage semantics is characterized by GL-stage models and a mapping which has been used for characterizing the grounded, stable, preferred, complete, semi-stable and CF2 semantics (see Section 4).

The rest of the paper is structured as follows: In Section 2, basic background about logic programming and argumentation is introduced. In Section 3, our characterizations of semi-stable and stage semantics as 2-valued logical models are introduced. In Section 4, a discussion about related work is introduced. In the last section, an outline of our conclusions and future work is presented.

2. Background

In this section, we start defining the syntax of normal logic programs. After this, the logic programming semantics p-stable and stable model semantics are presented. In the last part of this section, we present some basic concepts of argumentation theory and a pair of mappings from argumentation frameworks into logic programs.

2.1. Logic Programs: Syntax

A signature $\mathcal{L}$ is a finite set of elements that we call atoms. A literal is an atom a (called *a positive literal*), or the negation of an atom $not\ a$ (called *a negative literal*). Given a set of atoms $\{a_1, \ldots, a_n\}$, we write $not\ \{a_1, \ldots, a_n\}$ to denote the set of literals $\{not\ a_1, \ldots, not\ a_n\}$. A normal clause C is of the form:

$$a_0 \leftarrow a_1, \ldots, a_j, not\ a_{j+1}, \ldots, not\ a_n$$

in which a_i is an atom, $0 \leq i \leq n$. When $n = 0$ the normal clause is called a fact and is an abbreviation of $a_0 \leftarrow \top$, where $\top$ is the ever true atom. A normal logic program is a finite set of normal clauses. Sometimes, we denote a clause *C* by $a \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-$, where $\mathcal{B}^+$ contains all the positive body literals and $\mathcal{B}^-$ contains all the negative body literals. We also use $body(C)$ to denote $\mathcal{B}^+,\ not\ \mathcal{B}^-$. When $\mathcal{B}^- = \emptyset$, the clause *C* is called a definite clause. A definite program is a finite set of definite clauses. We denote by $\mathcal{L}_P$ the signature of P, *i.e.* the set of atoms that occurs in P. Given a signature $\mathcal{L}$, we write $Prog_{\mathcal{L}}$ to denote the set of all the programs defined over $\mathcal{L}$.

Logical consequence in classical logic is denoted by $\vdash$. Given a set of proposition symbols S and a theory (a set of well-formed formulae) Γ, $\Gamma \vdash S$ if and only if $\forall s \in S\ \Gamma \vdash s$. When we treat a logic program as a logical theory, each negative literal $not\ a$ is replaced by $\neg a$ such that $\neg$ is regarded as the classical negation in classic logic. Given a normal logic program P, if $M \subseteq \mathcal{L}_P$, we write $P \Vdash M$ when: $P \vdash M$ and M is a classical 2-valued model of P (*i.e.* atoms in M are set to true, and atoms not in M to false; a set of atoms is a classical model of P if the induced interpretation evaluates P to true). We say that a mode M of a program P is minimal iff a model M' of P different from M such that $M' \subset M$ does not exist.

2.2. Logic Programs: Semantics

In this section, we introduce two logic programming semantics: *Stable model* and *P-stable semantics*.

Stable model semantics is one of the most influencing logic programming semantics in the non-monotonic reasoning community and is defined as follows:

Definition 2.1. [10] Let *P* be a normal logic program. For any set $S \subseteq \mathcal{L}_P$, let P^S be the definite logic program obtained from *P* by deleting

(i) each rule that has a formula $not\ l$ in its body with $l \in S$, and then

(ii) all formulæ of the form $not\ l$ in the bodies of the remaining rules.

Hence *S* is a stable model of *P* iff *S* is a minimal model of P^S.

From hereon, whenever we say Gelfond-Lifschitz (GL) reduction, we mean the reduction P^S. As we can observe GL reduction is the core of the stable model semantics.

There is an extension of the stable model semantics which is called *p-stable semantics* [17]. Like stable model semantics, p-stable semantics is defined in terms of a single reduction which is defined as follows:

Definition 2.2. [17] Let P be a normal program and M be a set of literals. We define $RED(P, M) := \{l \leftarrow \mathcal{B}^+,\ not\ (\mathcal{B}^- \cap M) | l \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^- \in P\}$.

As we can see, GL reduction and RED reduction have different behaviors. On the one hand, the output of GL reduction always is a definite program; on the other hand, the output of RED reduction can contain normal clauses.

By considering the RED reduction, the *p-stable* semantics for normal logic programs is defined as follows:

Definition 2.3. [17] Let P be a normal program and M be a set of atoms. We say that M is a p-stable model of P if $RED(P, M) \Vdash M$. $P\text{-}stable(P)$ denotes the set of p-stable models of P.

In general terms, a 2-valued logic programming semantics SEM is a function from $Prog_{\mathcal{L}}$ to $2^{\{0,1\}}$. Hence, given two logic programming semantics SEM_1 and SEM_2, SEM_1 is *stronger than* SEM_2 iff $SEM_1 \subseteq SEM_2$. Let us observe that the relation *stronger than*, between logic programming semantics, is basically an order between logic programming semantics.

2.3. Argumentation theory

In this section, we introduce the definition of some argumentation semantics mainly *stable, preferred, complete, semi-stable and stage semantics*. To this end, we start by introducing the basic structure of an argumentation framework.

Definition 2.4. [7] 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* $\subseteq AR \times AR$.

We say that *a attacks b* (or b is attacked by a) if $attacks(a, b)$ holds. Similarly, we say that a set S of arguments attacks b (or b is attacked by S) if b is attacked by an argument in S.

Let us observe that an argumentation framework is a simple structure which captures the conflicts of a given set of arguments. In order to select *coherent points of view* from a set of conflicts of arguments, Dung introduced a set of *patterns of selection of arguments*. These patterns of selection of arguments were called *argumentation semantics*. Dung defined his argumentation semantics based on the basic concept of *admissible set*:

Definition 2.5. [7]

- 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*.
- An argument $a \in AR$ is acceptable with respect to a set S of arguments if and only if for each argument $b \in AR$: If b attacks a then b is attacked by S.
- A conflict-free set of arguments S is admissible if and only if each argument in S is acceptable *w.r.t.* S.

Before defining some argumentation semantics, let us introduce some notation. Let $AF := \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. $S^+ = \{b | a \in S \text{ and } (a, b) \in attacks\}$.

Definition 2.6. [7] Let $AF := \langle AR, attacks \rangle$ be an argumentation framework. An admissible set of argument $S \subseteq AR$ is:

- *stable* if and only if S attacks each argument which does not belong to S.
- *preferred* if and only if S is a maximal (*w.r.t.* inclusion) admissible set of AF .
- *complete* if and only if each argument, which is acceptable with respect to S, belongs to S.

- *semi-stable* if and only if S is a complete extension such that $S \cup S^+$ is maximal.

In addition to the argumentation semantics based on admissible sets; in the state of art, there are other approaches for defining argumentation semantics [2]. One of these approaches is the approach based on *stage semantics* which is an argumentation semantics based on conflict-free sets [23]. Stage semantics is defined as follows:

Definition 2.7. Let $AF := \langle AR, attacks \rangle$ be an argumentation framework. E is a stage extension iff E is a conflict free set and $E \cup E^+$ is maximal *w.r.t.* set inclusion.

Before moving on, let us observe that both semi-stable and stage semantics are based on the so-called *range* which is defined as follows: If E is a set of arguments, then $E \cup E^+$ is called range. According to the literature, the notion of range was first introduced by Verheij [23].

In Section 3, we will see that the notion of range can be regarded as a logical model. Indeed, we will see that the notion of range will play an important role for characterizing both semi-table and stage semantics as 2-valued logical models.

2.4. Mappings from argumentation frameworks into normal programs

The first requirement for studying the structure of an argumentation framework as a logic program is to manage an argumentation framework as a logic program. To this end, a pair of mappings from an argumentation framework into a logic programs will be presented. Let us observe that these mappings are basically *declarative representations* of an argumentation framework. In particular, these mappings are based on the ideas of *conflictfreeness* and *reinstatement* which are the basic concepts behind the definition of admissible sets.

In these mappings, the predicate $def(x)$ is used, with the intended meaning of $def(x)$ being "x is a defeated argument". A pair of transformation functions *w.r.t.* an argument is defined as follows.

Definition 2.8. Let $AF := \langle AR, attacks \rangle$ be an argumentation framework and $a \in AR$. We define a pair of transformation functions:

$$\Pi^-(a) = \bigcup_{b:(b,a) \in attacks} \{def(a) \leftarrow \ not\ def(b)\}$$

$$\Pi^+(a) = \bigcup_{b:(b,a) \in attacks} \{def(a) \leftarrow \bigwedge_{c:(c,b) \in attacks} def(c)\}$$

Let us observe that $\Pi^-(a)$ suggests that an argument a is defeated when anyone of the arguments which attack a is not defeated. $\Pi^+(a)$ suggests that an argument a is defeated when all the arguments that defends[1] a are defeated. Moreover, if a given argument a has no attacks, then $\Pi^-(a) = \{\}$ and $\Pi^+(a) = \{\}$. This situation happens because an argument that has no attacks is an acceptable argument which means that it belongs to all admissible sets of AF.

By considering $\Pi^-(a)$ and $\Pi^+(a)$, a couple of mappings from argumentation frameworks into logic programs are introduced.

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

Definition 2.9. Let $AF := \langle AR, attacks \rangle$ be an argumentation framework. We define their associated normal programs as follows:

$$\Pi^{-}_{AF} := \bigcup_{a \in AR} \{\Pi^{-}(a)\}$$

$$\Pi_{AF} := \Pi^{-}_{AF} \cup \bigcup_{a \in AR} \{\Pi(a)^{+}\}$$

In Definition 2.9, two mappings from argumentation frameworks into logic programs are presented. It is obvious that Π^{-}_{AF} is a subset of Π_{AF}. However, each mapping is capturing different concepts: Π^{-}_{AF} is basically a declarative specification of the idea of conflictfreeness and Π_{AF} is basically a declarative specification of the ideas of conflictfreeness and reinstatement. Indeed, one can see that the 2-valued logical models of Π^{-}_{AF} characterize the conflict free sets of AF and the 2-valued logical models of Π_{AF} characterize the admissible sets of AF.

Before moving on, we want to point out that Π_{AF} has been shown to be a flexible mapping for studying argumentation theory as logic programming with negation as failure. Until now, the grounded, stable, preferred, complete and ideal semantics have been characterized by, well-acceptable logic programming semantics such as, the well-founded, stable, p-stable, Clark's completion and well-founded^{+} semantics, respectively [5, 14, 13]. Moreover Π^{-}_{AF}, which contains basically the clauses with negation as failure of Π_{AF}, has been used for characterizing argumentation semantics such as ground, stable, preferred, complete, semi-table and CF2 [7, 15, 21][2].

In the following section, we will extend the results of both Π_{AF} and Π^{-}_{AF}. More accurately, we will show that Π_{AF} is able to characterize the semi-stable semantics; moreover, we will show that Π^{-}_{AF} is able to capture the stage semantics.

3. Semi-stable and Stage extensions as 2-valued models

In this section, new results about the relationship between argumentation semantics and logic programming semantics are introduced. In particular, we observe that both semi-stable and stage extension can be characterized by 2-valued logical models.

3.1. Semi-Stable Semantics

We start presenting our results *w.r.t.* semi-stable semantics. To this end, let us start defining the concept of a *supported model*; moreover, we will introduce some notation. Let P be a logic program and M be a 2-valued model of P. M is *a supported model* of P iff for each $a \in M$, there is $a_0 \leftarrow \mathcal{B}^{+},\ not\ \mathcal{B}^{-} \in P$ such that $a = a_0$, $\mathcal{B}^{+} \in M$ and $\mathcal{B}^{-} \cap M = \emptyset$. $Facts(P) = \{a | a \leftarrow \top \in P\}$.

In order to regard 2-valued models as set of arguments, let us introduces the following functions: Let $E_M = \{x | def(x) \in \mathcal{L}_{\Pi_{AF}} \setminus M\}$ and $E^{+}_{M} = \{x | def(x) \in Facts(\Pi^{M}_{AF})\}$ such that $M \subseteq \mathcal{L}_{\Pi_{AF}}$. Informally speaking, both E_M and E^{+}_{M} consider a set of atoms M as input and return a set of arguments.

As we saw in Definition 2.6, semi-stable extensions are defined in terms of complete extensions. Moreover, it has been shown that the supported models of Π_{AF} characterize the complete extensions of

[2]The mapping introduced by Dung [7] does not really match with Π^{-}_{AF}; however, it is easy to see that Π^{-}_{AF} is equivalent to the mapping introduced by Dung

a given argumentation framework AF [19]. By having in mind this result, we introduce the concept of *GL-supported-model*.

Definition 3.1. Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework and M be a supported model of Π_{AF}. M is a GL-supported-model of Π_{AF} iff $Facts(\Pi_{AF}^{M}) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion.

There are some important observations with respect to Definition 3.1 which we want to point out:

1. M is a supported model of Π_{AF} iff E_M is a complete extension of AF.
2. M is a supported model of Π_{AF} iff $E_M^+ = E^+$ such that E is a complete extension of AF.
3. M is a supported model of Π_{AF} iff $E_M \cup E_M^+$ is a *range* with respect to the complete extension E_M.

From these observations, we can clearly see that the definition of a GL-supported model is similar to the definition of a semi-stable extension. One of the main constructions of the definition of a GL-supported model is: $Facts(\Pi_{AF}^{M})$. This part of the construction of a GL-supported model is basically characterizing a set E^+ where E is a complete extension. We can see that the GL reduction is quite important for this construction. As we saw in Definition 2.1, GL reduction is the core of the definition of stable models.

We want to point out that the definition of GL-supported models can also be based on the RED reduction which is the reduction used for defining p-stable models (see Definition 2.3). This similarity between RED and GL reductions argues that both RED and GL reductions can play an important role for capturing the idea of range of an argumentation framework from a logic programming point of view. As we will see in the following theorem, GL-supported models characterize semi-stable extensions; hence, both RED and GL reductions play an important role for capturing semi-stable extension as 2-valued logical models.

Theorem 3.1. Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework. M is a GL-supported model of Π_{AF} if and only if $AR \setminus \{a | def(a) \in M\}$ is a semi-stable extension of AF.

Proof:
Let us start introducing some observations:

1. E is a complete extension of AF iff $E = \{x | def(x) \in \mathcal{L}_{\Pi_{AF}} \setminus M$ and M is a supported model of $\Pi_{AF}\}$ [19].
2. If E is a complete extension of AF, then $E^+ = \{x | def(x) \in Facts(RED(\Pi_{AF}, M))$ and M is a supported model of $\Pi_{AF}\}$.

=> If M is a GL-supported model of Π_{AF} then $Facts(\Pi_{AF}^{M}) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion and M is a supported model. By Observation 1, E_M is a complete extension of AF such that M is a supported model. By observation 2, $E_M^+ = E^+$ such that M is a supported model and E is a complete extension. Hence, $E_M \cup E_M^+$ is a semi-stable extension of AF.

<= Similar to the previous case.

□

An interesting property of GL-supported models is that they can be characterized by both p-stable and 2-valued models.

Proposition 3.1. Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework.

1. M is a GL-supported model of Π_{AF} iff $Facts(\Pi_{AF}^{M}) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion such that M is a 2-valued model of Π_{AF}.

2. M is a GL-supported model of Π_{AF} iff $Facts(\Pi_{AF}^{M}) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion such that M is a p-stable model Π_{AF}.

Proof:
We start introducing the following observations from the state of the art:

1. S is a 2-valued model of Π_{AF} iff $AR \setminus \{x | def(x) \in S\}$ is an admissible extension of AF [13].

2. According to Proposition 4 from [4] the following statements are equivalent:

   (a) E is a complete extension such that $E \cup E^{+}$ is maximal (*w.r.t.* set inclusion).

   (b) E is an admissible set such that $E \cup E^{+}$ is maximal (*w.r.t.* set inclusion).

3. S is a p-stable model of Π_{AF} iff $AR \setminus \{x | def(x) \in S\}$ is a preferred extension of AF [5].

Now let us prove each of the points of the proposition:

1. M is a GL-supported model of Π_{AF} iff $Facts(\Pi_{AF}^{M}) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion and M is a supported model. By Theorem 3.1, $Facts(\Pi_{AF}^{M}) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion and M is a supported model iff $E_M \cup E_M^{+}$ is maximal and E_M is a complete extension of AF. By Observation 2, $E_M \cup E_M^{+}$ is maximal and E_M is a complete extension of AF iff $E_M \cup E_M^{+}$ is maximal and E_M is an admissible extension of AF. Hence, the result follows by Observation 1 which argues that any 2-valued model of Π_{AF} characterizes an admissible set of AF.

2. Let us start by observing that semi-stable extensions can be characterized by preferred extension with maximal range which mean: E is a semi-stable extension iff $E \cup E^{+}$ is maximal (*w.r.t.* set inclusion) and E is a preferred extension (see Proposition 13 from [2]). Hence, the result follows by Observation 3 and Theorem 3.1.

□

A direct consequence of Proposition 3.1 and Theorem 3.1 is the following corollary which introduces a pair of characterizations of semi-stable extensions as 2-valued models and p-stable models of Π_{AF}.

Corollary 3.1. Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework.

1. Let M be a p-stable model of Π_{AF}. $AR \setminus \{a | def(a) \in M\}$ is a semi-stable extension of AF iff $Facts(\Pi_{AF}^{M}) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion.

2. Let M be a 2-valued model of Π_{AF}. $AR \setminus \{a|def(a) \in M\}$ is a semi-stable extension of AF iff $Facts(\Pi_{AF}^{M}) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion.

Observing Corollary 3.1, we can see that there is an interval of logic programming semantics which characterize semi-stable extensions. This interval of logic programming is defined by the order relation of *stronger than* between logic programming semantics. This result is formalized by the following corollary. To this end, let us introduce the following notation, let P be a logic program, $2SEM(P)$ denotes the 2-valued models of P.

Corollary 3.2. Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework and SEM be a logic programming semantics such that SEM is stronger than $2SEM$ and $P\text{-}stable$ is stronger than SEM. If $M \in SEM(\Pi_{AF})$, then $AR \setminus \{a|def(a) \in M\}$ is a semi-stable extension of AF iff $Facts(\Pi_{AF}^{M}) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion.

Given the relation of semi-stable extensions with the stable and preferred extensions, we can observe some relations between GL-supported models *w.r.t.* the stable model semantics [10] and p-stable semantics.

Proposition 3.2. Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework.

1. If M is a stable model of Π_{AF} then M is a GL-supported model of Π_{AF}.

2. If M is a GL-supported model of Π_{AF} then M is a p-stable model of Π_{AF}.

Proof:

1. It follows from Theorem 3.1 and Theorem 2 from [4].

2. It follows from Theorem 3.1 and Theorem 3 from [4].

□

Given a logic program P, $Stable(P)$ denotes the set of stable models of P and $GLModels(P)$ denotes the GL-supported models of P.

Proposition 3.3. Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework such that $Stable(\Pi_{AF}) \neq \emptyset$. $Stable(\Pi_{AF}) = GLModels(\Pi_{AF})$.

Proof:
We know that E is a stable extension of AF iff $E = AR \setminus \{x|def(x) \in M$ and M is a stable model $\Pi_{AF}\}$ (Theorem 5 from [5]). Hence, the result follows from Theorem 3.1 and Theorem 5 from [4]. □

3.2. Stage Semantics

We have seen that the idea of range with respect to complete extensions can be captured in terms of logical models by considering logic programming reductions such as *GL* and *RED*. The idea of range was originally introduced by Verheij [23]. Indeed, by using the idea of range, Verheij introduced the so-called *stage semantics* (see Definition 2.7). Unlike semi-stable semantics which is based on admissible sets (*e.g.,* complete extensions), stage semantics is based on conflict-free sets. However, one can observe that semi-stable and stage semantics share similar constructions (see Definition 2.6 and Definition 2.7). Hence the following question arises:

> [Q2] *Can the reduction* GL *(or RED) characterize ranges w.r.t. conflict-free sets in order to characterize stage extensions as logical models?*

In order to give answer the previous question, let us remember that in Section 2.4, we introduced a couple of mappings: Π^-_{AF} and Π_{AF}. We have observed that Π^-_{AF} is basically a declarative specification of conflict-free sets and Π_{AF} is a declarative specification of admissible sets.

Given that stage semantics is based on conflict free sets, we will consider Π^-_{AF} for exploring stage extensions as logical models. In the following definition, we introduce the class of GL-stage models which are based on Π^-_{AF} .

Definition 3.2. Let $AF = \langle AR, Attacks\rangle$ be an argumentation framework and M be a 2-valued model of Π^-_{AF}. M is a GL-stage model of Π_{AF} iff $Facts((\Pi^-_{AF})^M) \cup \{\mathcal{L}_{\Pi_{AF}} \setminus M\}$ is maximal *w.r.t.* set inclusion.

Like GL-supported models, GL-stage models are based on the reduction GL; however, once again, one can use RED reduction for defining GL-stage models. Moreover, we can observe that:

1. M is a 2-valued model of Π^-_{AF} iff E_M is a conflict-free set.
2. M is a 2-valued model of Π^-_{AF} iff $E^+_M = E^+$ such that E^+ is a conflict free set of AF.
3. M is a 2-valued model of Π^-_{AF} iff $E^+_M \cup E_M$ is a range with respect to the conflict free set E_M.

Given these observations, one can clearly observe that GL-stage models characterize stage extensions.

Theorem 3.2. Let $AF := \langle AR, attacks\rangle$ be an argumentation framework. M is a GL-stage model of Π^-_{AF} iff $AR \setminus \{a|def(a) \in M\}$ is a stage extension of AF.

Proof:
The proof is direct by fact that if M is a GL-stage model of Π^-_{AF} then E_M is a conflict free set of AF.
□

4. Related work

In this section, a discussion about related work is presented. In particular, we present some other results about the characterization of argumentation inferences in terms of logic programming models.

In the literature, there are some other characterizations of semi-stable inference as logic programming inference [3, 21]. Caminada *et al.*, [3], showed that the semi-stable semantics can be characterized by the *L-stable semantics* and the mapping P_{AF} which is defined as follows: Given an argumentation framework $AF := \langle AR, attacks \rangle$:

$$P_{AF} = \bigcup_{x \in AR} \{x \leftarrow \bigwedge_{(y,x) \in attacks} not\ y\}$$

Unlike GL-supported models which are 2-valued models, the models of the L-stable semantics are 3-valued. Moreover, unlike Π_{AF} which is a declarative specification of admissible sets, P_{AF} seems as a declarative specification of conflict-free sets.

Strass [21] has also showed that the semi-stable semantics can by characterized by both the so-called *L-supported models* and *L-Stable models*. Unlike Caminada's characterization and our characterizations, Strass considered the mapping Π^-_{AF}. As we have observed in Section 2.4, the clauses of Π^-_{AF} is a subset of Π_{AF} which is the mapping that we considered in both Theorem 3.1 and Corollary 3.2. It is worth mentioning that the mapping introduced by Dung [7] can be transformed into Π^-_{AF}.

We cannot argue that one characterization is better than the other; however, we can observe that all these characterizations, including the ones introduced in this paper, offer different interpretations of semi-stable inference. Moreover, given that semi-stable inference has been characterized in terms of both L-stable semantics and L-supported modes, it seems that these semantics are related to GL-supported semantics.

In the literature, there are different characterizations of argumentation semantics in terms of logic programming semantics. A summary of these characterization is presented in Table 1.

According to Table 1, there are three main mappings which have been explored in order to map argumentation frameworks into logic programs. Π_{AF} and Π^-_{AF} are basically declarative specifications of the concept of *a conflict free set*. On the other hand, Π_{AF} is basically a declarative specification of *an admissible set*.

On one way or another, Table 1 argues for a *strong* relationship between argumentation inference and logic programming inference. Moreover, we can observe that the argumentation semantics which have been able to be characterized as logic programming inferences have been studies from different points of view, *e.g.,* Labellings [2]. This evidence argues that any *well-defined argumentation semantics* must be characterized by a logic programming semantics. However, further research is required in order to identify the necessary conditions which could support a basic definition of a *Well-defined Non-monotonic Inference* of any argumentation semantics.

The exploration of argumentation as logic programming inference is not limited to the characterization of argumentation semantics in terms logic programming semantics. Since Dung's presented his seminal paper [7], he showed that logic programming can support the construction of argumentation-based systems. Currently there are quite different logic-based argumentation engines which support the inference of argumentation semantics [8, 22, 11, 6]. It is well-known that the computational complexity of the decision problems of argumentation semantics ranges from NP-complete to $\Pi_2^{(p)}$-complete. In this set-

Table 1. Characterization of argumentation semantics as logic programming inferences.

Argumentation semantics	**Logic programming semantics using P_{AF}**	**Logic programming semantics using Π_{AF}**	**Logic programming semantics using Π^{-}_{AF}**
Grounded Semantics	Well-founded semantics [3], the Kripke-Kleene model [21]	Well-founded semantics [5]	Well-founded semantics [7], the Kripke-Kleene model [21]
Stable Semantics	Stable model semantics [3, 16], Supported models [21]	Stable model semantics [5]	Stable models semantics [7], Supported models [21]
Preferred Semantics	Regular semantics [3], M-supported models, M-stable models [21]	P-stable Semantics [5]	M-supported models, M-stable models [21]
Complete Semantics	3-valued stable semantics [24, 21], 3-valued supported models [21]	Supported Models [19]	3-valued stable semantics, 3-valued supported models [21]
Semi-stable Semantics	L-Stable [3, 21], L-Supported models [21]	GL-supported models (Theorem 3.1)	L-supported models, L-stable models [21]
Ideal Semantics		WFS^{+} [13]	
CF2 Semantics			MM^{*} [15]
Stage Semantics			GL-stage models (Theorem 3.2)

ting, Answer Set Programming has been consolidated as a strong approach for building argumentation-based systems [8, 12, 16, 22, 6].

Before moving on let us observe that from the logic programming side, most of the logic programming semantics used for characterizing argumentation inference have some relations with the stable model semantics [10]. We know that semi-stable semantics is an extension of stable semantics; however, none of the logic programming semantics which characterize semi-stable semantics are really presented as extensions of stable model semantics. Hence an interesting research question can be:

> [Q3] *Is there an extension of stable model semantics which characterizes semi-stable semantics?*

Let us observe that by giving answer to this question, semi-stable inference will be related to *Answer Set Programming's inference* (ASP). Currently, there are several extensions of stable models which start from stable model's definition for defining new logic programming semantics [1]. A common approach for extending stable models is following *an abductive approach*. Following ideas of abductive reasoning, in [18], the *stable_abductive argumentation semantics* was introduced. This semantics is based on the stable model semantics; moreover, like semi-stable semantics, stable_abductive argumentation is an intermediate semantics between stable and preferred semantics. Indeed, empirically speaking, we can observe that both semantics coincide. For instance, if we consider the argumentation framework: $AF = \langle\{a, b, c\}, \{(a, a), (b, c), (c, b)\}\rangle$, both semantics have the same extensions: $\{b\}$ and $\{c\}$. We have showed that both semantics satisfy common properties such as *relevance* [18]. This evidence makes us *conjecture* that stable_abductive argumentation semantics is equivalent to semi-stable semantics. In our future work, we will show this equivalence. Let us observe that if this conjecture is true, we will have an abductive construction of semi-stable semantics based on stable model semantics. Hence, this result will argue that semi-stable semantics is close to stable model semantics. In case that semi-stable and stable_abductive argumentation semantics are different, it will be relevant to observe which argumentation semantics is stronger than the other.

5. Conclusions

Argumentation inference is strongly influenced by Dung's argumentation style. Since Dung's approach was introduced, it has been showed that this approach can be regarded as logic programming inference. Currently, most of the well acceptable argumentation semantics have been characterized as logic programming inference. This evidence argue that whenever appears a new semantics it is totally reasonable to ask to be characterized as logic programming inference. However, further research is required in order to identify the necessary conditions which could support a basic definition of a *Well-defined Non-monotonic Inference* of any argumentation semantics.

According to Theorem 3.1, semi-stable semantics share the same mapping (*i.e.* Π_{AF}) with grounded, stable, preferred, complete and ideal semantics for been characterized as logic programming inference. This result argue that all these argumentation semantics can share the same interpretation of an argumentation framework as a logic program. Certainly, the logic programming semantics which are consider for characterizing these argumentation semantics share also a common interpretation of the argumentation inference which is restricted to the class of programs defined by Π_{AF}.

We have also showed that stage semantics can be also characterized by a logic programming semantics (see Theorem 3.2). This result argues that stage semantics has also logic programming foundations.

An interesting observation, from the results of this paper, is that the concept of range which is fundamental for defining semi-stable and stage semantics can be captured from the logic programming point of view by considering well-acceptable reductions from logic programming. Mainly GL-supported models and GL-stage models were defined by using GL reduction; moreover, it was observed that these models can also be defined by RED reduction. It is worth mentioning that reductions as GL and RED suggest some general rules for managing negation as failure. Hence, it could be interesting to explore if GL-supported models and GL-stage models are relevant from the logic point of view.

Acknowledgment

This research has been partially supported by VINNOVA (The Swedish Governmental Agency for Innovation Systems), the Swedish Brain Power Program and CONACyT (the Mexican Governmental Agency for science and technology) [CB-2008-01 No.101581].

References

[1] Baral, C.: *Knowledge Representation, Reasoning and Declarative Problem Solving*, Cambridge University Press, Cambridge, 2003.

[2] Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics, *Knowledge Eng. Review*, **26**(4), 2011, 365–410.

[3] Caminada, M., Sá, S., Alcântara, J.: *On the Equivalence between Logic Programming Semantics and Argumentation Semantics*, Technical Report ABDN-CS-13-01, University of Aberdeen, 2013.

[4] Caminada, M. W. A., Carnielli, W. A., Dunne, P. E.: Semi-stable semantics, *Journal of Logic and Computation*, **22**(5), 2012, 1207–1254.

[5] Carballido, J. L., Nieves, J. C., Osorio, M.: Inferring Preferred Extensions by Pstable Semantics, *Iberoamerican Journal of Artificial Intelligence (Inteligencia Artificial) ISSN: 1137-3601, (doi: 10.4114/ia.v13i41.1029)*, **13**(41), 2009, 38–53.

[6] Charwat, G., Dvorák, W., Gaggl, S. A., Wallner, J. P., Woltran, S.: Methods for solving reasoning problems in abstract argumentation - A survey, *Artificial Intelligence*, **220**, 2015, 28–63.

[7] Dung, P. M.: On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games., *Artificial Intelligence*, **77**(2), 1995, 321–358.

[8] Egly, U., Alice Gaggl, S., Woltran, S.: Answer-set programming encodings for argumentation frameworks, *Argument and Computation*, **1**(2), 2010, 147–177.

[9] Gelder, A. V., Ross, K. A., Schlipf, J. S.: The Well-Founded Semantics for General Logic Programs., *Journal of the ACM*, **38**(3), 1991, 620–650.

[10] Gelfond, M., Lifschitz, V.: The Stable Model Semantics for Logic Programming, *5th Conference on Logic Programming* (R. Kowalski, K. Bowen, Eds.), MIT Press, 1988, 1070–1080.

[11] Gómez-Sebastià, I., Nieves, J. C.: WizArg: Visual Argumentation Framework Solving Wizard, *13th International Conference of the Catalan Association for Artificial Intelligence (CCIA)*, 210, IOS Press, 2010, 249–258.

[12] Guerrero, E., Nieves, J. C., Lindgren, H.: Semantic-based Construction of Arguments: an Answer Set Programming Approach, *International Journal of Approximate Reasoning*, **DOI: 10.1016/j.ijar.2015.06.009**, 2015.

[13] Nieves, J. C., Osorio, M.: Ideal Extensions as Logical Programming Models, *Journal of Logic and Computation*, **DOI:10.1093/logcom/exu014**, 2014.

[14] Nieves, J. C., Osorio, M., Cortés, U.: Preferred Extensions as Stable Models, *Theory and Practice of Logic Programming*, **8**(4), July 2008, 527–543.

[15] Nieves, J. C., Osorio, M., Zepeda, C.: A Schema for Generating Relevant Logic Programming Semantics and its Applications in Argumentation Theory, *Fundamenta Informaticae*, **106**(2-4), 2011, 295–319.

[16] Nieves, J. C., Osorio, M., Zepeda, C., Cortés, U.: Inferring acceptable arguments with Answer Set Programming, *Sixth Mexican International Conference on Computer Science (ENC 2005)*, IEEE Computer Science Press, September 2005, 198–205.

[17] Osorio, M., Navarro, J. A., Arrazola, J. R., Borja, V.: Logics with Common Weak Completions, *Journal of Logic and Computation*, **16**(6), 2006, 867–890.

[18] Osorio, M., Nieves, J. C., Carballido, J. L.: The Stable Abducible Argumentation Semantics, *Latin American Workshop on Non-Monotonic Reasoning 2011*, 804, 2011, 57–68.

[19] Osorio, M., Nieves, J. C., Santoyo, A.: Complete Extensions as Clark's Completion Semantics, *Mexican International Conference on Computer Science*, IEEE Computer Science Press, 2013, 81–88.

[20] Prakken, H., Vreeswijk, G. A. W.: Logics for defeasible argumentation, in: *Handbook of Philosophical Logic* (D. Gabbay, F. Günthner, Eds.), vol. 4, second edition, Kluwer Academic Publishers, Dordrecht/Boston/London, 2002, 219–318.

[21] Strass, H.: Approximating operators and semantics for abstract dialectical frameworks, *Artif. Intell.*, **205**, 2013, 39–70.

[22] Toni, F., Sergor, M.: Argumentation and Answer Set Programming, in: *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning* (M. Balduccini, T. C. Son, Eds.), vol. 6565 of *LNCS*, Springer, 2011.

[23] Verheij, B.: Two approaches to dialectical argumentation: admissible sets and argumentation stages, *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC 1996)*, 1996.

[24] Wu, Y., Caminada, M., Gabbay, D. M.: Complete Extensions in Argumentation Coincide with 3-Valued Stable Models in Logic Programming, *Studia Logica*, **93**(2-3), 2009, 383–403.