Ideal Extensions as Logical Programming Models

JUAN CARLOS NIEVES
Department of Computing Science
Umeå University
SE-901 87, Umeå, Sweden
jcnieves@cs.umu.se

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

December 13, 2013

Abstract

We show that the ideal sets of an argumentation framework can be characterized by two kinds of logical models: *ideal models* (2-valued logical models) and *p-stable models* (2-valued logical models). We also show that the maximal ideal set of an argumentation framework can be characterized by the well-founded$^+$ model (a 3-valued logical model). These results argue for the logical foundations of the ideal sets of an argumentation framework. Moreover, these results consolidate the strong relationship between argumentation semantics and logic programming semantics with *negation as failure*. More accurately, we prove that the five argumentation semantics suggested by Dung *et al.*, grounded, stable, preferred, complete and ideal semantics, can be characterized by the well-founded model, stable-model, p-stable, Clark's completion and well-founded$^+$ model semantics, respectively by using a unique mapping from argumentation frameworks into logic programs. We observe that the labellings of these argumentation semantics can be inferred by the logical models of a logic program.

1 Introduction

Recently, theoretical argumentation research has been strongly influenced by the abstract argumentation theory of Dung [17]. This approach is mainly oriented towards managing the interaction between arguments. In order to capture the interaction between arguments, Dung introduced a single structure called *Argumentation Framework (AF)*. An argumentation framework is basically a tuple of two sets: a set of arguments

and a set of disagreements between arguments called *attacks*. An argumentation framework can be regarded as a directed graph in which the arguments are represented by nodes and the attack relations are represented by the edges. In Figure 1, an example of an argumentation framework and its graph representation is given.

Given an argumentation framework, a wide range of patterns of selection of arguments has been proposed [2]. These patterns of selection of arguments have been called *argumentation semantics*. Among them, *ideal semantics* has emerged as an alternative for performing *skeptical reasoning* in argumentation reasoning [18, 20, 21]. As Dunne *et al.* argue, ideal semantics suggests a rigorous basis for the notion of *justifiably* accepted skeptical belief [21]:

> "an argument must not only be skeptically accepted (as in the standard sense), but also must belong to a set of internally consistent and self-defending collection of skeptically accepted arguments"

The concept of an ideal set has been defined as an admissible set which is contained in every preferred extension of a given argumentation framework. For instance, one can see that the argumentation framework of Figure 1 has six admissible sets: $\{\}$, $\{a\}$, $\{b\}$, $\{d\}$, $\{a, d\}$, $\{b, d\}$ and two preferred extensions: $\{a, d\}$, $\{b, d\}$. Therefore, the argumentation framework of Figure 1 has two ideal sets: $\{\}$, $\{d\}$. Of these ideal sets, *the maximal ideal set* is the more interesting one to infer from an argumentation framework because it is a super set of the grounded extension.

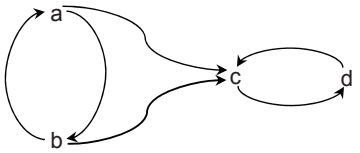

Figure 1: Graph representation of $AF := \langle\{a, b, c, d\}, \{(a, b), (b, a), (a, c), (b, c), (c, d), (d, c)\}\rangle$.

The inference of argumentation semantics has been explored from the point of view of different theories, *e.g.,* equation theory [26], graph theory [4], and logic programming theory [10, 11, 17]. A basic requirement for exploring argumentation inference in terms of another theory is to express argumentation frameworks in terms of the given theory and to then characterize argumentation semantics in the given theory. For instance, Gabbay introduced some mappings for expressing argumentation frameworks in terms of *numerical equations* [26]. The solutions of these numerical equations characterize different argumentation semantics. One of the main advantages of characterizing argumentation inference from the point of view of another theory is that different approaches are defined for computing argumentation semantics. It is worth mentioning that the computational complexity of the decision problems of argumentation semantics range from linear to $\Pi_2^{(p)}$-complete [19]. By following Gabbay's approach, one can use tools such as MAPLE or MATLAB for computing argumentation semantics.

Dung showed that argumentation can be viewed as logic programming with *negation as failure* and vice versa. This strong relationship between argumentation and logic programming has given way to intensive research in order to explore the relationship between argumentation and logic programming [10, 11, 17, 31, 29, 35, 30, 38]. A basic requirement for exploring the relationship between argumentation and logic programming is to identify proper mappings which allow us to transform an argumentation framework into a logic program and vice versa. The flexibility of these mappings will restrict the understanding of argumentation as logic programming (and vice versa). Therefore, defining simple and flexible mappings which regard argumentation as logic programming (and vice versa) will impact the use of logic programming in argumentation (and vice versa). Currently, we can find different mappings for regarding argumentation as logic programming (and vice versa) [10, 11, 17, 26]. All of them offer different interpretations of argumentation as logic programming (and vice versa). Depending on these interpretations, one can identify direct relationships between the argumentation inferences and the logic programming inferences.

In this paper, we will limit our attention to only the interpretation of argumentation as logic programming. We will extend results which have shown that by considering *a unique mapping* of an argumentation framework into a logic program (Π_{AF}^{acc}), one can characterize grounded, stable, preferred and complete semantics in terms of well-founded, stable, p-stable, Clark-completion logical models, respectively. In particular, following the mapping Π_{AF}^{acc} and the characterization of argumentation semantics in terms of logic programming semantics, a couple of research questions arise in the context of ideal sets:

Q1: can the ideal sets of an argumentation framework be characterized by a credulous logic programming semantics?

Q2: If so, is such semantics interesting or well known?

Let us observe that, by answering these questions, we identify *logical foundations* to the ideal semantics in the sense that the ideal sets can be defined in terms of *logical models* of a given logical theory (*i.e.* a logic program), and, therefore, using logic-based tools for computing ideal semantics. Currently, it is known that logic-based tools, *e.g.,* SAT Solvers, Answer Set Programming tools, offer solid solutions for computing high complexity problems.

It is worth mentioning that if the ideal semantics is characterized by Π_{AF}^{acc} and a logic programming semantics, the ideal semantics will share the same mapping with grounded, stable, preferred and complete semantics in order to be characterized by logical models. This suggests that Π_{AF}^{acc} defines *a basic root* for capturing argumentation semantics based on admissible sets in terms of logical models.

In order to illustrate our approach, let us map the argumentation framework AF of Figure 1 into the logic program Π_{AF}^{acc} which is of the form:

$def(a) \leftarrow \ not\ def(b).$ $\quad$ $def(a) \leftarrow def(a).$
$def(b) \leftarrow \ not\ def(a).$ $\quad$ $def(b) \leftarrow def(b).$
$def(c) \leftarrow \ not\ def(a).$ $\quad$ $def(c) \leftarrow def(b).$
$def(c) \leftarrow \ not\ def(b).$ $\quad$ $def(c) \leftarrow def(a).$
$def(c) \leftarrow \ not\ def(d).$ $\quad$ $def(c) \leftarrow def(c).$
$def(d) \leftarrow \ not\ def(c).$ $\quad$ $def(d) \leftarrow def(a), def(b), def(d).$

$acc(a) \leftarrow \ not\ def(a)$ $\quad$ $acc(b) \leftarrow \ not\ def(b)$
$acc(a) \leftarrow \ not\ def(c)$ $\quad$ $acc(b) \leftarrow \ not\ def(d)$

Let us observe that the clauses of the form $def(x) \leftarrow \ not\ d(y)$ suggest that the argument x is defeated (which means that it cannot be part of an admissible set) if the argument y is not defeated. In the same way, the clauses of the form $def(x) \leftarrow d(y_1), \ldots, d(y_n)$ suggest that the argument x is defeated if the arguments $y_1, \ldots, y_n$ are defeated. The last clauses of the form $acc(x) \leftarrow \ not\ def(x)$ suggest that if argument x is not defeated, it is an accepted argument which means that the argument belongs to an admissible set.

One can see that the 2-valued models of Π_{AF}^{acc} are: $\{def(a), def(b), def(c), def(d)\}$, $\{acc(a), def(b), def(c), def(d)\}$, $\{acc(b), def(a), def(c), def(d)\}$, $\{acc(d), def(a), def(b), def(c)\}$, $\{acc(a), acc(d), def(b), def(c)\}$, $\{acc(b), acc(d), def(a), def(c)\}$. If we observe only the atoms of the form $acc(x)$ of each model, we can see that each model of Π_{AF}^{acc} characterizes an admissible set of AF. This suggests that Π_{AF}^{acc} defines an *isomorphism* of the admissible sets of AF in terms of the logical models of Π_{AF}^{acc}.

Among the different logic programming semantics for performing skeptical reasoning, WFS is one of the well-accepted semantics [1, 14]. Besides WFS, WFS$^+$ (also called *the Well-Founded-By-Cases Semantics*) [14, 36] has shown itself to be a sound logic programming semantics for performing skeptical reasoning. Indeed, Dix showed that WFS$^+$ is *a well-behaved semantics* [15], which means that it satisfies basic principles of non-monotonic reasoning (see [15] for a formal definition of a well-behaved semantics). In addition to being a well-behaved semantics, Schlipf [36] showed that *WFS by cases* (WFS$^+$) is a logic programming semantics which satisfies a good number of *common sense goals*. It is worth mentioning that, like WFS, WFS$^+$ is a 3-valued semantics. To relate WFS$^+$ and argumentation inference may help to consolidate WFS$^+$ as a sound skeptical reasoning approach.

Based on this background, we introduce new results *w.r.t.* argumentation as logic programming with negation as failure. On the one hand, by following the ideal set definition, we introduce the concept of an *ideal model* of a logic program. The definition of an ideal model will be based on 2-valued classical models and p-stable models. On the other hand, by considering an argumentation framework AF and a uniform mapping of an AF into a logic program Π_{AF}^{acc}, we show that

- the ideal models of Π_{AF}^{acc} characterize the ideal sets of AF (Theorem 5) and
- the WFS$^+$ model of Π_{AF}^{acc} characterizes the maximal ideal set of AF (Theorem 11).
- the well founded model of Π_{AF}^{acc} characterizes the grounded extension of AF (Theorem 3),

- the stable models of Π_{AF}^{acc} characterize the stable extensions of AF (Theorem 3),
- the p-stable models of Π_{AF}^{acc} characterize the preferred extensions of AF (Theorem 3) and
- the supported models of Π_{AF}^{acc} characterize the complete extensions of AF (Theorem 3).

For instance, the ideal models of Π_{AF}^{acc} are: $\{def(a), def(b), def(c), def(d)\}$ and $\{acc(d), def(a),\ def(b), def(c)\}$. If we only observe the atoms of the form $acc(x)$ which belong to these models, we can see that these models characterize the ideal sets of AF, *i.e.* $\{\}$ and $\{d\}$. On the other hand, we can see that $WFS^{+}(\Pi_{AF}^{acc}) = \langle\{acc(d)\},\ \{def(a),\ def(b), def(c)\}\rangle$. By considering only the positive atoms of $WFS^{+}(\Pi_{AF}^{acc})$, WFS^{+} characterizes the maximal ideal set of AF.

Based on these results, we show that the ideal sets of an argumentation framework can be characterized in terms of p-stable models (Corollary 1). Since the p-stable models of Π_{AF}^{acc} characterize the preferred semantics and the p-stable models also are able to characterize the ideal sets of an argumentation framework, the p-stable semantics introduces logical definitions of both the preferred semantics and ideal sets.

Let us observe that WFS^{+} suggests a different interpretation of the maximal ideal set of an argumentation framework in terms of *a logical model*. More accurately, WFS^{+} introduces a logical definition of the maximal ideal set of an argumentation framework. From the logic programming point of view, this characterization argues that WFS^{+} is a sound approach for performing non-monotonic skeptical reasoning.

Given that WFS^{+} can be characterized in terms of *rewriting systems* [16], our characterization of the maximal ideal set of an argumentation framework in terms of WFS^{+} at least has the following implications:

- WFS^{+} introduces a simple *equational* approach for computing the maximal ideal set of an argumentation framework.
- The rewriting system which characterizes WFS^{+} identifies a set of logical equivalents which satisfies the maximal ideal set of an argumentation framework.

Rewriting is a very powerful method for dealing computationally with *equations*. Oriented equations, called *rewrite rules*, are used to replace equals with equals, but only in one direction. The theory of rewriting centers around the concept of *normal form*, an expression that cannot be rewritten any further. Computation consists of rewriting to a normal form; when the normal form is unique, it is taken as the value of the initial expression. When rewriting equal terms always leads to the same normal form, the set of rules is said to be convergent and rewriting can be used to decide validity of identities in the equational theory. Rewriting has the computational power of Markov algorithms and of *recursive functions* and *Turing machines* [13]. It is worth mentioning that, unlike Gabbay's approach, which is a numerical approach [24], our approach is both a symbolic and logic-based approach. In Section 6, we argue more about the similarities and differences between Gabbay's approach and our approach.

This paper is an improved and extended version of the paper [28]. In the new version of the paper, in addition to the formal presentation of proofs, we introduce new

results of the relationship between ideal models and rewriting systems. We show that WFS$^+$ not only characterizes an ideal set of an argumentation framework AF, but also characterizes the maximal ideal of AF. We present a related work section where we compare our approach with the state of the art. Indeed, we observe that our approach is able to infer the labellings of the five argumentation semantics suggested by Dung *et al.*

The rest of the paper is divided as follows: In Section 2, a basic background of logic programs and logic programming semantics is introduced. The definitions of grounded, stable, preferred, complete and ideal semantics are presented. At the end of this section, our mappings (Π_{AF} and Π_{AF}^{acc}) of argumentation frameworks into logic programs are presented. In Section 3, we explore the relationship between ideal models and ideal sets. In Section 4, we explore how the minimal ideal model can be characterized by the WFS$^+$ model. This relationship will suggest a relationship between the WFS$^+$ model and the maximal ideal set of an argumentation framework. All the results of Section 4 are presented in terms of Π_{AF}. In Section 5, the characterization of the maximal ideal set of an argumentation framework is introduced in terms of Π_{AF}^{acc}. In Section 6, an overview of the related work is presented. In the last section, we outline our conclusions.

2 Background

In this section, we first define the syntax of a valid logic program; after that the p-stable, Well-Founded and Well-Founded$^+$ Semantics are presented. In the context of argumentation, we present some basic concepts of argumentation theory and a mapping of argumentation frameworks into logic programs.

2.1 Basic Notions

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 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 an abbreviation of $a_0 \leftarrow \top$, where $\top$ and $\bot$ are the ever true and ever false propositions respectively. 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; the set of atoms is a classical model of P if the induced interpretation evaluates P to true).

2.2 P-stable semantics

An important logic programming semantics that we consider in the relationship between argumentation semantics and logic programming semantics is the so called *p-stable semantics* [32]. There are some results which have shown that the preferred extensions of an argumentation framework can be characterized in terms of p-stable models [11]. The p-stable model semantics has its logical foundations in paraconsistent logics.

P-stable semantics is defined in terms of a single reduction which is defined as follows:

Definition 1 *[32] Let P be a normal program and M be a set of literals. We define $RED(P, M) := \{l \leftarrow \mathcal{B}^+,\ not\ (\mathcal{B}^- \cap M) | l \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^- \in P\}$.*

Let us consider the set of atoms $M_1 := \{a, b\}$ and the following normal program P_1:

$a \leftarrow\ not\ b,\ not\ c.$
$a \leftarrow b.$
$b \leftarrow a.$

We can see that $RED(P_1, M_1)$ is:
$a \leftarrow\ not\ b.$
$a \leftarrow b.$
$b \leftarrow a.$

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

Definition 2 *[32] 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.*

Let us consider again M_1 and P_1 in order to illustrate the definition. We want to show whether M_1 is a p-stable model of P_1. First, we can see that M_1 is a model of P_1, *i.e.* $\forall\ C \in P_1$, M_1 evaluates C to true. Now, we have to prove each atom of M_1 from $RED(P_1, M_1)$ by using classical inference, *i.e.* $RED(P_1, M_1) \vdash M_1$. Let us consider the proof of the atom a, which belongs to M_1, from $RED(P_1, M_1)$.

1. $(a \vee b) \rightarrow ((b \rightarrow a) \rightarrow a)$	Tautology
2. $\neg\, b \rightarrow a$	Premise from $RED(P_1, M_1)$
3. $a \vee b$	From 2 by logical equivalency
4. $(b \rightarrow a) \rightarrow a)$	From 1 and 3 by Modus Ponens
5. $b \rightarrow a$	Premise from $RED(P_1, M_1)$
6. a	From 4 and 5 by Modus Ponens

Remember that the formula $\neg\, b \rightarrow a$ corresponds to the normal clause $a \leftarrow \ not\ b$ which belongs to the program $RED(P_1, M_1)$. The proof for the atom b, which also belongs to M_1, is similar. Then we can conclude that $RED(P_1, M_1) \Vdash M_1$. Hence, M_1 is a *p-stable model* of P_1.

2.3 The Well-Founded Semantics and WFS$^+$

The well-founded semantics (WFS) is a logic programming semantics which has shown to be close to argumentation semantics. Let us recall that Dung showed that the grounded semantics can be characterized by WFS [17]. In this section, we will present a definition of WFS in terms of rewriting systems. In addition to WFS' definition, an extension of WFS will be presented which is called well founded$^+$ semantics (WFS$^+$) [15]. We will show that WFS$^+$ is also closely related to argumentation semantics.

Given that both WFS and WFS$^+$ are 3-valued logic semantics, we start presenting some definitions *w.r.t.* 3-valued logic semantics. A partial interpretation based on a signature $\mathcal{L}$ is a disjoint pair of sets $\langle I_1, I_2\rangle$ such that $I_1 \cup I_2 \subseteq \mathcal{L}$. A partial interpretation is total if $I_1 \cup I_2 = \mathcal{L}$. Given two interpretations $I = \langle I_1, I_2\rangle$, $J = \langle J_1, J_2\rangle$, we set $I \leq_k J$ *if, by definition,* $I_i \subseteq J_i$, $i = 1, 2$. Clearly $\leq_k$ is a partial order. When we look at interpretations as sets of literals then $\leq_k$ corresponds to $\subseteq$. A general semantics SEM is a function on $\text{Prog}_{\mathcal{L}}$ which associates every program with a partial interpretation.

Given a signature $\mathcal{L}$ and two semantics SEM_1 and SEM_2, we define $\text{SEM}_1 \leq_k \text{SEM}_2$ if for every program $P \in \text{Prog}_{\mathcal{L}}$, $\text{SEM}_1(P) \leq_k \text{SEM}_2(P)$.

Definition 3 (SEM) *For any logic program P, we define HEAD$(P) = \{a|\ a \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^- \in P\}$ — the set of all head-atoms of P. We also define SEM$(P) = \langle P^{true}, P^{false}\rangle$, where $P^{true} := \{p|\ p \leftarrow \top \in P\}$ and $P^{false} := \{p|\ p \in \mathcal{L}_P \backslash \text{HEAD}(P)\}$.*

Now, we define some basic transformation rules for normal logic programs which will be considered for characterizing WFS and WFS$^+$.

Definition 4 (Basic Transformation Rules) *[16] A transformation rule is a binary relation on $Prog_{\mathcal{L}}$. The following transformation rules are called* basic. *Let a program $P \in Prog_{\mathcal{L}}$ be given.*

RED$^+$: *This transformation can be applied to P, if there is an atom a which does not occur in HEAD(P). **RED**$^+$ transforms P to the program where all occurrences of $not\ a$ are removed.*

RED$^-$: *This transformation can be applied to P, if there is a rule $a \leftarrow \top \in P$. **RED**$^-$ transforms P to the program where all clauses that contain $not\ a$ in their bodies are deleted.*

Success: *Suppose that P includes a fact $a \leftarrow \top$ and a clause $q \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-$ such that $a \in \mathcal{B}^+$. Then we replace the clause $q \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-$ by $q \leftarrow \mathcal{B}^+ \setminus \{a\},\ not\ \mathcal{B}^-$.*

Failure: *Suppose that P contains a clause $q \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-$ such that $a \in \mathcal{B}^+$ and $a \notin HEAD(P)$. Then we erase the given clause.*

Loop: *We say that P_2 results from P_1 by Loop_A if, by definition, there is a set A of atoms such that 1. for each rule $a \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^- \in P_1$, if $a \in A$, then $\mathcal{B}^+ \cap A \neq \emptyset$, 2. $P_2 := \{a \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^- \in P_1 | \mathcal{B}^+ \cap A = \emptyset\}$, 3. $P_1 \neq P_2$.*

SUB : *If P contains two clauses $a \leftarrow \mathcal{B}_1^+,\ not\ \mathcal{B}_1^-$ and $a \leftarrow \mathcal{B}_2^+,\ not\ \mathcal{B}_2^-$, where $\mathcal{B}_1^+ \subseteq \mathcal{B}_2^+$ and $\mathcal{B}_1^- \subseteq \mathcal{B}_2^-$, then $a \leftarrow \mathcal{B}_2^+,\ not\ \mathcal{B}_2^-$ is removed from P.*

TAUT : *Suppose P contains a rule C which has the same atom in its head and its positive body. Then we remove this rule.*

LC : *Suppose $P \vdash a$ for an atom a. Then we add the rule $a \leftarrow \top$ in P.*

Let us observe that most of the basic transformations are syntactically applicable in the sense that one can easily see when a given basic transformation is applicable by a direct syntactic analysis of the rules of a given logic program. However, the loop transformation is possibly the transformation rule which is not applicable in a straightforward way. One can observe that the application of the loop transformation depends on the set A which is basically *an unfounded set* [27]. Informally speaking, an unfounded set consists of positive atoms which are not possibly true, *i.e.* cannot be derived by assuming all negative literals to be true. In this sense, one can see that by considering the greatest unfounded set of P_1 for defining A, the loop transformation maximizes the number of rules which can be removed from P_1. This means that if P_2 results from P_1 by using the maximal A, one cannot apply once again $Loop_A$ to P_2. In [6], a general method for computing the maximal A of a given logic program was introduced. The algorithm for computing the maximal A (which means the greatest unfounded set of P_1) works as follows:

1. $P' = \{a \leftarrow \mathcal{B}^+ | a \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^- \in P_1\}$
2. Let MM_P be the minimal model of P'[1].
3. $A = \mathcal{L}_{P_1} \setminus MM_P$

By considering A, one can get P_2 as follows:

$$P_2 = \{a \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^- | a \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^- \in P_1 \text{ and } \mathcal{B}^+ \cap A = \emptyset\}$$

In the rest of the paper, whenever the loop transformation is applied, we assume that the greatest unfounded set was used for defining A.

The transformation rules of Definition 4 will be used to define two rewriting systems as follows:

[1] Since P' is a definite program P' has a unique minimal model.

CS_0 = $\{RED^+, RED^-, Success, Failure, Loop\}$

CS_1 = $CS_0 \cup \{SUB, TAUT, LC\}$

The uniquely determined *normal form* of a normal logic program P with respect to a rewriting system $\mathcal{CS}$ is denoted by $norm_{\mathcal{CS}}(P)$. By the normal form of a normal logic program P, we mean the resulting normal logic program $norm_{\mathcal{CS}}(P)$ after applying the transformation rules which belong to $\mathcal{CS}$ and none of the transformation rules which belong to $\mathcal{CS}$ can be applied to $norm_{\mathcal{CS}}(P)$.

In order to illustrate the basic transformation rules, let us consider the following example:

Example 1 *Let P be the following normal program:*

$$d(b) \leftarrow not\ d(a). \qquad d(b) \leftarrow \top. \qquad d(c) \leftarrow not\ d(b).$$
$$d(c) \leftarrow d(a).$$

*Now, let us apply CS_0 to P. Since $d(a) \notin HEAD(P)$, then, we can apply **RED**$^+$ to P. Thus we get:*

$$d(b) \leftarrow \top. \qquad d(c) \leftarrow not\ d(b). \qquad d(c) \leftarrow d(a).$$

*Notice that we can apply **RED**$^-$ to the new program, thus we get:*

$$d(b) \leftarrow \top. \qquad\qquad d(c) \leftarrow d(a).$$

*Finally, we can apply **Failure** to the new program, thus we get:*

$$d(b) \leftarrow \top.$$

This last program is the normal form *of P w.r.t. CS_0, because none of the transformation rules from CS_0 can be applied.*

Every rewriting system $\mathcal{CS}$ induces a 3-valued logic semantics SEM$_{\mathcal{CS}}$ as follows:

$$\text{SEM}_{\mathcal{CS}}(P) := \text{SEM}(norm_{\mathcal{CS}}(P))$$

WFS was introduced in [27] and was characterized in terms of rewriting systems in [7]. This characterization is defined as follows:

Theorem 1 *[7] CS_0 is a confluent rewriting system. It induces a 3-valued semantics that is the Well-founded Semantics.*

Example 2 *Let P be the normal logic program that was introduced in Example 1. We saw in Example 1 that $norm_{\mathcal{CS}_0}(P)$ is:*

$$d(b) \leftarrow \top.$$

This means that $SEM_{\mathcal{CS}_0}(P) = \langle\{d(b)\}, \{d(a), d(c)\}\rangle$. Let us observe that the atoms $d(a)$ and $d(c)$ do not appear in $norm_{\mathcal{CS}_0}(P)$; however, they belong to the signature of P. Therefore, they are considered as atoms of the signature of $norm_{\mathcal{CS}_0}(P)$. Hence, by Theorem 1, $WFS(P) = \langle\{d(b)\}, \{d(a), d(c)\}$

There is an extension of WFS called WFS^+. This extension of WFS is a logic programming semantics which, like WFS, is *a well-behaved semantics*[2] [15]. WFS^+ was introduced in [14] and characterized in terms of rewriting systems in [16].

Theorem 2 *[16] The WFS^+ semantics is induced by the confluent rewriting system CS_1.*

Before moving on, let us observe that the consideration of rewriting systems which are confluent and terminating defines a general methodology for characterizing different logic programming semantics. Observe that the differences between WFS and WFS$^+$ are three rewriting rules, *i.e.* SUB, TAUT, LC.

2.4 Argumentation theory

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

Definition 5 *[17] 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 $\subseteq AR \times AR$.

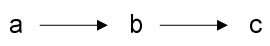

Figure 2: Graph representation of $AF := \langle \{a, b, c\}, \{(a, b), (b, c)\} \rangle$

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 is represented as it is shown in Figure 2. 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 6 *[17]*

- *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.*

[2]The class of well-behaved semantics is a class of logic programming semantics which satisfies a set of well-expected properties for performing non-monotonic reasoning. A study of these logic programming semantics can be found in [15].

- *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.*

By considering the concept of admissible set, in [17], Dung introduced four basic argumentation semantics: the grounded, stable, preferred and complete semantics. Even though all of them are based on admissible sets, each of them represents a different pattern of selection of arguments. Three of them follow a credulous approach, which means that given a set of conflicts of arguments, these semantics can suggest different sets of arguments which can be regarded as coherent points of view. These credulous argumentation semantics are defined as follows.

Definition 7 *[17] 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.*

In [17], Dung introduced a skeptical argumentation semantics which means that, given a set of conflicts of arguments, this semantics can suggest only one set of arguments which can be regarded as a coherent point of view. This semantics is called *grounded semantics* and is defined in terms of a *characteristic function*.

Definition 8 *[17] The characteristic function, denoted by F_{AF}, of an argumentation framework $AF = \langle AR, attacks \rangle$ is defined as follows:*
$F_{AF} : 2^{AR} \rightarrow 2^{AR}$
$F_{AF}(S) = \{A |$ *A is acceptable w.r.t. S* $\}$

Definition 9 *[17] The grounded extension of an argumentation framework AF, denoted by GE_{AF}, is the least fixed point of F_{AF}.*

From the semantics introduced in [17], the grounded semantics was the only semantics which follows a skeptical approach. However in [18], an extension of the grounded semantics, which is called *ideal semantics*, was introduced.

Definition 10 *[18] Let $AF := \langle AR, attacks \rangle$ be an argumentation framework. $S \subseteq AR$ is an ideal set if and only if S is admissible and it is contained in every preferred set of AF.*

Given that the set of ideal sets of an argumentation framework defines a join-semilattice[3], the *maximal ideal set* is usually the interesting ideal set to infer from an argumentation framework.

[3]Dung *et al.* showed that the union of two ideal sets is an ideal set (see Lemma 2.1 from [18]).

Since the first argumentation semantics were introduced in [17], Dung's argumentation semantics have given place to different formal studies about their properties. One of these formal studies has been to regard them as formal non-monotonic reasoning inferences. In this setting, one can find that the argumentation semantics are closely related to logic programming semantics with *negation as failure*. In the following sections, we will study different relations between *ideal sets* and *logic programming semantics* with negation as failure.

2.5 Mapping from argumentation frameworks to 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 mapping from an argumentation framework into a logic program will be presented. Let us observe that this mapping basically is *a declarative representation* of an argumentation framework by having in mind the ideas of *conflictfreeness* and *reinstatement* which are the basic concepts behind the definition of admissible sets.

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

Definition 11 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework and $a \in AR$. We define the transformation function $\Pi(a)$ as follows:*

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

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

Let us see that if a given argument a has no attacks, $\Pi(a) = \{\}$. This situation happens because any argument that has no attacks is an acceptable argument which means that it belongs to all admissible sets of AF. On the other hand, we can identify two conditions in the mapping $\Pi(a)$:

1. The first condition of $\Pi(a)$ ($\bigcup_{b:(b,a)\in attacks}\{def(a) \leftarrow \ not\ def(b)\}$) suggests that the argument a is defeated whenever one of its adversaries is not defeated.

2. The second condition of $\Pi(a)$ ($\bigcup_{b:(b,a)\in attacks}\{def(a) \leftarrow \bigwedge_{c:(c,b)\in attacks} def(c)\}$) suggests that the argument a is defeated when all the arguments that defend[4] a are defeated. Hence, one can see that if an argument a is attacked by an argument b which has no attacks, this condition generates the set $\{def(a) \leftarrow \top\}$. This means that a cannot be part of an admissible set; hence, a is a defeated argument.

The transformation function $\Pi(a)$ with respect to an argumentation framework AF is defined as follows:

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

Definition 12 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework. We define its associated normal program as follows:*

$$\Pi_{AF} := \bigcup_{a \in AR} \{\Pi(a)\}$$

Let us observe that if we consider an argumentation framework of n arguments in which all the arguments attack each other, the first part of the mapping Π_{AF}, *i.e.* $(\bigcup_{b:(b,a)\in attacks}\{def(a) \leftarrow \ not\ def(b)\})$, generates n^2 clauses of size 2, one atom in the head and one atom in the body, and the second condition of the mapping Π_{AF}, *i.e.* $(\bigcup_{b:(b,a)\in attacks}\{def(a) \leftarrow \bigwedge_{c:(c,b)\in attacks} def(c)\})$, generates n clauses of size $n+1$, n atom in the body and one in the head. Hence, in this case the mapping Π_{AF} generates at most $n^2 + n$ clauses[5]. Moreover, by considering the size of each of the clauses, the size of the output, in this case, is $2n^2 + (n+1)n$.

As one can see in Π_{AF}, the language of Π_{AF} only identifies the arguments which can be considered as defeated. By considering *total interpretations*, as the ones suggested by logic programming semantics such as stable model semantics, p-stable semantics, we can assume that any argument which is not defeated in a model of Π_{AF} will be acceptable. This means that given an argumentation framework $AF = \langle AR, Attacks\rangle$ if M is a model of Π_{AF}, then any atom $def(x)$ which is false in M will identify an argument x which is acceptable. This assumption suggests a normal clause of the following form:

$$acc(x) \leftarrow \ not\ def(x).$$

where $acc(x)$ denotes that the argument x can be considered as accepted. This clause essentially fixes as acceptable any argument which is not fixed as defeated in Π_{AF}. On the other hand, this clause suggests an easy form for inferring the acceptable arguments of AF. Being aware of these observations, we extend Π_{AF} as follows:

$$\Pi^{acc}_{AF} := \Pi_{AF} \cup \bigcup_{a \in AR} \{acc(a) \leftarrow \ not\ def(a)\}$$

It is easy to see that Π^{acc}_{AF} is basically a conservative extension of Π_{AF}. In other words, the logical models of Π_{AF} can be inferred from the logical models of Π^{acc}_{AF} by intersecting each model of Π^{acc}_{AF} with the language of Π^{acc}_{AF}.

Observation
For the sake of simplicity of the proofs, part of the results will be concentrated on Π_{AF}; however, as we have observed, Π^{acc}_{AF} is a conservative extension of Π_{AF}.

In order to illustrate Π_{AF} and Π^{acc}_{AF}, let us consider the following example.

Example 3 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework such that $AR := \{a1, b1, b2, c1, c2, c3\}$ and $attacks := \{(c1, b1), (c2, b1), (c3, b2), (b1, a1), (b2, a1)\}$. The graph representation of AF is presented in Figure 3.*

[5]It is worth mentioning that in this case, the grounded program of the Dung's mapping introduced in [17] generates n^2 clauses, counting only the clauses of the form $def(x) \leftarrow acc(y)$, each of these clauses is of size 2.

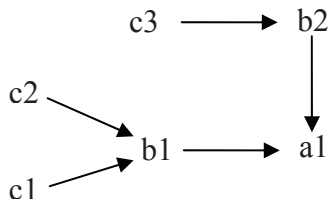

Figure 3: Graph representation of $AF := \langle\{a1, b1, b2, c1, c2, c3\}, \{(c1, b1), (c2, b1), (c3, b2), (b1, a1), (b2, a1)\}\rangle$

One can see that Π_{AF} *is:*

$$def(b1) \leftarrow not\ def(c1). \qquad def(b1) \leftarrow not\ def(c2).$$
$$def(b1) \leftarrow \top.$$
$$def(b2) \leftarrow not\ def(c3). \qquad def(b2) \leftarrow \top.$$
$$def(a1) \leftarrow not\ def(b1). \qquad def(a1) \leftarrow def(c1), def(c2).$$
$$def(a1) \leftarrow not\ def(b2). \qquad def(a1) \leftarrow def(c3).$$

On the other hand, Π_{AF}^{acc} *is* Π_{AF} *union with the following set of clauses:*

$$acc(a1) \leftarrow not\ def(a1).$$
$$acc(b1) \leftarrow not\ def(b1).$$
$$acc(b2) \leftarrow not\ def(b2).$$
$$acc(c1) \leftarrow not\ def(c1).$$
$$acc(c2) \leftarrow not\ def(c2).$$
$$acc(c3) \leftarrow not\ def(c3).$$

Before moving on, let us observe that by considering either Π_{AF} or Π_{AF}^{acc}, the grounded, stable, preferred and complete semantics can be characterized by different logic programming semantics [11, 35]. In order to introduce these characterizations, we introduce the following notation:

Given a set of arguments E:

$$tr(E) = \{acc(a)|a \in E\} \cup \{def(b)|b \text{ is an argument and } b \notin E\}$$

$$tr^{acc}(E) = \{acc(a)|a \in E\}$$

$$tr^{def}(E) = \{def(a)|a \in E\}$$

Theorem 3 *[11, 35] Let* $AF = \langle AR, Attacks\rangle$ *be an argumentation framework,* $E \subseteq AR$ *and* $E^+ = \{b|a \in E \text{ and } (a, b) \in Attacks\}$*. Then:*

- *E is the grounded extension of AF iff* $\langle tr^{acc}(E) \cup tr^{def}(E^+), tr^{acc}(E^+) \cup tr^{def}(E)\rangle$ *is the well-founded model of* Π_{AF}^{acc}*.*

- *E is a stable extension of AF iff $tr(E)$ is a stable model of Π_{AF}^{acc}.*
- *E is a preferred extension of AF iff $tr(E)$ is a p-stable model of Π_{AF}^{acc}.*
- *E is a complete extension of AF iff $tr(E)$ is a Clark's completion of Π_{AF}^{acc}.*

Let us observe that disregarding the apparent syntactic differences between the Dung's mapping and Π_{AF}^{acc}, Π_{AF}^{acc} basically is adding constraints to the Dung's mapping (see Section 6). The authors in [11] showed that the constraints which are added by Π_{AF}^{acc} to the Dung's mapping do not affect neither the well-founded nor stable models semantics for characterizing the grounded and the stable semantics, respectively. In this sense, we can argue that Theorem 3 extends Theorem 62 from [17]. It is worth mentioning that Theorem 62 from [17] introduced the first relationships between argumentation semantics and logic programming semantics.

3 Ideal Sets as ideal models

The ideal semantics represents an extension of the grounded extension and is based on the preferred semantics (see Definition 10). On the other hand, the ideal semantics has shown to have important computational properties [20, 21]. Hence, we will explore the ideal semantics in terms of logic programming semantics with negation as failure. More accurately, we will prove that the ideal sets of an argumentation framework can be characterized in terms of both ideal models (Theorem 4, Theorem 5) and p-stable models (Corollary 1).

3.1 Ideal models

Let us start by studying the ideal extensions of an argumentation framework in terms of *logical models*. To this end, a class of logical programming models that are called *ideal models* will be defined. The definition of ideal models will be based on the p-stable semantics. We will show that ideal models are able to characterize the ideal sets of an argumentation framework.

For the sake of simplicity of the proofs, the formalization of the results will be based on the mapping Π_{AF}; however, at the end of the section, the main result is introduced in terms of Π_{AF}^{acc}.

Definition 13 *Let P be a normal logic program and $M \subseteq \mathcal{L}_P$. M is an ideal model of P if and only if M is a model of P and it contains every p-stable model of P.*

Let us observe that the definition of *an ideal model* is quite similar to the definition of *an ideal set* (see Definition 10). In particular, instead of talking about extensions (set of arguments), in Definition 13, we are talking in terms of logical models. On the other hand, instead of talking about preferred extensions in Definition 13, we are talking in terms of p-stable models. The main difference that we can observe is the contained relation (subset relation). In Definition 10, a given admissible set is an ideal set of AF if it is contained in every preferred extension of AF. In Definition 13, a

model of P is an ideal model if it contains every p-model of P. In an abuse of the language, we can say that the definition of an ideal model is a dual definition of an ideal set. These similarities will define a direct relationship between the ideal sets of a given argumentation framework AF and the ideal models of Π_{AF}. This relationship will be formalized in Theorem 4.

In order to illustrate Definition 13, let us consider the following example.

Example 4 *Let P be the following normal logic program*[6]*:*

$$
\begin{array}{ll}
def(a) \leftarrow \ not\ def(b). & def(a) \leftarrow def(a). \\
def(b) \leftarrow \ not\ def(a). & def(b) \leftarrow def(b). \\
def(c) \leftarrow \ not\ def(a). & def(c) \leftarrow def(b). \\
def(c) \leftarrow \ not\ def(b). & def(c) \leftarrow def(a). \\
def(c) \leftarrow \ not\ def(d). & def(c) \leftarrow def(c). \\
def(d) \leftarrow \ not\ def(c). & def(d) \leftarrow def(a), def(b), def(d).
\end{array}
$$

One can see that P has six 2-valued models: $S_1 = \{def(a), def(b), def(c), def(d)\}$, $S_2 = \{def(b), def(c), def(d)\}$, $S_3 = \{def(a), def(c), def(d)\}$, $S_4 = \{def(a), def(b), def(c)\}$, $S_5 = \{def(b), def(c)\}$ and $S_6 = \{def(a), def(c)\}$. Among these models, one can see that only S_5 and S_6 are p-stable models. Observe that, both S_5 and S_6 are subsets of S_1 and S_4. This means that S_1 and S_4 are the ideal models of P.

A basic property of Π_{AF} *w.r.t.* ideal models is that Π_{AF} always has at least one ideal model.

Lemma 1 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework. Π_{AF} always has an ideal model which is $\mathcal{L}_{\Pi_{AF}}$.*

Proof. Since every p-stable model M is a subset of $\mathcal{L}_{\Pi_{AF}}$, the lemma follows by the fact that $\mathcal{L}_{\Pi_{AF}}$ always is a model of Π_{AF}. ■

An important property of the ideal models is that they are able to characterize ideal sets via Π_{AF}. In order to show this property of the ideal models, let us show that the 2-valued models of Π_{AF} characterize the admissible sets of an argumentation framework AF.

Lemma 2 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework. S is a 2-valued model of Π_{AF} if and only if $AR \setminus \{a | def(a) \in S\}$ is an admissible set of AF.*

Proof. By Proposition 6 from [5], $S \subseteq AR$ is an admissible set of AF iff S is a model of the formula:

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

[6]Let us observe that the program P is basically the program Π_{AF} where AF is the argumentation framework introduced in Figure 1.

By the proof of Theorem 1 from [29], Π_{AF} is logically equivalent to $g(\beta(AF))$ where $g(T)$ denotes a theory obtained from T by replacing every occurrence of an atom x in T by $\neg def(x)$. Hence the lemma follows by Proposition 6 from [5] and the fact that Π_{AF} is logically equivalent to $g(\beta(AF))$.

■

Let us consider the following example in order to illustrate how the ideal sets of an argumentation framework can be characterized in terms of ideal models.

Example 5 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework, where AR := $\{a, b, c, d\}$ and attacks := $\{(a, b), (b, a), (a, c), (b, c), (c, d), (d, c)\}$ (see Figure 1). Π_{AF} corresponds to the program P of Example 4. We know that Π_{AF} has two ideal models which are: $Id_1 = \{def(a), def(b), def(c), def(d)\}\}$ and $Id_2 = \{def(a), def(b), def(c)\ \}$. One can see that:*

$$AR \setminus \{x | def(x) \in Id_1\} = \{\}$$
$$AR \setminus \{x | def(x) \in Id_2\} = \{d\}$$

Therefore, $\{\}$ and $\{d\}$ are the two ideal sets of AF.

Formally, we can characterize ideal sets in terms of ideal models as follows:

Theorem 4 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework. S is an ideal model of Π_{AF} if and only if $AR \setminus \{a | def(a) \in S\}$ is an ideal set of AF.*

Proof. Let us start introducing some notation: Let $M \subseteq \mathcal{L}_{\Pi_{AF}}$. $S_M = \{a | a \in AR$ and $def(a) \notin M\}$. Let $S \subseteq AF$. $M_S = \{def(a) | a \in AR$ and $a \notin S\}$. Now let us introduce a couple of observations:

1. If M_1 and M_2 are models of Π_{AF} such that $M_1 \subseteq M_2$, then $S_{M_2} \subseteq S_{M_1}$
2. If $S_1, S_2 \subseteq AR$ such that $S_1 \subseteq S_2$, then $M_{S_2} \subseteq M_{S1}$.

=> If M_{ID} is an ideal model of Π_{AF}, then $S_{M_{ID}}$ is an admissible set of AF (Lemma 2). Since M_{ID} is an ideal model, then for all $M \in P\text{-}stable(\Pi_{AF})$, $M \subseteq M_{ID}$. Therefore by Observation 1, for all $M \in P\text{-}stable(\Pi_{AF})$, $S_{M_{ID}} \subseteq S_M$. We know that if $M \in P\text{-}stable(\Pi_{AF})$, then S_M is a preferred extension of AF (Theorem 3). Hence, since $S_{M_{ID}}$ is an admissible set and is a subset of every preferred extension of AF, then $S_{M_{ID}}$ is an ideal set of AF.

<= If S is an ideal set of AF, then S is an admissible set of AF. Therefore M_S is a model of Π_{AF} (Lemma 2). Moreover for all $E \in Preferred(AF)$, $S \subseteq E$ such that $Preferred(AF)$ denotes the set of preferred extensions of AF. We know that if $E \in Preferred(AF)$, then M_E is a p-stable model of Π_{AF} (Theorem 3). Since for all $E \in Preferred(AF)$, $S \subseteq E$, then, by Observation 2, $M_E \subseteq M_S$. Therefore M_S is an ideal model of Π_{AF}.

■

Given that the construction of the ideal models is based on P-stable models, a direct consequence of Theorem 4 is that the ideal extensions of an argumentation framework can be characterized in terms of P-stable models.

Corollary 1 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework. If S is a model of Π_{AF} and $\forall M \in Pstable(\Pi_{AF}), M \subseteq S$ then $AR \setminus \{a|def(a) \in S\}$ is an ideal set of AF.*

We will finish this section by introducing the result of Theorem 4 in terms of Π^{acc}_{AF}. To this end, let us introduce the following lemma which was proved in [11].

Lemma 3 *[11] Let P be a normal logic program such that there exist P_1 and P_2 satisfying:*

1. *$P = P_1 \cup P_2$ and $P_1 \cap P_2 = \{\}$.*
2. *The atoms in the head of P_1 do not occur in P_2.*
3. *The atoms in the body of P_1 do not occur in the head of P_1.*

Then M is a p-stable model of P if and only if there exist M_1 and M_2 such that $M = M_1 \cup M_2$, M_2 is a p-stable model of P_2 and $M_1 = \{x|x \leftarrow \alpha \in P_1, M_2(\alpha) = 1\}$.

Theorem 5 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework. S is an ideal model of Π^{acc}_{AF} if and only if $\{x|acc(x) \in S\}$ is an ideal set of AF.*

Proof. By definition of ideal model (Definition 13) and Lemma 3, M is an ideal model of Π_{AF} if and only if M' is an ideal model of Π^{acc}_{AF} and $M = M' \cap \mathcal{L}_{\Pi_{AF}}$. Therefore, the result follows from Theorem 4.

■

4 Ideal Sets and the WFS$^+$ semantics

By considering the relationship between ideal models and ideal sets which is suggested by Theorem 4, in this section, we will explore the relationship between ideal sets and the WFS$^+$ model. One of the main results of this section will be that WFS$^+$ characterizes the maximal ideal set of an argumentation framework using the mapping Π_{AF} (Theorem 10). To this end, we will also show that WFS$^+$ characterizes the minimal ideal model of Π_{AF} (Theorem 8).

We start by considering a couple of examples in order to illustrate that WFS and WFS$^+$ are able to characterize ideal sets of an argumentation framework via the mapping Π_{AF}.

Example 6 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework, in which $AR := \{a, b, c\}$ and $attacks := \{(a, a), (a, b), (b, c), (c, b)\}$ (see Figure 4). Hence, Π_{AF} is:*

$$
\begin{array}{ll}
def(a) \leftarrow \ not\ def(a). & def(a) \leftarrow def(a). \\
def(b) \leftarrow \ not\ def(a). & def(b) \leftarrow \ not\ def(c). \\
def(b) \leftarrow def(a). & def(b) \leftarrow def(b). \\
def(c) \leftarrow \ not\ def(b). & def(c) \leftarrow def(c), def(a).
\end{array}
$$

Figure 4: Graph representation of $AF := \langle\{a, b, c\}, \{(a, a), (a, b), (b, c), (c, b)\}\rangle$.

We can see that the argumentation framework AF has a preferred extension which is $\{c\}$ and the grounded extension is empty. It is obvious that this argumentation framework has two ideal sets: $\{\}$ and $\{c\}$. Now let us see in the following table which ideal sets can be induced by WFS and WFS$^+$.

Semantics	*3-valued programming model*	*Ideal Set*
$WFS(\Pi_{AF})$	$\langle T = \{\}, F = \{\}\rangle$	$\{\} = \{x \mid def(x) \in F\}$
$WFS^+(\Pi_{AF})$	$\langle T = \{def(a), def(b)\}, F = \{def(c)\}\rangle$	$\{c\} = \{x \mid def(x) \in F\}$

Following the idea that the negative atoms of WFS and WFS$^+$ suggest (respectively) a potential set of acceptable arguments, one can see from this table that WFS and WFS$^+$ characterize different ideal sets. Indeed, WFS^+ is able to infer a non-empty ideal set; moreover, the ideal set suggested by WFS$^+$ is the maximal ideal set of the argumentation framework AF.

Strictly speaking, one can formalize a relationship between the maximal ideal set of an argumentation framework AF and the WFS^+ model of Π_{AF} using *the minimal ideal model* of Π_{AF}. In order to illustrate the important role of the minimal ideal model of Π_{AF} for characterizing the maximal ideal set of an argumentation framework, let us consider the following example:

Example 7 *Let $AF := \langle AR, attacks\rangle$ be an argumentation framework in which $AR := \{a, b, c, d, e, f, x, y\}$ and $attacks := \{(a, b), (b, a), (a, c), (b, c), (c, d), (d, e), (e, f), (x, y)\}$ (see Figure 5).*

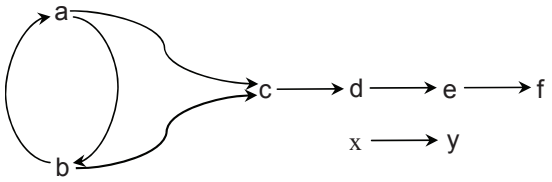

Figure 5: Graph representation of $AF := \langle\{a, b, c, d, e, f, x, y\}, \{(a, b), (b, a), (a, c), (b, c), (c, d), (d, e), (e, f), (x, y)\}\rangle$.

One can see that Π_{AF} is:

$def(a) \leftarrow not\ def(b).$ $\quad def(a) \leftarrow def(a).$
$def(b) \leftarrow not\ def(a).$ $\quad def(b) \leftarrow def(b).$
$def(c) \leftarrow not\ def(a).$ $\quad def(c) \leftarrow def(b).$
$def(c) \leftarrow not\ def(b).$ $\quad def(c) \leftarrow def(a).$
$def(d) \leftarrow not\ def(c).$ $\quad def(d) \leftarrow def(a), def(b).$
$def(e) \leftarrow not\ def(d).$ $\quad def(e) \leftarrow def(c).$
$def(f) \leftarrow not\ def(e).$ $\quad def(f) \leftarrow def(d).$
$def(y) \leftarrow not\ def(x).$ $\quad def(y) \leftarrow \top.$

Hence, one can see that the normal form of Π_{AF} *w.r.t.* CS_1 *is the following normal logic program* $norm_{CS_1}(\Pi_{AF})$*:*

$def(a) \leftarrow not\ def(b).$ $\quad def(f) \leftarrow def(d).$
$def(b) \leftarrow not\ def(a).$ $\quad def(d) \leftarrow def(a), def(b).$
$def(c) \leftarrow \top.$ $\quad def(e) \leftarrow \top.$
$def(y) \leftarrow \top.$

One can notice that $P\text{-}stable(\Pi_{AF}) = \{\{def(c), def(e), def(y), def(a)\}, \{def(c), def(e), def(y), def(b)\}\}$*. On the other hand, from* $norm_{CS_1}(P)$*, one can see that* $WFS^+(\Pi_{AF}) = \langle\{def(c), def(e), def(y)\}, \{def(x)\}\rangle$[7]*.*

According to WFS^+*, the set of atoms* $I = \{def(a), def(b), def(d), def(f)\}$ *identifies the atoms which are considered undefined*[8]*. In order to clarify some properties of the elements of* I*, let us split* I *into two sets of atoms:*

- $I_1 = \{def(a), def(b)\}$ *such that* I_1 *contains the atoms which appear in the head of a rule* $r \in norm_{CS_1}(\Pi_{AF})$ *of the form* $x \leftarrow not\ y$*.*
- $I_2 = \{def(d), def(f)\}$ *such that* $I_2 = I \setminus I_1$*.*

From the WFS^+ *model of* Π_{AF} *and* $P\text{-}stable(\Pi_{AF})$*, one can observe that:*

1. $\{def(c), def(e), def(y)\}$ *is a subset of the minimal ideal model of* Π_{AF}*.*
2. *Each atom of* I_1 *belongs to a p-stable model of* Π_{AF}*; hence,* I_1 *is also a subset of the minimal ideal set of* Π_{AF}*.*
3. *Hence, by the previous observations and the fact that each p-stable model of* Π_{AF} *is a subset of every ideal model* Π_{AF}*,* $\{def(c), def(e), def(y), def(a), def(b)\}$ *is also a subset of the minimal ideal model of* Π_{AF}*.*

Now, let us split I_2 *into two sets of atoms:*

- $I_2^* = \{def(d)\}$ *such that* I_2^* *contains the atoms which appear as no-negated in the bodies of the clauses of* $norm_{CS_1}(\Pi_{AF})$ *and do not belong to* I_1

[7] Let us observe that the atom $def(x)$ does not appear in $norm_{CS_1}(\Pi_{AF})$; however, $def(x)$ belongs to the signature of Π_{AF}. Hence $def(x)$ is considered as an atom of the signature of $norm_{CS_1}(\Pi_{AF})$.

[8] Let us observe that the elements of I belong to neither the true atoms nor the false atoms of $WFS^+(\Pi_{AF})$.

- $I_2^{**} = \{def(f)\}$ *such that* $I_2^{**} = I_2 \setminus I_2^*$. *Observe that the elements of* I_2^{**} *only appear in the head of definite clauses.*

Now let us observe that since $Loop \in CS_1$ *then Loop cannot be applied to* $norm_{CS_1}(\Pi_{AF})$. *Let us remember that Loop is a transformation rule which looks for atoms which appear in the bodies of positive clauses that can be considered false. Hence if we cannot apply Loop to* $norm_{CS_1}(\Pi_{AF})$, *it means that none of the positive atoms which appear in the bodies of the positive clauses can be false. Hence the elements of* I_2^* *cannot be false in the minimal ideal model of* Π_{AF} *by the assumption that the minimal ideal model of* Π_{AF} *contains* I_1. *Hence,* I_2^* *is a subset of the minimal ideal model of* Π_{AF}. *By the definition of* I_2^{**}, *the elements of* I_2^{**} *have to be true if the atoms of* I_1 *and* I_2^* *are true in the minimal ideal model of* Π_{AF}. *Therefore,* I_2^{**} *also belongs to the minimal ideal set of* Π_{AF}.

Hence, one can see that $\{def(c), def(e), def(y)\} \cup I$ *is* the minimal ideal set *of* Π_{AF}. *This means that the atoms which are fixed to false by WFS*$^+$ *define* a maximal set. *Hence by Theorem 4, one can see that* $\{x\}$ *is the maximal ideal set of the argumentation framework* AF.

In the following sections, we will formalize the intuitive ideas introduced in Example 7. In particular the relationship between the minimal ideal model of Π_{AF} and the maximal ideal set of AF will be explored.

4.1 Ideal Models and the normal form of Π_{AF}

The aim of this subsection is to show that the minimal ideal model of the normal form Π_{AF} *w.r.t.* CS_1 can be characterized using WFS^+ (Theorem 6).

Given that the definition of the ideal models is based on p-stable models, we will show some properties of the p-stable semantics *w.r.t.* CS_1. Hence, let us show that the p-stable semantics is preserved with respect to the transformation rules which belong to CS_1.

Lemma 4 *Let* $AF = \langle AR, attacks\rangle$ *be an argumentation framework.* M *is a p-stable model of* Π_{AF} *iff* M *is a p-stable model of* $norm_{CS_1}(\Pi_{AF})$.

Proof. The proof is direct by Proposition 3 to Proposition 9 of [12]. These propositions show that the p-stable semantics is preserved with respect to each transformation rule which belongs to CS_1. ■

An important transformation rule which belongs to CS_1 is classical logical implication (LC). One can see that if an atom is inferred from a given logic program P, it can be inferred from the normal form of P with respect to CS_1.

Lemma 5 *Let* P *be a normal logic programs and* $a \in \mathcal{L}_P$. *If* $P \vdash a$ *then* $norm_{CS_1}(P) \vdash a$.

Proof. The proof is direct by the construction of $norm_{CS_1}(P)$ and the fact that $LC \in CS_1$. ■

A relevant property *w.r.t.* the p-stable models of Π_{AF} and the minimal models of Π_{AF} is that both sets of models coincide.

Lemma 6 *Let $AF = \langle AR, attacks\rangle$ be an argumentation framework. M is a minimal model of Π_{AF} iff M is a p-stable model of Π_{AF}.*

Proof. By Theorem 1 of [29], M is a minimal model of Π_{AF} iff $AR \setminus \{a | def(a) \in M\}$ is a preferred extension of AF. By Theorem 6 of [11], $AR \setminus \{a | def(a) \in M\}$ is a preferred extension of AF iff M is a p-stable model of Π_{AF}. Therefore, Theorem 1 of [29] and Theorem 6 of [11], M is a minimal model of Π_{AF} iff M is a p-stable model of Π_{AF}. ■

As we saw in Example 7, the rules $def(a) \leftarrow not\ def(b)$ and $def(b) \leftarrow not\ def(a)$ belong to $norm_{CS_1}(\Pi_{AF})$. On the other hand, we also observed that $def(a)$ and $def(b)$ belong (respectively) to a p-stable model of $P\text{-}stable(norm_{CS_1}(\Pi_{AF}))$. This property can be formalized as follows:

Lemma 7 *Let AF be an argumentation framework. If $x \leftarrow\ not\ y \in norm_{CS_1}(\Pi_{AF})$ then $\exists M \in P\text{-}stable(norm_{CS_1}(\Pi_{AF}))$ such that $x \in M$.*

Proof. Given that $x \leftarrow\ not\ y \in norm_{CS_1}(\Pi_{AF})$ and the transformation rule $LC \in CS_1$, one cannot apply LC to $norm_{CS_1}(\Pi_{AF})$ and $norm_{CS_1}(\Pi_{AF}) \nvdash y$[9]. Moreover, by Lemma 5, $P \nvdash y$. Hence, there is a model N of P such that y is false *w.r.t.* N. Then there is a minimal model M of P such that y is false *w.r.t.* M. By Lemma 6, the minimal models of Π_{AF} coincide with $P\text{-}stable(\Pi_{AF})$, then M is a p-stable model of Π_{AF}. By Lemma 4, we know that M' is a p-stable model of Π_{AF} iff M' is a p-stable model of $norm_{CS_1}(\Pi_{AF})$; then, M is a p-stable model of $P\text{-}stable(norm_{CS_1}(\Pi_{AF}))$. Therefore, since y is false *w.r.t.* M then x is true *w.r.t.* M. ■

By considering the previous results, we can show that the minimal ideal model of the normal form Π_{AF} *w.r.t.* CS_1 can be characterized via WFS^+.

Theorem 6 *Let $AF = \langle AR, attacks\rangle$ be an argumentation framework, $WFS^+(\Pi_{AF}) = \langle Tr, Fl\rangle$, $I = \{def(x) | x \in AR\} \setminus (Tr \cup Fl)$. If $TrI = Tr \cup I$, then TrI is the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$.*

Proof. Let us observe that $TrI = \mathcal{L}_{norm_{CS_1}(\Pi_{AF})}$ such that $\mathcal{L}_{norm_{CS_1}(\Pi_{AF})}$ is the set of atoms which appear in $norm_{CS_1}(\Pi_{AF})$. Therefore, we will show that each atom in $\mathcal{L}_{norm_{CS_1}(\Pi_{AF})}$ must belong to the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$. One can see that there are four kinds of atoms in $norm_{CS_1}(\Pi_{AF})$:

1. Atoms which are facts: An atom a is a fact in $norm_{CS_1}(\Pi_{AF})$ if $a \leftarrow \top \in norm_{CS_1}(\Pi_{AF})$; hence, for all $M \in P\text{-}stable(norm_{CS_1}(\Pi_{AF}))$, $a \in M$. Therefore, a must belong to the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$.

2. Atoms x which appear in $x \leftarrow\ not\ y \in norm_{CS_1}(\Pi_{AF})$. By Lemma 7, there is $M \in P\text{-}stable(norm_{CS_1}(\Pi_{AF}))$ such that $x \in M$. Therefore, x belongs to the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$.

[9]Let us observe that if it was the case that $norm_{CS_1}(\Pi_{AF}) \vdash y$, then by LC,$y \leftarrow \top \in norm_{CS_1}(\Pi_{AF})$ and RED^- have removed $x \leftarrow\ not\ y$ from $norm_{CS_1}(\Pi_{AF})$.

3. Atoms x which appear positively in the body of a clause $r \in norm_{CS_1}(\Pi_{AF})$. Given that the transformation rule Loop cannot be applied to $norm_{CS_1}(\Pi_{AF})$, x does not belong to the greatest unfounded set of Π_{AF}. Therefore x belongs to the minimal model of $Poss(norm_{CS_1}(\Pi_{AF}))$ such that $Poss(norm_{CS_1}(\Pi_{AF}))$ is the definite logic program obtained from $norm_{CS_1}(\Pi_{AF})$ by deleting each negative literal which appears in the bodies of the clauses $r \in norm_{CS_1}(\Pi_{AF})$. Therefore, by this observation and case 2, it is clear that x belongs to the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$.

4. The remainder of atoms x which do not fall into the previous cases are atoms which occur in the head of positive clauses $x \leftarrow body \in norm_{CS_1}(\Pi_{AF})$. One can see that *body* only contains atoms which fall in case 2 and case 3; hence, x must belong to the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$.

From the previous four observations, for all $x \in \mathcal{L}_{norm_{CS_1}(\Pi_{AF})}$, x belongs to the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$. Moreover, clearly no other atom belongs to the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$. Therefore, TrI is the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$. ■

4.2 Ideal Models between Π_{AF} and $norm_{CS_1}(\Pi_{AF})$

Now in this section, we will show that the ideal models of Π_{AF} are closed *w.r.t.* CS_1. This means that if M is an ideal model of Π_{AF}, then M is an ideal model of $norm_{CS_1}(\Pi_{AF})$ (Theorem 7).

We start by showing an important property of $norm_{CS_1}(P)$ *w.r.t.* P. Specially, we will show that the signature of $norm_{CS_1}(P)$ (the set of atoms which appears in $norm_{CS_1}(P)$) defines a model of the logic program P.

Lemma 8 *Let P be a normal logic program. If L is the set of atoms which occur in $norm_{CS_1}(P)$, then L is a model of P.*

Proof. The proof is by induction *w.r.t.* the number of steps n (applications of transformation rules) for getting the normal form of P *w.r.t.* CS_1:

Base case: If $n = 0$, then P is in its normal form. Hence, it is trivial that L is a model of $norm_{CS_1}(P)$. Therefore L is a model of P, by the fact that $P = norm_{CS_1}(P)$.

Inductive step: Let us suppose that P is transformed by one step to P_1. Since $norm_{CS_1}(P) = norm_{CS_1}(P_1)$ and L is a model of $norm_{CS_1}(P)$, then L is a model of $norm_{CS_1}(P_1)$ and, by inductive hypothesis, L is a model of P_1. By verifying each transformation rule $r \in CS_1$, one can confirm that L is a model of P. In particular, the relevant transformation rule to verify is the loop transformation rule: let P_1 be the resulting program by applying the transformation rule loop to P such that P^-_{loop} is the set of clauses which were deleted by loop. One can see that each clause $r \in P^-_{loop}$ has a positive atom a in its body such that $a \notin HEAD(P_1)$. Therefore, a does not occur in $norm_{CS_1}(P_1)$. Then, a is false *w.r.t.* L. Hence,

the body of r is false *w.r.t.* L. Then r is true *w.r.t.* L. Therefore P^{-}_{loop} is true *w.r.t.* L. Hence L is a model of $P_1 \cup P^{-}_{loop}$. Then L is a model of P.

■

Now let us show that $WFS^{+}(\Pi_{AF})$ defines an ideal model of Π_{AF}.

Lemma 9 *Let $AF = \langle AR, attacks\rangle$ be an argumentation framework, $WFS^{+}(\Pi_{AF}) = \langle Tr, Fl\rangle$, $I = f(AR) \setminus (Tr \cup Fl)$. If $TrI = Tr \cup I$, then TrI is an ideal model of Π_{AF}.*

Proof. By Theorem 6, TrI is an ideal model of $norm_{CS_1}(\Pi_{AF})$. By Lemma 8, TrI is a model of Π_{AF}. By Lemma 4, M is a p-stable model of Π_{AF} iff M is a p-stable model of $norm_{CS_1}(\Pi_{AF})$. Since TrI is an ideal model of $norm_{CS_1}(\Pi_{AF})$, then every model M which is a p-stable model of $norm_{CS_1}(\Pi_{AF})$, $M \subset TrI$. Therefore, TrI is an ideal model of Π_{AF}.

■

In the following lemma, we will show that the models of Π_{AF} are closed *w.r.t.* the transformation rules of CS_1.

Lemma 10 *Let $AF = \langle AR, attacks\rangle$ be an argumentation framework. If M is a model of Π_{AF}, then M is a model of $norm_{CS_1}(\Pi_{AF})$.*

Proof. By definition, $CS_1 = \{RED^{+}, RED^{-}, Success, Failure, Loop, SUB, TAUT, LC\}$. Let $CS'_1 = CS_1 \setminus \{RED^{+}\}$. If $t \in CS'_1$ and Π^{t}_{AF} is the resulting program after applying t, then the following observation holds:

Observation 1: If M is a model of Π_{AF} then M is a model of Π^{t}_{AF}. Notice that the transformation rules RED^{-}, $Failure$, $Loop$, SUB, $TAUT$ mainly remove rules from Π_{AF}; therefore, the observation is straightforward *w.r.t.* these transformation rules. On the other hand, the transformation rules $Success$ and LC define logically equivalent theories; hence, the observation also holds with *w.r.t.* $Success$ and LC.

Formally, the proof is by induction *w.r.t.* the number of the steps (applications of transformation rules) for getting the normal form of Π_{AF}; however, the proof is trivial and by Observation 1, the main transformation to see is the RED^{+} transformation. Hence, the proof is reduced to prove that $norm_{CS_1}(\Pi_{AF}) = norm_{CS'_1}(\Pi_{AF})$. In order to prove this equality, we can see that the following observation holds:

Observation 2: RED^{+} cannot be applied to $norm_{CS'_1}(\Pi_{AF})$.

Hence, by Observation 2 and the fact that it is known that CS_1 is a confluent and terminating rewriting system, $norm_{CS_1}(\Pi_{AF}) = norm_{CS'_1}(\Pi_{AF})$. ■

Now we will show that the ideal models of Π_{AF} are closed *w.r.t.* CS_1.

Theorem 7 *Let AF be an argumentation framework. If M is an ideal model of Π_{AF}, then M is an ideal model of $norm_{CS_1}(\Pi_{AF})$.*

Proof. Let M be an ideal model of Π_{AF}. Hence, by Definition 13, M is a model of Π_{AF} and $\forall M' \in P\text{-}stable(\Pi_{AF})$, $M' \subseteq M$. By Lemma 4, M is a model of Π_{AF}, $\forall M' \in P\text{-}stable(norm_{CS_1}(\Pi_{AF}))$ and $M' \subseteq M$. By Lemma 10, M is a model of $norm_{CS_1}(\Pi_{AF})$ and $\forall M' \in P\text{-}stable(norm_{CS_1}(\Pi_{AF}))$ and $M' \subseteq M$. Therefore, M is an ideal model of $norm_{CS_1}(\Pi_{AF})$. ■

In accordance with Theorem 7, we can see that an ideal model of Π_{AF} is also an ideal model of $norm_{CS_1}(\Pi_{AF})$; however, the reverse of the implication is not true. To illustrate this observation, let us consider the following example:

Example 8 *Let $AF = \langle\{a, b, c\}, \{(a, b), (b, c)\}\rangle$ be an argumentation framework. Hence, Π_{AF} is:*

$$def(b) \leftarrow \top. \qquad def(c) \leftarrow not\ def(b). \quad def(c) \leftarrow def(a).$$

One can see that $norm_{CS_1}(\Pi_{AF})$ is:

$$def(b) \leftarrow \top.$$

Now let us observe that Π_{AF} has three ideal models:

$$\{def(b)\} \quad \{def(b), def(c)\} \quad \{def(a), def(b), def(c)\}$$

On the other hand, $norm_{CS_1}(\Pi_{AF})$ has four ideal models:

$$\{def(b)\} \quad \{def(a), def(b)\} \quad \{def(b), def(c)\} \quad \{def(a), def(b), def(c)\}$$

It is clear that the ideal models of Π_{AF} are ideal models of $norm_{CS_1}(\Pi_{AF})$; however, $norm_{CS_1}(\Pi_{AF})$ has an ideal model which is not a model of Π_{AF}, i.e. *, $\{def(a), def(b)\}$.*

4.3 The maximal ideal set using Π_{AF}

So far, we have seen that $WFS^+(\Pi_{AF})$ characterizes the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$ (Theorem 6). On the other hand, we have seen that the ideal models of Π_{AF} are closed *w.r.t.* CS_1 (Theorem 7). These results will be fundamental in proving the main result of this section: $WFS^+(\Pi_{AF})$ characterizes the maximal ideal set of the argumentation framework AF (Theorem 10).

Let us start by showing that $WFS^+(\Pi_{AF})$ also characterizes the minimal ideal set of Π_{AF}.

Theorem 8 *Let AF be an argumentation framework, $WFS^+(\Pi_{AF}) = \langle Tr, Fl\rangle$, $I = \mathcal{L}_{\Pi_{AF}} \setminus (Tr \cup Fl)$. If $TrI = Tr \cup I$, then TrI is the minimal ideal model of Π_{AF}.*

Proof. By Theorem 6, TrI is the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$ and, by Lemma 9, it is an ideal model of Π_{AF}. Hence, we will prove that TrI is the minimal ideal model of Π_{AF}. The proof is by contradiction. Let us suppose that there is $M \subset TrI$ such that M is an ideal model of Π_{AF}. By Theorem 7, M is an ideal model of

$norm_{CS_1}(\Pi_{AF})$. Since M is a strict subset of TrI, then TrI is not the minimal ideal model of $norm_{CS_1}(\Pi_{AF})$. This is a contradiction *w.r.t.* Theorem 6. ■

Given that the p-stable semantics extends logic programming semantics such as WFS, WFS+, one can formalize a relationship between the ideal models and WFS, WFS+.

Theorem 9 *Let AF be an argumentation framework and $M \subseteq \mathcal{L}_{\Pi_{AF}}$. If M is an ideal model of Π_{AF}, then the following conditions hold:*

a. *if $WFS^+(\Pi_{AF}) = \langle Tr, Fl \rangle$, then $Tr \subseteq M$.*

b. *if $WFS(\Pi_{AF}) = \langle Tr, Fl \rangle$, then $Tr \subseteq M$.*

Proof.

a. By Theorem 8, we know that if $I = \mathcal{L}_{\Pi_{AF}} \setminus (Tr \cup Fl)$, then $TrI = Tr \cup I$ is the minimal ideal model of Π_{AF}. Hence, $Tr \subseteq TrI$ and $TrI \subseteq M$. Therefore, $Tr \subseteq M$.

b. It is known that $WFS(P) \leq_k WFS^+(P)$; hence, the proof is direct by a.

■

A couple of obvious relationships between the WFS (and WFS$^+$) model of Π_{AF} with ideal sets are the following:

Lemma 11 *Let $AF := \langle AR, attacks \rangle$ be an argumentation framework.*

1. *if $WFS(\Pi_{AF}) = \langle Tr, Fl \rangle$, then $\{x | d(x) \in Fl\}$ is an ideal set of AF.*
2. *if $WFS^+(\Pi_{AF}) = \langle Tr, Fl \rangle$, then $\{x | d(x) \in Fl\}$ is an ideal set of AF.*

Proof. (1) It follows by Theorem 7 from [11] which proves that WFS and Π_{AF} characterize the grounded extension and the fact that the grounded extension is an ideal set. (2) It is direct by Theorem 4 and Theorem 8. ■

A straightforward implication of Theorem 4 and Theorem 8 is the characterization of the maximal ideal set via WFS$^+$.

Theorem 10 *Let AF be an argumentation framework. If $WFS^+(\Pi_{AF}) = \langle Tr, Fl \rangle$, then $S = \{a | def(a) \in Fl\}$ is the maximal ideal set of AF.*

Proof. The proof is direct by Theorem 4 and Theorem 8. ■

5 The maximal ideal set using Π^{acc}_{AF}

In Section 4, we have shown that WFS^+ using Π_{AF} characterizes the maximal ideal set of the argumentation framework AF. In this section, we present the results of Theorem 10 in terms of the mapping Π^{acc}_{AF}. To this end, let us start by introducing an example.

Example 9 *Let AF be the argumentation framework which was introduced in Example 6. As we saw in Example 6, Π_{AF} is:*

$$\begin{array}{ll} def(a) \leftarrow \ not\ def(a). & def(a) \leftarrow def(a). \\ def(b) \leftarrow \ not\ def(a). & def(b) \leftarrow \ not\ def(c). \\ def(b) \leftarrow def(a). & def(b) \leftarrow def(b). \\ def(c) \leftarrow \ not\ def(b). & def(c) \leftarrow def(c), def(a). \end{array}$$

Hence, Π^{acc}_{AF} is Π_{AF} union with the following set of clauses:

$$\begin{array}{l} acc(a) \leftarrow \ not\ def(a). \\ acc(b) \leftarrow \ not\ def(b). \\ acc(c) \leftarrow \ not\ def(c). \end{array}$$

One can see that $norm_{CS_1}(\Pi^{acc}_{AF})$ is:

$$\begin{array}{l} def(a) \leftarrow \top. \\ def(b) \leftarrow \top. \\ acc(a) \leftarrow \top. \end{array}$$

Hence, $WFS^+(\Pi^{acc}_{AF}) = \langle\{def(a), def(b), acc(c)\}, \{acc(a), acc(b), def(c)\}\rangle$. By considering the atoms of the form $acc(X)$ which belong to the true atoms of $WFS^+(\Pi^{acc}_{AF})$, we can see that the set $\{acc(c)\}$ is characterizing an ideal set. Indeed, we can see that this set is characterizing the maximal ideal set of AF which is $\{c\}$. This means that $WFS^+(\Pi^{acc}_{AF})$ characterizes the maximal ideal set of an argumentation framework. This property of $WFS^+(\Pi^{acc}_{AF})$ will be formalized in the following theorem.

Theorem 11 *Let AF be an argumentation framework. If $WFS^+(\Pi^{acc}_{AF}) = \langle Tr, Fl\rangle$, then $S = \{a | acc(a) \in Tr\}$ is the maximal ideal set of AF.*

Proof. By definition:

$$\Pi^{acc}_{AF} := \Pi_{AF} \cup \bigcup_{a \in AR} \{acc(a) \leftarrow \ not\ def(a)\}$$

Since CS_1 is a confluent rewriting system and there is a stratification in Π^{acc}_{AF}, $norm_{CS_1}(\Pi^{acc}_{AF}) = norm_{CS_1}(norm_{CS_1}(\Pi_{AF}) \cup \bigcup_{a \in AR}\{acc(a) \leftarrow \ not\ def(a)\})$.

In order to infer $norm_{CS_1}(\Pi^{acc}_{AF})$, we only have to reduce each rule of the form $acc(a) \leftarrow \ not\ def(a)$ by considering the grounded atom $def(a)$ *w.r.t.* $norm_{CS_1}(\Pi_{AF})$ and CS_1. There are three cases:

1. if $def(a)$ does not occur in $HEAD(norm_{CS_1}(\Pi_{AF}))$, then the rule $acc(a) \leftarrow \ not\ def(a)$ is reduced to $acc(a) \leftarrow \top$ by RED^+. This means $acc(a) \leftarrow \top \in norm_{CS_1}(\Pi^{acc}_{AF})$.

2. if $def(a) \leftarrow \top \in norm_{CS_1}(\Pi_{AF})$, then the rule $acc(a) \leftarrow \ not\ def(a)$ is removed by RED^-. This means $acc(a) \notin HEAD(norm_{CS_1}(\Pi^{acc}_{AF}))$.

3. if $def(a) \leftarrow body \in HEAD(norm_{CS_1}(\Pi_{AF}))$ such that $body \neq \top$, then $acc(a) \leftarrow \ not\ def(a) \in norm_{CS_1}(\Pi^{acc}_{AF})$.

Let $SEM_{CS_1}(\Pi_{AF}) = \langle Tr_1, Fl_1 \rangle$, $SEM_{CS_1}(\Pi_{AF}^{acc}) = \langle Tr_2, Fl_2 \rangle$, $S_1 = \{a | def(a) \in Fl_1\}$ and $S_2 = \{a | acc(a) \in Tr_2\}$. From the previous analysis, we can observe that $S_1 = S_2$. By Theorem 10, S_1 is the maximal ideal set of AF; therefore, S_2 is the maximal ideal set of AF. ■

6 Related Work

In this section, we are going to identify some of the approaches, in the argumentation research literature, which are closely related to the results presented in this paper[10].

6.1 Argumentation Frameworks as Logic Programs

We start with the approaches which regard argumentation as logic programming. The usual way to study argumentation as logic programming is to map argumentation frameworks into logic programs (also called logical theories). Currently, we can find different mappings of argumentation frameworks into logical theories. Dung introduces the following basic meta-interpreter (or mapping) in terms of logic programs with negation as failure:

(C1) $acc(X) \leftarrow not\ def(X).$

(C2) $def(X) \leftarrow attack(Y, X), acc(Y).$

By using this mapping, Dung characterized the grounded and the stable semantics in terms of the well-founded and stable model semantics respectively. This mapping basically is the first mapping which was suggested for regarding argumentation frameworks as logic programs. An obvious question is: what is the relationship between Dung's mapping and the mapping introduced by Definition 11? In order to illustrate this relation, let us consider the argumentation framework introduced by Figure 1. By considering Dung's mapping and the argumentation framework of Figure 1, we can get the following grounded program:

$acc(a) \leftarrow not\ def(a).$ $\quad$ $acc(b) \leftarrow not\ def(b).$
$acc(c) \leftarrow not\ def(c).$ $\quad$ $acc(d) \leftarrow not\ def(d).$
$def(b) \leftarrow acc(a).$ $\quad$ $def(a) \leftarrow acc(b).$
$def(c) \leftarrow acc(a).$ $\quad$ $def(c) \leftarrow acc(b).$
$def(c) \leftarrow acc(d).$ $\quad$ $def(d) \leftarrow acc(c).$

If we apply *partial evaluation* to this program, we get:

$acc(a) \leftarrow not\ def(a).$ $\quad$ $acc(b) \leftarrow not\ def(b).$
$acc(c) \leftarrow not\ def(c).$ $\quad$ $acc(d) \leftarrow not\ def(d).$
$def(b) \leftarrow not\ def(a).$ $\quad$ $def(a) \leftarrow not\ def(b).$
$def(c) \leftarrow not\ def(a).$ $\quad$ $def(c) \leftarrow not\ def(b).$
$def(c) \leftarrow not\ def(d).$ $\quad$ $def(d) \leftarrow not\ def(c).$

[10] We apologize if we omit a relevant reference which could be important for this section.

As we can see, this program is basically a subprogram of the program Π_{AF}^{acc} which was introduced in the introduction section. Indeed, we can say that the mapping introduced by Definition 11 adds more constraints to Dung's mapping. More accurately, Dung's mapping is only aware of *conflict-freeness*, *i.e.* an extension of an argumentation semantics cannot include conflicting arguments; on the other hand, the mapping of Definition 11 is aware of both conflict-freeness and *reinstatement* (*i.e.* the arguments defeated by an extension E (set of acceptable arguments) play no role in the selection of arguments to be included in E). In general terms, we can say that the mapping of Definition 11 is an extension of Dung's mapping. It is worth mentioning that if we consider a mapping $\mathcal{M}$ from an argumentation framework into a logic program and two argumentation frameworks AF_1 and AF_2, $\mathcal{M}$ is called *modular* iff $\mathcal{M}(AF_1 \cup AF_2) = \mathcal{M}(AF_1) \cup \mathcal{M}(AF_2)$. In this setting, it is obvious that the Dung's mapping is modular. On the other hand, Π_{AF}^{acc} is not modular; however, since Π_{AF}^{acc} is basically an extension of Dung's mapping, the part of Π_{AF}^{acc} which coincides with Dung's mapping is also modular.

Besnard and Doutre introduced a set of mappings of an argumentation framework into propositional theories [5]. By using maximal and minimal 2-valued models of the resulting propositional theories, they characterized Dung's argumentation semantics introduced in [17]. Currently there are several meta-interpreters in terms of answer set programs [22, 37]. These meta-interpreters allow us to compute different argumentation semantics in terms of answer sets. These meta-interpreters do not include ideal semantics; however, in [23], an approach for computing ideal semantics in terms of answer sets was introduced. It is worth mentioning that by considering answer set programs, one can infer argumentation semantics which are not based on admissible sets such as CF2 [34].

In [10, 38], the authors have explored the relationship between argumentation and logic programming. Unlike our approach, which makes direct relationships between logical models and extensions of argumentation semantics, the authors in [10, 38] use a labeling approach in which extensions of argumentation semantics are expressed in terms of 3-valued labellings; after this, the 3-valued labellings are expressed in terms of 3-valued logical models. In this approach, argumentation frameworks are also transformed into logic programs. For instance in [38], the authors use the following mapping: Given an argumentation framework $AF := \langle AR, attacks \rangle$:

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

This mapping was first introduced in [31] in order to show that the answer sets of P_{AF} correspond to the stable extensions of AF (see Theorem 1 of [31]). In [38], the authors showed that the complete semantics can be characterized in terms of the 3-valued stable semantics and P_{AF}. This mapping was also explored by Gabbay in order to map argumentation frameworks into logic programs [26].

From the declarative point of view, the transformation P_{AF} only specifies a basic specification of *why an argument can belong to an extension of an argumentation semantics*. Indeed, like Dung's mapping, P_{AF} is only capturing the idea of *conflict-freeness*. On the other hand, the transformation introduced by Definition 11 specifies

why an argument cannot belong to an extension of an argumentation framework; moreover, this transformation captures two basic ideas which must satisfy an admissible set: *conflict-freeness* and *reinstatement*.

Understanding argumentation semantics from a logic programming point of view depends on two variables:

1. The declarative specification of an argumentation framework in terms of logic programs and
2. The logic programming semantics which infers the given argumentation semantics.

For instance in [38], the authors were able to characterize the complete semantics by using a specification of only conflict-freeness and the 3-valued stable model semantics. In this setting, it seems that the 3-valued stable model semantics does not require the reinstatement principle in order to capture the complete semantics. In contrast, the Clark's Completion Semantics requires both conflict-freeness and reinstatement principles in order to capture the complete semantics [35]. Indeed, we know that the answer set semantics also only require conflict-freeness for characterizing the stable semantics [31]; however, we also already know that the answer set semantics require both *conflict-freeness* and *reinstatement* for inferring the preferred semantics [29].

Comparing P_{AF} and the mapping introduced by Definition 11, what can we observe? First, we want to point out that all the 2-valued models of Π_{AF} characterize the admissible sets of AF (see Lemma 2). Hence, regardless of the logic programming semantics which we use, Π_{AF} will characterize argumentation semantics based on admissible sets. Regarding P_{AF}, we can observe that the models of P_{AF} only characterize the conflict free set of AF. Therefore, the characterization of the admissible sets of an argumentation framework strongly depends on the logic programming semantics which is used for consulting P_{AF}. Having mappings whose models are able to characterize conflict-free sets gives place to characterizing argumentation semantics such as CF2. For instance, by considering only the first part of the mapping introduced in Definition 11:

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

the authors in [30] were able to characterize CF2 by using a logic programming semantics based on minimal models. At this point, let us observe that by considering basic principles, *i.e. conflict-freeness* and *reinstatement*, in a declarative specification of an argumentation frameworks, we can explore *pseudo-argumentation semantics*. By pseudo-argumentation semantics, we mean argumentation semantics which do not have a definition in terms of Dung's style; however, these semantics can be motivated by a particular specification and a given logic programming semantics. For example, in [33], the authors introduced the so called *stable-abducible argumentation semantics*, which is an intermediate semantics between stable semantics and preferred semantics. The relevance of this new semantics is that it is always defined for any argumentation framework, and coincides with the stable argumentation semantics whenever this is non-empty. Moreover, it satisfies the notion of *relevance* introduced by Caminada [8].

These observations point out the importance of both the transformation of an argumentation framework into a logic program and the logic programming semantics which are used for understanding a given argumentation semantics from a logic programming point of view.

Dunne showed that the ideal semantics has a high computational complexity [20]; hence, to identify computational algorithms which can compute the ideal semantics is a relevant research issue in order to use the ideal semantics in real applications. Due to the fact that WFS$^+$ is an extension of WFS, one can use WFS' algorithms for computing WFS$^+$. As we observed in Section 2.3, the main difference between WFS and WFS$^+$ is isolated by the transformation rules: SUB, $TAUT$, LC. From a naive point of view, one can observe that both SUB and $TAUT$ can be implemented in an efficient and straightforward way; however, LC is a computationally expensive transformation due to the fact that it depends on the computational cost of classical logical inference. In this setting, let us point out that, currently, classical logical inference can be computed by different logic-based algorithms, *e.g.,* SAT algorithms, Hyper-resolution-based algorithms. Therefore, the characterization of the ideal semantics in terms of WFS$^+$ offers a direct approach for computing the ideal semantics.

Dix showed that WFS$^+$ is *a well-behaved semantics*. This means WFS$^+$ satisfies cut, closure, weak model-property, isomorphy, M_P-extension, transformation, relevance, PPE and modularity (see [15] for the formal definition of these properties). All these properties have been motivated in order to identify non-monotonic reasoning inferences with a flavor of common-sense reasoning. Dix also showed that WFS$^+$ coincides with the Well-Founded-By-Cases Semantics [36]. According to Schlipf Well-Founded-By-Cases Semantics is a reasonable logic programming semantics for pursuiting common-sense reasoning. Indeed, he showed that Well-Founded-By-Cases Semantics obeys some goals for common-sense semantics. In this setting, we point out that, with the results of this paper, we are converging three different inferences of common-sense reasoning: the ideal semantics, WFS$^+$ and the Well-Founded-By-Cases Semantics.

6.2 Labellings Argumentation

Recently, Caminada *et al.* has also explored using a 3-valued labeling approach in order to regard argumentation as logic programming and vice versa [10, 38]. In this approach, a labeling is a function Lab from a set of arguments to $\{in, out, undec\}$ in which the labels *in*,*out*,*undec* denote states of an argument, mainly *accepted*, *defeated* and *undecidable*, respectively.

In order to explore argumentation as logic programming, Caminada *et al.* have explored 3-valued logic programming semantics and the mapping P_{AF}. In this setting, they have defined *a strict relationship* between 3-valued labellings and 3-valued models [10, 38]. More accurately, let $AF = \langle AR, Attacks\rangle$ be an argumentation framework and $I = \langle T, F\rangle$ be a 3-valued interpretation of P_{AF}. For all $a \in AR$, the following conditions hold:

1. $Lab(a) = in$ iff $a \in T$.
2. $Lab(a) = out$ iff $a \in F$.

3. $Lab(a) = undec$ iff $a \in \mathcal{L}_{P_{AF}} \setminus \{T \cup F\}$.

Given that there are different labellings which characterize different argumentation semantics [9, 38], the main idea is to identify the proper 3-valued semantics which infer the proper 3-valued interpretations in order to characterize argumentations semantics. For instance, Caminada *et al.* showed that the complete labelling (which characterizes the complete semantics) can be directly inferred by the 3-stable semantics and P_{AF} [38].

A good question to ask is: can we infer the 3-valued labellings of the Dung's semantics from the logical models of Π_{AF}^{acc}? To answer this question, let us introduce the function labelling $MOD2LAB^{acc}$ as follows: Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework and $M \subseteq \mathcal{L}_{\Pi_{AF}^{acc}}$:

$$IN(M) = \{x | x \in AR \wedge acc(x) \in M\}$$
$$OUT(M) = \{x | x, y \in AR \wedge (y, x) \in Attacks \wedge x \in IN(M)\}$$
$$UNDEC(M) = AR \setminus \{IN(M) \cup OUT(M)\}$$

Let $M \subseteq \mathcal{L}_{\Pi_{AF}^{acc}}$; hence, $MOD2LAB^{acc}(M) = \langle IN(M), OUT(M), UNDEC(M) \rangle$. Let us observe that like the labelling function Ext2Lab which takes as input a set of arguments and returns a labelling [38], $MOD2LAB^{acc}(M)$ takes M as input and returns a labelling. In this setting, M is regarded as a set of arguments. Given the results around $Ext2Lab$ [9, 38] and the properties of the models of Π_{AF}^{acc} (Theorem 3 and Theorem 11), we can observe the following corollary:

Corollary 2 *Let $AF = \langle AR, Attacks \rangle$ be an argumentation framework.*

- *$\langle T, F \rangle$ is the well-founded model of Π_{AF}^{acc} then $MOD2LAB^{acc}(T)$ is the grounded labelling of AF. This result follows by Theorem 2.12 from [9] and Theorem 3.*
- *M is a stable model of Π_{AF}^{acc} then $MOD2LAB^{acc}(M)$ is a stable labelling of AF. This result follows by Theorem 2.12 from [9] and Theorem 3.*
- *M is a p-stable model of Π_{AF}^{acc} then $MOD2LAB^{acc}(M)$ is a preferred labelling of AF. This result follows by Theorem 2.12 from [9] and Theorem 3.*
- *M is a Clark's completion of Π_{AF}^{acc} then $MOD2LAB^{acc}(M)$ is a complete labelling of AF. This result follows by Theorem 8 from [38] and Theorem 3.*
- *$\langle T, F \rangle$ is the WFS$^+$ model of Π_{AF}^{acc} then $MOD2LAB^{acc}(T)$ is the ideal labelling of AF. This result follows by Theorem 3.7 from [9] and Theorem 11.*

This corollary argues that we can use logic-based tools for computing the five labelligs related to Dung's semantics. Unlike Caminada *et al.*'s approach [10, 38] which requires computing 3-valued models, our approach only requires 2-valued models. In this setting, let us highlight that, currently, SAT-problem, which is mainly oriented to compute 2-valued logical models, is the prototypical and best-researched NP-complete problem.

6.3 Argumentation Frameworks as Equations

Another branch of the argumentation research in which argumentation frameworks have been transformed into other theories is in regarding argumentation frameworks as *equations*. Given that WFS^+ can be characterized in terms of *rewriting systems*, we argue that our suggested approach introduces a simple *equational* approach for computing the maximal ideal set of an argumentation framework. In this setting, it is important to point out the *numerical equational* approach introduced by Gabbay for computing argumentation semantics [24]. From Gabbay's approach, we can highlight the following observations:

- In both Gabbay's approach and our approach, the argumentation frameworks are mapped into numerical equations and logic programs, respectively. At this point, let us observe that in Gabbay's approach, the manipulation is done by arithmetic operations, and, in our approach, the manipulation is done by syntactic operations (by the so called *Basic Transformation Rules*, see Definition 4).

- In both approaches, two different mappings may not yield the same argumentation semantics. This means that, in both approaches, finding proper mappings is a central issue.

- In both Gabbay's approach and our approach, it is an important issue to identify criteria for selecting meaningful mappings (equations) for inferring argumentation semantics.

- Given that our approach is logical model oriented, we can use logic-based solvers such SAT-solvers for inferring argumentation semantics; on the other hand, Gabbay's approach requires numerical mathematical tools such as MAPLE and MATLAB for inferring argumentation semantics. This means that both approaches offer different strategies for computing argumentation semantics.

- In both approaches, if we change the mapping from argumentation frameworks into equations (logic programs), we may get something different from Dung semantics. In this setting, unlike Gabbay's approach, which considers different mappings for different argumentation semantics, in our approach we have show that we can consider only one mapping (based on the definition of admissible sets) and different logic programming semantics for inferring different argumentation semantics based on admissible sets.

- Both approaches give place to exploring pseudo-argumentation semantics. By pseudo-argumentation semantics, we mean argumentation semantics which do not have a definition in terms of Dung's style; however, these semantics can be motivated by a particular specification or a set of equations. In this setting, in our approach, we can also build pseudo-argumentation semantics by considering different logic programming semantics [30, 33].

- Both approaches can characterize semantics which are not based on admissible sets such as CF2 [25, 30].

7 Conclusions and Future Work

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 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 and regards argumentation semantics as non-monotonic reasoning inferences.

In this paper, we introduce new results about argumentation inference as logic programming with negation as failure. By considering an argumentation framework AF and a unique mapping of AF into a logic program Π_{AF}^{acc}, we show that:

- the ideal models of Π_{AF}^{acc} characterize the ideal sets of AF (Theorem 5),
- the WFS$^+$ model of Π_{AF}^{acc} characterizes the maximal ideal set of AF (Theorem 11).
- the well founded model of Π_{AF}^{acc} characterizes the grounded extension of AF (Theorem 3),
- the stable models of Π_{AF}^{acc} characterize the stable extensions of AF (Theorem 3),
- the p-stable models of Π_{AF}^{acc} characterize the prefer extension of AF (Theorem 3) and
- the supported models of Π_{AF}^{acc} characterize the complete extensions of AF (Theorem 3).

Around these results, we show that:

- The ideal sets of an argumentation framework can be characterized in terms of p-stable models (Corollary 1).
- The ideal models are closed with respect to the transformation rules: RED^+, RED^-, $Success$, $Failure$, $Loop$, SUB, $TAUT$, LC (Theorem 7).
- The minimal ideal model of Π_{AF} is characterized by WFS^+ (Theorem 8).

The characterization of argumentation semantics in terms of logic programming semantics does not only contribute to the inference of the well-accepted argumentation semantics but this approach also contributes to the study of non-monotonic reasoning properties of the argumentation semantics. For instance, we have shown that WFS$^+$ characterizes the maximal ideal set of an argumentation framework. It is known that WFS$^+$ is a well-behaved semantics [15]. From a non-monotonic reasoning point of view, since WFS$^+$ introduces a logical definition of the maximal ideal set of an argumentation framework, the maximal ideal set of an argumentation framework can be regarded as a well-behaved semantics. Besides being a well-behaved semantics, Schlipf [36] showed that WFS by cases (WFS$^+$) is a logic programming semantics which satisfies a good number of *common sense goals*. In this setting, one can argue that the

maximal ideal set of an argumentation framework follows a common sense reasoning approach. A part of our future work is to study the interpretation of the properties of a well-behaved semantics and the properties introduced by Schlipf as argumentation inference properties.

With the results of this paper, we have shown that the five argumentation semantics suggested by Dung *et al.* can be characterized uniformly by considering a unique mapping from argumentation frameworks into logic programs (Π_{AF}^{acc}). There are argumentation semantics such as *semi-stable semantics* which, like Dung's semantics, are based on admissible sets. Therefore, a good research question is: can semi-stable semantics be characterized by Π_{AF}^{acc} and a logic programming semantics? Even in the worst case that semi-stable semantics could not be characterized by Π_{AF}^{acc} and a logic programming semantics; it will be important to observe why semi-stable could not be characterized by Π_{AF}^{acc} and a logic programming semantics. Indeed, a good question could arise, which are the properties that share grounded, stable, preferred, complete and ideal semantics that allow them to be characterized by a unique logic program (Π_{AF}^{acc})?

Characterizing Dung's semantics with Π_{AF}^{acc} argues that Π_{AF}^{acc} defines a common root of Dung's semantics. In this setting, by considering this common root among Dung's semantics, one can define new argumentation semantics. In [33], the authors introduced the so called *stable-abducible argumentation semantics*. This argumentation semantics is based on Π_{AF}^{acc} and a logic programming semantics. Moreover, this semantics is an intermediate semantics between stable semantics and preferred semantics. This evidence suggests that stable-abducible argumentation semantics is similar to semi-stable semantics. Part of our future work will be to explore the similarities and differences between stable-abducible argumentation semantics and semi-stable semantics.

Let us observe that, nowadays, the number of new argumentation semantics in the context of Dung's argumentation approach [3, 2] has increased. However, many of these new argumentation semantics are only motivated by specific examples; hence, the identification of non-monotonic reasoning properties, that a particular argumentation semantics satisfies, will have relevance supporting the well-behaviour of an argumentation semantics. In [3], a set of basic principles was defined in order to evaluate argumentation semantics. We believe that the set of principles described in [3] can be enriched by the identification of the non-monotonic reasoning properties that must satisfy any argumentation semantics. Of course, this study could be explored by the characterization of argumentation semantics in terms of logic programming semantics.

Acknowledgment

We are grateful to anonymous referees for their useful comments. This research has been partially supported by VINNOVA (The Swedish Governmental Agency for Innovation Systems), the Swedish Brain Power and CONACyT (the Mexican Governmental Agency for science and technology) [CB-2008-01 No.101581].

References

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

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

[3] P. Baroni and M. Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence.*, 171(10-15):675–700, 2007.

[4] P. Baroni, M. Giacomin, and G. Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168:162–210, October 2005.

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

[6] S. Brass, J. Dix, B. Freitag, and U. Zukowski. Transformation-based bottom-up computation of the well-founded model. *TPLP*, 1(5):497–538, 2001.

[7] S. Brass, U. Zukowski, and B. Freitag. Transformation-based bottom-up computation of the well-founded model. In T. C. P. Jürgen Dix, Luís Moniz Pereira, editor, *Non-Monotonic Extensions of Logic Programming, NMELP '96*, volume 1216 of *Lecture Notes in Computer Science*, pages 171–201, 2007.

[8] M. Caminada. Semi-Stable semantics. In P. E. Dunne and T. J. Bench-Capon, editors, *Proceedings of COMMA*, volume 144, pages 121–130. IOS Press, 2006.

[9] M. Caminada. A labelling approach for ideal and stage semantics. *Argument and Computation*, 2(1):1–21, 2011.

[10] M. Caminada, S. Sá, and J. Alcântara. On the equivalence between logic programming semantics and argumentation semantics. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 12th European Conference, ECSQARU 2013*, volume 7958 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2013.

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

[12] J. L. Carballido, M. Osorio, and J. Arrazola. Equivalence for the G'$_3$-stable models semantics. *J. Applied Logic*, 8(1):82–96, 2010.

[13] N. Dershowitz and D. A. Plaisted. *Handbook of Automated Reasoning*, chapter Rewriting. Elsevier Science Publishers, 2001.

[14] J. Dix. A classification theory of semantics of normal logic programs: I. strong properties. *Fundam. Inform.*, 22(3):227–255, 1995.

[15] J. Dix. A classification theory of semantics of normal logic programs: II. weak properties. *Fundam. Inform.*, 22(3):257–288, 1995.

[16] J. Dix, M. Osorio, and C. Zepeda. A general theory of confluent rewriting systems for logic programming and its applications. *Ann. Pure Appl. Logic*, 108(1-3):153–188, 2001.

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

[18] P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10-15):642–674, 2007.

[19] P. E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10-15):701–729, 2007.

[20] P. E. Dunne. The computational complexity of ideal semantics. *Artificial Intelligence*, 173(18):1559–1591, 2009.

[21] P. E. Dunne, W. Dvorák, and S. Woltran. Parametric properties of ideal semantics. *Artificial Intelligence*, 202:1–28, 2013.

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

[23] W. Faber and S. Woltran. Manifold answer-set programs and their applications. In *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning - Essays Dedicated to Michael Gelfond on the Occasion of His 65th Birthday*, volume 6565 of *Lecture Notes in Computer Science*, pages 44–63. Springer, 2011.

[24] D. M. Gabbay. Equational approach to argumentation networks. *Argument and Computation*, 3(2-3):87–142, 2012.

[25] D. M. Gabbay. The Equational Approach to CF2 Semantics. volume 245 of *Frontiers in Artificial Intelligence and Applications*, pages 141–152. IOS Press, 2012.

[26] D. M. Gabbay and A. S. d'Avila Garcez. Logical modes of attack in argumentation networks. *Studia Logica*, 93(2-3):199–230, 2009.

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

[28] J. C. Nieves and M. Osorio. Studying ideal semantics via logic programming semantics. In *10th Mexican International Conference on Artificial Intelligence*, pages 3–8. IEEE Press, 2011.

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

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

[31] J. C. Nieves, M. Osorio, C. Zepeda, and U. Cortés. Inferring acceptable arguments with answer set programming. In *Sixth Mexican International Conference on Computer Science (ENC 2005)*, pages 198–205. IEEE Computer Science Press, September 2005.

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

[33] M. Osorio, J. C. Nieves, and J. L. Carballido. The stable abducible argumentation semantics. In *Latin American Workshop on Non-Monotonic Reasoning 2011*, volume 804 of *CEUR Workshop Proceedings*, pages 57–68, 2011.

[34] M. Osorio, J. C. Nieves, and I. Gómez-Sebastià. CF2-extensions as Answer-set Models. In *Computational Models of Argument - COMMA*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 391–402. IOS Press, 2010.

[35] M. Osorio, J. C. Nieves, and A. Santoyo. Complete Extensions as Clark's Completion Semantics. In *Mexican International Conference on Computer Science*, page IN PRESS. IEEE Computer Science Press, 2013.

[36] J. S. Schlipf. Formalizing a logic for logic programming. *Ann. Math. Artif. Intell.*, 5(2-4):279–302, 1992.

[37] T. Wakaki and K. Nitta. Computing Argumentatoin Semantics in Answer Set Progamming. In *JSAI'2008*, volume 5447 of *Lecture Notes in Computer Science*, pages 254–269, 2009.

[38] Y. Wu, M. Caminada, and D. M. Gabbay. Complete extensions in argumentation coincide with 3-valued stable models in logic programming. *Studia Logica*, 93(2-3):383–403, 2009.