Journal of Artificial Intelligence Research 30 (2007) 501-523 Submitted 06/07; published 12/07

On the Semantics of Logic Programs with Preferences

Sergio Greco GRECO@DEIS.UNICAL.IT
DEIS, Università della Calabria
via P. Bucci, 87030 Rende - Italy

Irina Trubitsyna IRINA@DEIS.UNICAL.IT
DEIS, Università della Calabria
via P. Bucci, 87030 Rende - Italy

Ester Zumpano ZUMPANO@DEIS.UNICAL.IT
DEIS, Università della Calabria
via P. Bucci, 87030 Rende - Italy

Abstract

This work is a contribution to prioritized reasoning in logic programming in the presence of preference relations involving atoms. The technique, providing a new interpretation for prioritized logic programs, is inspired by the semantics of *Prioritized Logic Programming* and enriched with the use of structural information of preference of *Answer Set Optimization Programming*. Specifically, the analysis of the logic program is carried out together with the analysis of preferences in order to determine the choice order and the sets of comparable models. The new semantics is compared with other approaches known in the literature and complexity analysis is also performed, showing that, with respect to other similar approaches previously proposed, the complexity of computing preferred stable models does not increase.

1. Introduction

The increased interest in preferences is witnessed by an extensive number of proposals and systems for preference handling (Grell, Konczak, & Schaub, 2005; Van Nieuwenborgh & Vermeir, 2003; Wakaki, Inoue, Sakama, & Nitta, 2003, 2004). The literature distinguishes *static* from *dynamic* preferences. Static preferences are fixed at the time a theory is specified, i.e. they are "external" to the logic program, whereas dynamic preferences appear within the logic program and are determined "on the fly". The most common form of preference consists in specifying preference conditions among rules (Brewka, 1996; Brewka & Eiter, 1999, 2000; Delgrande, Schaub, & Tompits, 2000a, 2000b, 2003; Gelfond & Son, 1997; Schauba & Wang, 2001; Van Nieuwenborgh & Vermeir, 2002, 2004; Wang, Zhou, & Lin, 2000; Zhang & Foo, 1997), whereas, some recent proposals admit the expression of preference relations among atoms (Brewka, Niemela, & Truszczynski, 2003; Brewka, 2004; Sakama & Inoue, 2000; Wakaki et al., 2003). More sophisticated forms of preferences also allow the specification of priorities between conjunctions (disjunctions) of literals (Brewka et al., 2003; Delgrande et al., 2000a; Sakama & Inoue, 2000) and numerical penalties for suboptimal options (Brewka, 2004).

This work is a contribution to prioritized reasoning in logic programming in the presence of preference conditions involving atoms. In particular, priorities are applied by following the natural ordering defined by dependencies, as proposed in the *Answer Set Optimiza-*

tion ($\mathcal{ASO}$) semantics (Brewka et al., 2003), and the comparison strategy, proposed in the *Preferred Stable Model* ($\mathcal{PSM}$) semantics (Sakama & Inoue, 2000), is reviewed by also introducing the concept of *comparable models*. The next example describes the intuition at the basis of the proposed approach.

Example 1 The following prioritized program $\langle \mathcal{P}_1, \Phi_1 \rangle$, inspired by a program presented by Brewka et al. (2003), describes different menus and the preferences among drinks and desserts:

$$\begin{array}{rl} \mathcal{P}_1: & \texttt{fish} \oplus \texttt{beef} \leftarrow \\ & \texttt{red} \oplus \texttt{white} \leftarrow \\ & \texttt{pie} \oplus \texttt{ice-cream} \leftarrow \\ & \leftarrow \texttt{fish},\ \texttt{white} \\ & \leftarrow \texttt{beef},\ \texttt{pie} \\ & \leftarrow \texttt{fish},\ \texttt{ice-cream} \end{array} \qquad \begin{array}{rl} \Phi_1: & \varrho_1: \ \texttt{white} > \texttt{red} \leftarrow \texttt{fish} \\ & \varrho_2: \ \texttt{red} > \texttt{white} \leftarrow \texttt{beef} \\ & \varrho_3: \ \texttt{pie} > \texttt{ice-cream} \leftarrow \texttt{red} \end{array}$$

The symbol $\oplus$ denotes exclusive disjunction, i.e. if the body of the rule is true exactly one atom in the head is true, whereas a rule with empty head defines a constraint, i.e. a rule which is satisfied only if the body is false. The first three rules of $\mathcal{P}_1$ select the main dish, the drink and the dessert; the last three rules are constraints and state that a feasible solution cannot contain i) `fish` and `white` or ii) `beef` and `pie` or iii) `fish` and `ice-cream`. Prioritized rules in Φ_1 introduce preferences among drinks (ϱ_1, ϱ_2) and desserts (ϱ_3).

The program $\mathcal{P}_1$ has three stable models: $M_1 = \{\texttt{fish}, \texttt{red}, \texttt{pie}\}$, $M_2 = \{\texttt{beef}, \texttt{white}, \texttt{ice-cream}\}$ and $M_3 = \{\texttt{beef}, \texttt{red}, \texttt{ice-cream}\}$. The $\mathcal{PSM}$ returns M_1 as the unique preferred model; whereas the $\mathcal{ASO}$ technique, following the natural ordering of preference rules (ϱ_1 and ϱ_2 precede ϱ_3), derives that M_3 is the unique solution. Thus, the two approaches provide different results. □

The structure of preference rules in the above example suggests that i) `fish` and `beef` are alternative options for the main dish and ii) the choice of drink depends on the selected main dish and precedes the choice of dessert. The second conclusion is based on the observation that ϱ_1 and ϱ_2 provide opposite valuations of the choice of drink and they define two different classes of models (menus), which should be considered separately. In other words, the model M_1 (associated with the menu containing `fish`) should not be compared with the models M_2 and M_3 (associated with the menus containing `beef`). Consequently, both M_1 and M_3 should be preferred.

Observe that in the above example the $\mathcal{PSM}$ semantics derives that M_1 is the preferred model as M_1 is preferable to M_3 (due to the presence of rule ϱ_3), M_3 is preferable to M_2 (due to the presence of rule ϱ_2) and, transitively, that M_1 is also preferable to M_2. It is worth noting that the use of the transitive closure makes the comparison of models much more complex as two models cannot be compared directly. On the other hand the $\mathcal{ASO}$ semantics is sensitive to syntactic changes of programs. This fact is illustrated by means of the following example.

Example 2 Consider the prioritized program $\langle P_2, \Phi_2 \rangle$, an extension of the prioritized program defined in Example 1:

$$
\begin{array}{ll}
P_2: & \texttt{fish} \oplus \texttt{beef} \leftarrow \\
& \texttt{red} \oplus \texttt{white} \oplus \texttt{beer} \leftarrow \\
& \texttt{pie} \oplus \texttt{ice-cream} \leftarrow \\
& \leftarrow \texttt{fish},\ \texttt{white} \\
& \leftarrow \texttt{beef},\ \texttt{pie} \\
& \leftarrow \texttt{fish},\ \texttt{ice-cream} \\
& \leftarrow \texttt{beer}
\end{array}
\qquad
\begin{array}{lll}
\Phi_2: & \widehat{\varrho}_1: & \texttt{beer} > \texttt{white} > \texttt{red} \leftarrow \texttt{fish} \\
& \widehat{\varrho}_2: & \texttt{beer} > \texttt{red} > \texttt{white} \leftarrow \texttt{beef} \\
& \widehat{\varrho}_3: & \texttt{pie} > \texttt{ice-cream} \leftarrow \texttt{red}
\end{array}
$$

This program is equivalent to the one reported in Example 1 as, even if it contains an additional choice (beer), this option is not feasible, for the presence of the constraint $\leftarrow$ beer. The set of stable models associated with the program P_2, coincides with those reported for the program $\mathcal{P}_1$ in Example 1, and consists of: $M_1 = \{\texttt{fish}, \texttt{red}, \texttt{pie}\}$, $M_2 = \{\texttt{beef}, \texttt{white}, \texttt{ice-cream}\}$ and $M_3 = \{\texttt{beef}, \texttt{red}, \texttt{ice-cream}\}$. Considering the set of preference rules, note that both preferences regarding the choice of drink have beer as the best option, but there is no stable model containing it. Intuitively, as the two problems $\langle \mathcal{P}_1, \Phi_1 \rangle$ in Example 1 and $\langle P_2, \Phi_2 \rangle$ are equivalent they must have the same preferred models. The $\mathcal{ASO}$ semantics is sensitive to the program change and gives M_1 and M_3 as a solution, whereas, for the equivalent program in Example 1, the returned preferred model is M_3 only. No change occurs in the set of preferred models for the $\mathcal{PSM}$ semantics. □

Thus, in this paper we present a new semantics, inspired by the $\mathcal{PSM}$ and $\mathcal{ASO}$ semantics, which seems to better capture the intuitive meaning of programs and avoids the above mentioned problems.

1.1 Contribution

The paper provides a new semantics for prioritized logic programs enriching the one proposed by Sakama and Inoue (2000) with additional information gained from the structure of preference rules as proposed by Brewka et al. (2003). In particular, the new semantics uses different preference relations among stable models and introduces a natural ordering among preferences that fixes the order of choices, on the basis of the *stratification* of the preference program. Each decision is determined by the set of choices belonging to the corresponding level and provides the subset of models given in input as solution. Once a decision is made, this output subset becomes the input set of the following decision and so on. The proposed semantics drives the decision making process taking into account catching additional information regarding non comparable sets of models so partitioning the set of models of the program by looking at their alternative decisions. To this end the concept of *incomparability*, not taken into account by previous approaches, is introduced.

The paper presents a detailed comparison of the approaches dealing with preference relations among atoms. Particular attention is devoted to the $\mathcal{PSM}$ and the $\mathcal{ASO}$ semantics. The analysis of the complexity of computing preferred answer sets is also performed, showing that, w.r.t. previous proposals, such as the $\mathcal{PSM}$ and the $\mathcal{ASO}$ semantics, the complexity of computing preferred stable models does not increase.

1.2 Plan Of The Paper

The rest of the paper is organized as follows: in Section 2 preliminaries on Disjunctive and Abductive Logic Programs, Prioritized Logic Programs and Answer Set Optimization semantics are given; in Section 3 a new interpretation for prioritized logic programs is presented; in Section 4 complexity results are provided; in Section 5 a comparison of the presented semantics with the $\mathcal{PSM}$ and $\mathcal{ASO}$ semantics is performed, and other approaches, known in the literature, are briefly described; finally, Section 6 outlines our conclusions.

2. Preliminaries

We assume familiarity with relational database theory, disjunctive logic programs, disjunctive deductive databases, (disjunctive) stable model semantics and computational complexity (Eiter, Gottlob, & Mannila, 1997b; Gelfond & Lifschitz, 1988, 1991; Papadimitriou, 1994).

2.1 Background

A *(disjunctive) logic program* is a finite set of rules r of the form $a_1 \vee \cdots \vee a_k \leftarrow b_1, ..., b_m$, $not\, c_1, ..., not\, c_n$ with $k + m + n > 0$, where $a_1, ..., a_k, b_1, ..., b_m$, $c_1, ..., c_n$ are atoms. The disjunction $a_1 \vee \cdots \vee a_k$, denoted by $head(r)$, is called *head* of r, while the conjunction $b_1, ..., b_m, not\, c_1, ..., not\, c_n$, denoted by $body(r)$, is called *body* of r.

The *Herbrand Universe* $U_\mathcal{P}$ of a program $\mathcal{P}$ is the set of all constants appearing in $\mathcal{P}$[1], and its *Herbrand Base* $\mathcal{B}_\mathcal{P}$ is the set of all ground atoms constructed from the predicates appearing in $\mathcal{P}$ and the constants from $U_\mathcal{P}$. A term (resp. an atom, a rule or a program) is *ground* if no variables occur in it. A rule r' is a *ground instance* of a rule r, if r' is obtained from r by replacing every variable in r with some constant in $U_\mathcal{P}$; $ground(\mathcal{P})$ denotes the set of all ground instances of the rules in $\mathcal{P}$.

The intuitive meaning of the previous disjunctive rule is that if $body(r)$ is true, i.e. $b_1, ..., b_m$ are true and $c_1, ..., c_n$ are false, then $head(r)$ must be true, i.e at least one of $a_1, ..., a_k$ has to be true (otherwise r is not satisfied). Rules with empty head, called *denials* or *constraints*, will be used to define constraints and are satisfied only if the body is false. In this paper exclusive disjunction, denoted by $\oplus$, is used in the head; the statement $head(r) = a_1 \oplus ... \oplus a_k$ is true, if exactly one of $a_1, ..., a_k$ is true, i.e. a disjunctive rule of the form $a_1 \oplus \cdots \oplus a_k \leftarrow body$ is a shorthand for the rule $a_1 \vee \cdots \vee a_k \leftarrow body$ and $(k \times (k-1)/2)$ constraints of the form $\leftarrow a_i, a_j, body$ with $1 \leq i < j \leq k$. The solution of a logic program $\mathcal{P}$ is given in terms of stable model (answer set) semantics (Gelfond & Lifschitz, 1988, 1991).

An interpretation M for $\mathcal{P}$ is a model of $\mathcal{P}$ if M satisfies all rules in $ground(\mathcal{P})$. The minimal model semantics, defined for positive $\mathcal{P}$, assigns to $\mathcal{P}$ the set of its *minimal models* $\mathcal{MM}(\mathcal{P})$, where a model M for $\mathcal{P}$ is minimal, if no proper subset of M is a model for $\mathcal{P}$ (Minker, 1982). The more general *disjunctive stable model semantics* also applies to programs with (unstratified) negation (Gelfond & Lifschitz, 1991). Disjunctive stable model semantics generalizes stable model semantics, previously defined for normal programs (Gelfond & Lifschitz, 1988). For any interpretation M, denote with $\mathcal{P}^M$ the ground positive program derived from $ground(\mathcal{P})$ by i) removing all rules that contain a negated atom

1. We are considering function free programs.

not a in the body and $a \in M$, and ii) removing all negated atoms from the remaining rules. An interpretation M is a (disjunctive) stable model of $\mathcal{P}$ if and only if $M \in \mathcal{MM}(\mathcal{P}^M)$. For general $\mathcal{P}$, the stable model semantics assigns to $\mathcal{P}$ the set $\mathcal{SM}(\mathcal{P})$ of its *stable models*. It is well known that stable models are minimal models (i.e. $\mathcal{SM}(\mathcal{P}) \subseteq \mathcal{MM}(\mathcal{P})$) and that for negation free programs, minimal and stable model semantics coincide (i.e. $\mathcal{SM}(\mathcal{P}) = \mathcal{MM}(\mathcal{P})$).

2.2 Extended And Abductive Programs

Given an atom $p(t)$, a literal is either $p(t)$ or its strong negation $\neg p(t)$. An extended program is a program where atoms are replaced by literals. The semantics of extended disjunctive programs can also be given in terms of stable models by considering p and $\neg p$ as different predicate symbols and considering the implicit constraint $\leftarrow p(X), \neg p(X)$ (Gelfond & Lifschitz, 1991).

Abductive logic programming is an extension of logic programming to perform abductive reasoning (Kakas, Kowalski, & Toni, 1992; Inoue & Sakama, 1998). An abductive program (ALP) is a pair $\langle \mathcal{P}, \mathcal{A} \rangle$, where $\mathcal{P}$ is an extended program and $\mathcal{A}$ is a set of literals called abducibles. $\langle \mathcal{P}, \mathcal{A} \rangle$ can be represented by means of an extended program $\Gamma = \mathcal{P} \cup \{g(t) \vee g'(t) \leftarrow \ \mid g(t) \in \mathcal{A}\} \cup \{g'(t) \leftarrow not\, g(t) \mid g(t) \in \mathcal{A}\}$.

Let Γ be an ALP and G a ground atom denoting an observation. Then, a set $S \subseteq \mathcal{A}$ is an explanation of G iff there is a stable model M of Γ such that $S = M \cap \mathcal{A}$ and $G \in M$; a stable model M is $\mathcal{A}$-minimal if there is no stable model N such that $N \cap \mathcal{A} \subset \mathcal{M} \cap \mathcal{A}$. Moreover, S is a minimal explanation (i.e. there is no explanation $S' \subset S$) of G iff $\Gamma \cup \{\leftarrow not\ G\}$ has a consistent $\mathcal{A}$-minimal stable model M such that $S = M \cap \mathcal{A}$ (Inoue & Sakama, 1998).

It has been shown that given an ALP $\mathcal{P}$ and a ground atom G denoting an observation, deciding whether there exists a $\mathcal{A}$-minimal explanation S for G is Σ_3^p-complete (Eiter, Gottlob, & Leone, 1997a).

The rest of this section will briefly review the two main approaches for prioritizing reasoning we refer to, i.e. Prioritized Logic Programs and Answer Set Optimization, proposed, respectively, by Sakama and Inoue (2000) and Brewka et al. (2003).

2.3 Prioritized Logic Programs

A *(partial) preference relation* $\succeq$ among atoms is defined as follows: given two atoms e_1 and e_2, the statement $e_1 \succeq e_2$ (called *priority*) means that e_1 has higher priority than e_2. Moreover, if $e_1 \succeq e_2$ and $e_2 \succeq e_3$, then $e_1 \succeq e_3$. A priority statement $e_1 \succeq e_2$ states that for each a_1 instance of e_1 and for each a_2 instance of e_2 the preference relation $a_1 \succeq a_2$ holds. A *prioritized logic program* (PLP) is a pair $\langle \mathcal{P}, \Phi \rangle$ where $\mathcal{P}$ is a disjunctive program and Φ is a set of priorities. Φ^* denotes the set of priorities which can be reflexively or transitively derived from Φ.

The statement $e_1 \succ e_2$ stands for $e_1 \succeq e_2$ and $e_2 \not\succeq e_1$. Clearly, if $e_1 \succ e_2$, the sets of ground instantiations of e_1 and e_2 have an empty intersection.

Definition 1 Given a prioritized logic program $\langle \mathcal{P}, \Phi \rangle$, the relation $\sqsupseteq$ is defined over the stable models of $\mathcal{P}$ as follows. For any stable models M_1, M_2 and M_3 of $\mathcal{P}$:

1. $M_1 \sqsupseteq M_1$,
2. $M_1 \sqsupseteq M_2$ if $\exists e_1 \in M_1 - M_2$, $\exists e_2 \in M_2 - M_1$ such that $(e_1 \succeq e_2) \in \Phi^*$ and $\nexists e_3 \in M_2 - M_1$ such that $(e_3 \succ e_1) \in \Phi^*$,
3. if $M_1 \sqsupseteq M_2$ and $M_2 \sqsupseteq M_3$, then $M_1 \sqsupseteq M_3$.

If $M_1 \sqsupseteq M_2$ then M_1 is *preferable* to M_2. Moreover, if $M_1 \sqsupseteq M_2$ and $M_2 \not\sqsupseteq M_1$ then $M_1 \sqsupset M_2$. □

An interpretation M is a *preferred* stable model of $\langle \mathcal{P}, \Phi \rangle$ if M is a stable model of $\mathcal{P}$ and $N \sqsupseteq M$ implies $M \sqsupseteq N$ for any stable model N (equivalently, if there is no interpretation N which is a stable model of $\mathcal{P}$ and $N \sqsupset M$). The set of preferred stable models of $\langle \mathcal{P}, \Phi \rangle$ will be denoted by $\mathcal{PSM}(\langle \mathcal{P}, \Phi \rangle)$. Note that the relation $\Phi_1 \subseteq \Phi_2$ between two PLPs $\langle \mathcal{P}, \Phi_1 \rangle$ and $\langle \mathcal{P}, \Phi_2 \rangle$ does not imply $\mathcal{PSM}(\langle \mathcal{P}, \Phi_2 \rangle) \subseteq \mathcal{PSM}(\langle \mathcal{P}, \Phi_1 \rangle)$.

In PLP priority relations are used to express priorities over atoms, whereas priorities over more general forms of knowledge (conjunctive, disjunctive knowledge, rules, preconditions) can be expressed by a simple rewriting of the preference program. For instance, a preference rule with precondition of the form $(e_1 \succeq e_2) \leftarrow B$ is expressed in PLP as $e'_1 \succeq e'_2$, where $e'_1 \leftarrow e_1, B$ and $e'_2 \leftarrow e_2, B$.

The semantics of prioritized programs proposed by Sakama and Inoue (2000) will be denoted by $\mathcal{PSM}$ semantics. A sound and complete procedure that allows preferred answer sets for a PLP program to be computed using a *generate and test algorithm* has been proposed by Wakaki et al. (2003). This algorithm translates a PLP program $\langle \mathcal{P}, \Phi \rangle$ and any answer set S of the program $\mathcal{P}$ into a single logic program $T[\mathcal{P}, \Phi, S]$, such that its answer sets are answer sets of $\mathcal{P}$ preferable to S. More details can be found in works presenting the semantics and the implementation (Sakama & Inoue, 2000; Wakaki et al., 2004).

The complexity of answering queries over PLP programs is at least one level above the complexity of answering queries over standard programs (without preferences). In particular, let $\langle \mathcal{P}, \Phi \rangle$ be a prioritized logic program, then i) deciding the existence of a preferred stable model is Σ_2^p-*hard*; ii) deciding whether an atom is true in some (resp. all) preferable stable model of $\langle \mathcal{P}, \Phi \rangle$ is Σ_3^p-hard (resp. Π_3^p-hard). It is worth noting that in the original work (Sakama & Inoue, 2000) it has been claimed that the complexity is exactly one level above the complexity of standard programs, but the proof does not take into account the transitivity property of the preference relation.

2.4 Answer Set Optimization

An answer set optimization program, denoted as ASO program, is a pair $\langle \mathcal{P}, \Phi \rangle$, where $\mathcal{P}$ is a disjunctive program, called *Generating Program*, and Φ is a *Preference Program* consisting of a finite set of rules of the form: $a_1 > \cdots > a_k \leftarrow b_1, ..., b_m, not\ c_1, ..., not\ c_n$ where b_is and c_js are literals and a_is are boolean combinations[2] of literals; here a literal is either an atom L or its (strong) negation $\neg L$. Φ determines a preference ordering on the answer sets described by the generating program $\mathcal{P}$.

2. A boolean combination is a formula built of atoms by means of disjunctions, conjunctions, and default negation.

Definition 2 Let $\langle \mathcal{P}, \Phi \rangle$ be an ASO program where $\Phi = \{\varrho_1, ..., \varrho_n\}$ and S be an answer set of $\mathcal{P}$, then S induces a satisfaction vector $V_S = (v_S(\varrho_1), ..., v_S(\varrho_n))$ where:

- $v_S(\varrho_j) = I$, if ϱ_j is *Irrelevant* to S, i.e. i) the body of ϱ_j is not satisfied in S or ii) the body of ϱ_j is satisfied, but none of the atoms in the head of ϱ_j is satisfied in S.

- $v_S(\varrho_j) = min\{i \mid S \models a_i \wedge \varrho_j = a_1 > \cdots > a_k \leftarrow b_1, .., b_m, not\ c_1, .., not\ c_n\}$, otherwise. □

In the comparison of models it is assumed that I is equal to 1 (i.e., $v_S(\varrho_j) = I$ is equivalent to $v_S(\varrho_j) = 1$).

Definition 3 Let S_1 and S_2 be two answer sets, then i) $V_{S_1} \leq V_{S_2}$ if $v_{S_1}(\varrho_i) \leq v_{S_2}(\varrho_i)$ for every $i \in [1..n]$; ii) $V_{S_1} < V_{S_2}$ if $V_{S_1} \leq V_{S_2}$ and for some $i \in [1..n]$ $v_{S_1}(\varrho_i) < v_{S_2}(\varrho_i)$. In these cases $S_1 \sqsupseteq S_2$ and $S_1 \sqsupset S_2$[3], respectively.

A set of literals S is an *optimal model* of an ASO program $\langle \mathcal{P}, \Phi \rangle$ if S is an answer set of $\mathcal{P}$ and there is no answer set S' of $\mathcal{P}$ such that $S' \sqsupset S$. □

The complexity of ASO programs depends on the class of generating programs. For disjunctive programs the complexity of answering queries over ASO programs is exactly one level above the complexity of answering queries over standard programs (without preferences), i.e. i) deciding the existence of a preferred stable model is Σ_2^p-*complete*; ii) deciding whether a literal is true in some (resp. all) preferable stable model of $\langle \mathcal{P}, \Phi \rangle$ is Σ_3^p-complete (resp. Π_3^p-complete).

The strategy is further extended by introducing *meta-preferences* among preference rules: a ranked ASO program is a sequence $\langle \mathcal{P}, \Phi_1, ..., \Phi_n \rangle$ consisting of a generating program $\mathcal{P}$ and a sequence of pairwise disjoint preference programs Φ_i. The rank of a rule $\varrho \in \Phi_1 \cup \cdots \cup \Phi_n$, denoted $rank(\varrho)$, is the unique integer i for which $\varrho \in \Phi_i$. Given two answer sets S_1 and S_2, $S_1 \sqsupseteq_{rank} S_2$ if for every preference rule ϱ' such that $v_{S_1}(\varrho') \leq v_{S_2}(\varrho')$ does not hold, there is a rule ϱ'' such that $rank(\varrho'') < rank(\varrho')$ and $v_{S_1}(\varrho'') < v_{S_2}(\varrho'')$.

Moreover, a procedure deriving the *natural ordering* of the preference rules is introduced. Firstly, given a preference program Φ, its dependency graph $G(\Phi)$ is defined. The atoms appearing in Φ form the vertex set of $G(\Phi)$. There is a directed edge from a vertex b to a vertex a in $G(\Phi)$ if there is a rule ϱ in Φ such that a appears in the head of r and b appears in the body of r. If the graph $G(\Phi)$ is acyclic, there is a natural ranking of its atoms which can be defined recursively as follows: $rank(a) = 0$ for every atom a that has no predecessors in $G(\Phi)$; otherwise $rank(a)$ is the maximum of the ranks of all predecessors of a in $G(\Phi)$ incremented by 1. The rank of a preference rule ϱ is then defined as the maximum rank of atoms in its head.

The standard semantics of ASO programs, where priorities are examined all together, will be denoted as $\mathcal{ASO}$ semantics. The alternative semantics, where priorities are divided into strata following the natural order, will be denoted by $\mathcal{RASO}$ (ranked $\mathcal{ASO}$) semantics.

3. In the original work (Brewka et al., 2003) the symbols $\geq$ and $>$ are used instead of $\sqsupseteq$ and $\sqsupset$.

3. Preferred Answer Sets

In this paper a syntax similar to the one proposed by Brewka et al. (2003) is used. Given two atoms a_1 and a_2, the statement $a_2 > a_1$ means that a_2 has higher priority than a_1. A *(partial) preference relation* $>$ among atoms is defined as follows.

Definition 4 A *prioritized program* Π is a pair $\langle \mathcal{P}, \Phi \rangle$ where $\mathcal{P}$ is a disjunctive program and Φ is set of *preference rules* of the form:

$$a_1 > a_2 > \cdots > a_k \leftarrow b_1, ..., b_m, not\, c_1, ..., not\, c_n \tag{1}$$

where $k > 1$ and $a_1, ..., a_k, b_1, ..., b_m, c_1, ..., c_n$ are atoms. □

In the following the head and body of a preference rule ϱ will be denoted by $head(\varrho)$ and $body(\varrho)$, respectively. A *ground* prioritized program, denoted by $ground(\langle \mathcal{P}, \Phi \rangle) = \langle ground(\mathcal{P}), ground(\Phi) \rangle$ is a prioritized program, where each rule $r \in (\mathcal{P} \cup \Phi)$ with variables is replaced with the set of its ground instances, i.e the set of rules obtained by replacing variables with constants.

Intuitively, a preference rule ϱ of the form (1) describes the *choice* among $a_1, ..., a_k$ (*choice options*) under the condition specified by the body of ϱ. The head of ϱ introduces the preference order among atoms: a_i is preferred to a_j for $1 \leq i < j \leq k$. As ϱ can be applied only if $body(\varrho)$ is true, the body of ϱ specifies the decisions which have to precede this choice. For instance, $a > c \leftarrow b$ states that if b is *true*, then a is preferred to c. A preference rule with exactly two atoms in the head will be called *binary* preference rule, whereas preference rules with empty bodies will be called *preference facts*. A prioritized program is said to be in *binary form* if all its preference rules are binary.

The following example, presenting a classical program proposed by Brewka et al. (2003), will be used as running example.

Example 3 Consider the prioritized program $\langle \mathcal{P}_3, \Phi_3 \rangle$ whose stable models define the menus of a restaurant:

$\mathcal{P}_3$:
$\texttt{fish} \oplus \texttt{beef} \leftarrow$
$\texttt{red} \oplus \texttt{white} \oplus \texttt{beer} \leftarrow$
$\texttt{pie} \oplus \texttt{ice-cream} \leftarrow$
$\leftarrow \texttt{beef},\ \texttt{pie}$
$\leftarrow \texttt{fish},\ \texttt{ice-cream}$

Φ_3:
$\varrho_1 :\ \texttt{white} > \texttt{red} > \texttt{beer} \leftarrow \texttt{fish}$
$\varrho_2 :\ \texttt{red} > \texttt{white} \leftarrow \texttt{beef}$
$\varrho_3 :\ \texttt{beer} > \texttt{white} \leftarrow \texttt{beef}$
$\varrho_4 :\ \texttt{pie} > \texttt{ice-cream} \leftarrow \texttt{beer}$

The first three rules of $\mathcal{P}_3$ select the main dish, the drink and the dessert; the constraints state that a feasible solution cannot contain both `beef` and `pie` or both `fish` and `ice-cream`; while the rules in Φ_3 introduce preferences among drinks and desserts.

The program $\mathcal{P}_3$ has six stable models:

$\texttt{M}_1 = \{\texttt{fish}, \texttt{white}, \texttt{pie}\}$ $\quad$ $\texttt{M}_4 = \{\texttt{beef}, \texttt{white}, \texttt{ice-cream}\}$
$\texttt{M}_2 = \{\texttt{fish}, \texttt{red}, \texttt{pie}\}$ $\quad$ $\texttt{M}_5 = \{\texttt{beef}, \texttt{red}, \texttt{ice-cream}\}$
$\texttt{M}_3 = \{\texttt{fish}, \texttt{beer}, \texttt{pie}\}$ $\quad$ $\texttt{M}_6 = \{\texttt{beef}, \texttt{beer}, \texttt{ice-cream}\}$

Both techniques proposed by Sakama and Inoue (2000) and Brewka et al. (2003) select the stable models M_1 and M_5 as preferred ones, but the motivation is different. Indeed, the $\mathcal{PSM}$ semantics states that $M_1 \sqsupseteq M_2 \sqsupseteq M_3 \sqsupseteq M_6 \sqsupseteq M_4$ and $M_5 \sqsupseteq M_4$, whereas the

$(\mathcal{R})\mathcal{ASO}$ semantics states that $M_i \sqsupseteq M_2 \sqsupseteq M_3$, $M_i \sqsupseteq M_6$ and $M_i \sqsupseteq M_4$ for $i \in \{1, 5\}$, i.e. in the $(\mathcal{R})\mathcal{ASO}$ semantics the models M_1 and M_6 are compared directly, whereas in the $\mathcal{PSM}$ semantics the models M_1 and M_6 are compared transitively. □

Before presenting the formal semantics of programs, some preliminary definitions are needed. A preference rule of the form $a_1 > a_2 > \cdots > a_k \leftarrow body$ is shorthand for the $k-1$ binary rules of the form $a_i > a_{i+1} \leftarrow body$, with $i \in [1..k-1]$ and the set of preferences established by Φ is given by its transitive closure Φ^* defined as follows:

Definition 5 Given a prioritized program $\langle \mathcal{P}, \Phi \rangle$, the (ground) transitive closure of Φ is $\Phi^* = \Phi' \cup \{a > c \leftarrow body_1, body_2 \mid a > b \leftarrow body_1 \in \Phi^* \wedge b > c \leftarrow body_2 \in \Phi^* \wedge a \neq c\}$, where Φ' is the set of binary preference rules derived from $ground(\Phi)$. □

Thus, Φ^* is defined as the set of rules explicitly representing the preference relations among choice options. In Section 4 we will show that any ground prioritized program $\Pi = \langle \mathcal{P}, \Phi \rangle$ can be rewritten into an 'equivalent' program $\hat{\Pi} = \langle \hat{\mathcal{P}}, \hat{\Phi} \rangle$ such that $\hat{\Phi}^*$ contains a number of rules which is polynomial in the size of Π.

The structure of prioritized programs can be examined in order to establish the precedence relation among choices made. For instance, the presence of the preference rule $a > c \leftarrow b$ suggests that the selection of b precedes the choice between a and c and thus establishes the precedence relation between $\{b\}$ and $\{a, c\}$. This idea was used by Brewka et al. (2003) for determining the natural ordering among preference rules. In more detail, the relational order among atoms appearing in Φ was captured by means of the corresponding dependency graph $G(\Phi)$; and the stratification of preference rules was established by considering their head atoms.

Unfortunately, the natural ordering among the preference rules can be established only if the corresponding dependency graph is acyclic. Thus, the presence of two rules `coffee` $>$ `tea` $\leftarrow$ `pie` and `pie` $>$ `ice-cream` $\leftarrow$ `coffee` in Φ does not admit the stratification of Φ, as they introduce two "mutually dependent" choices.

The stratification algorithm proposed in this paper overcomes this problem by introducing the concept of "collapsed graph", which maps to the same node the options of "mutually dependent" choices.

Given a (ground) prioritized program $\Pi = \langle \mathcal{P}, \Phi \rangle$, $G_{\Pi} = (V, E)$ denotes the *dependency graph* whose set of nodes consists of all atoms in $\mathcal{P} \cup \Phi$ whereas there is an arc from b to a labeled ϕ (resp. p) if there is a rule in Φ (resp. $\mathcal{P}$) containing a in the head and b in the body (resp. b in either the head or the body). As the body of (preference) facts is empty, we assume that their bodies contain the built-in atom $true$, so that every fact $a \leftarrow$ can be considered as a rule $a \leftarrow true$. G_{Π}^{C} denotes the (acyclic) *collapsed dependency graph* derived from G_{Π} by replacing maximal sets of mutual dependent nodes (i.e. nodes belonging to the same cycle) with a unique node. Clearly, each node in G_{Π}^{C} is associated with a set of nodes in G_{Π}.

To each node in G_{Π}^{C} it is possible to assign a level as follows:

- For each node A in G_{Π}^{C} with input degree zero, $level(A) = 0$;

- For each node A in G_Π^C with input degree greater than zero, $level(A) = max\{max\{level(B)+1 | \exists(B, Z, \phi) \text{ in } G_\Pi^C\}, max\{level(B) | \exists(B, Z, p) \text{ in } G_\Pi^C\}\}$.

Observe that the function *level* assigns to each node the maximum distance from some node with input degree zero. The following definition introduces the concept of stratification for preference rules.

Definition 6 *Stratification.* Given a (ground) prioritized program $\Pi = \langle \mathcal{P}, \Phi \rangle$, Φ^* can be partitioned into $\langle \Phi^*[0], \Phi^*[1], ..., \Phi^*[n] \rangle$ subprograms (called strata) such that

- For each atom a in Π, $level(a) = level(A)$, where A is the node of G_Π^C associated with a;
- For each rule ϱ in Φ^*, $level(\varrho) = max\{\ level(a) \mid a \in Body(\varrho)\ \}$;
- $\Phi^*[i] = \{\ \varrho \mid \varrho \in \Phi^* \wedge level(\varrho) = i\ \}$ consists of all ground preference rules associated with the level i. □

The partition of Φ^* into $\langle \Phi^*[0], \Phi^*[1], ..., \Phi^*[n] \rangle$ is called stratification.

The above definition of stratification of preference rules establishes the order in which preferences are applied by considering both $\mathcal{P}$ and Φ. Moreover, the assignment of the level to each rule differs from the one proposed by Brewka et al. (2003) in two main aspects: the level of atoms is defined by analyzing the collapsed dependency graph and the level of rules is established by considering body atoms instead of head atoms. A more detailed comparison of the two approaches will be presented in Section 5.

Example 4 Consider the prioritized program $\langle \mathcal{P}_3, \Phi_3 \rangle$ of Example 3. The transitive closure Φ_3^* consists of the binary preference rules $\varrho_{1,1}$: `white` $>$ `red` $\leftarrow$ `fish`, $\varrho_{1,2}$: `red` $>$ `beer` $\leftarrow$ `fish` and $\varrho_{1,3}$: `white` $>$ `beer` $\leftarrow$ `fish`, derived from ϱ_1, and the rules ϱ_2, ϱ_3 and ϱ_4. Φ_3^* can be stratified into the two strata $\Phi_3^*[1] = \{\varrho_{1,1}, \varrho_{1,2}, \varrho_{1,3}, \varrho_2, \varrho_3\}$ and $\Phi_3^*[2] = \{\varrho_4\}$. □

The structural analysis performed in our approach goes beyond the stratification process and tries to understand the comparability of models. To this end the concepts of *conflicting preferences* and *comparable models* are introduced.

Two ground (binary) preferences of the form $a > b \leftarrow body_1$ and $b > a \leftarrow body_2$ are said to be *conflicting*. For instance, the preferences ϱ_1 : `white` $>$ `red` $\leftarrow$ `fish` and ϱ_2 : `red` $>$ `white` $\leftarrow$ `beef` of Example 1 are conflicting, whereas the preferences ϱ_1 and ϱ_2' : `red` $>$ `water` $\leftarrow$ `beef` are not. A set of preferences Φ is said to be conflicting if Φ^* contains two conflicting preference rules.

The intuition on the basis of our approach is clarified in this example. Suppose there are two conflicting preferences $\varrho_1 : a > b \leftarrow c$ and $\varrho_2 : b > a \leftarrow d$. The two conflicting preferences ϱ_1 and ϱ_2 specify the preference between a and b in two different sets of models, having a or b, characterized by the presence of c and d, respectively. Thus, c and d, (and atoms on which c and d depend) define the alternative decisions. Once a decision has been made, the associated solutions (models) are no longer comparable.

A preference rule ϱ is said to be *relevant* for a stable model M, if ϱ can be used to compare M with other stable models, that is if $M \models body(\rho)$ and some of its head atoms belongs to M. Given a prioritized program $\langle \mathcal{P}, \Phi \rangle$ and a (ground) preference rule $\varrho : a > b \leftarrow \mathit{body} \in \Phi^*$, the set of stable models, for which ϱ is *relevant*, is $\mathcal{SM}(\mathcal{P}, \varrho) = \{M \mid M \in \mathcal{SM}(\mathcal{P}) \wedge M \models \mathit{body} \wedge (a \vee b)\}$.

Definition 7 *Comparable models.* Let $\langle \mathcal{P}, \Phi \rangle$ be a prioritized program, M_1 and M_2 two stable models for $\mathcal{P}$ and $\langle \Phi^*[0], ..., \Phi^*[n] \rangle$ be a stratification of Φ^*, then

1. M_1 and M_2 are comparable on $\Phi^*[0]$;
2. M_1 and M_2 are comparable on $\Phi^*[i+1]$ for $i \in [1..n]$, if
   (a) they are comparable on $\Phi^*[i]$, and
   (b) there do not exist two conflicting preference rules $\varrho_1, \varrho_2 \in \Phi^*[i]$ such that $M_1 \in \mathcal{SM}(\mathcal{P}, \rho_1) - \mathcal{SM}(\mathcal{P}, \rho_2)$ and $M_2 \in \mathcal{SM}(\mathcal{P}, \rho_2) - \mathcal{SM}(\mathcal{P}, \rho_1)$. □

Observe that, the second condition in the previous definition of comparable models states that the presence of two conflicting preference rules in a given level i identifies two sets of models, for which only one of the two conflicting rules is relevant. Two models, appearing in the different sets have to be considered separately in the next levels. In other words, two stable models M_1 and M_2, having as relevant preferences the conflicting rules $\varrho_1 : a > b \leftarrow \mathit{body}_1$ and $\varrho_2 : b > a \leftarrow \mathit{body}_2$ in a given level i, are not comparable at levels greater than i, if $M_1 \in \mathcal{SM}(\mathcal{P}, \rho_1) - \mathcal{SM}(\mathcal{P}, \rho_2)$ and $M_2 \in \mathcal{SM}(\mathcal{P}, \rho_2) - \mathcal{SM}(\mathcal{P}, \rho_1)$, that is $M_1 \models \mathit{body}_1 \wedge (a \vee b) \wedge \mathit{not}\ \mathit{body}_2$ and $M_2 \models \mathit{body}_2 \wedge (a \vee b) \wedge \mathit{not}\ \mathit{body}_1$.

Example 5 Consider the stable models $M_3 = \{\mathtt{fish}, \mathtt{beer}, \mathtt{pie}\}$, $M_6 = \{\mathtt{beef}, \mathtt{beer}, \mathtt{ice\text{-}cream}\}$, and the set of preference rules Φ_3^* of Example 4. The stable models M_3 and M_6 are comparable on $\Phi_3^*[0]$ by definition, while they are not comparable on $\Phi_3^*[1]$, because $\varrho_{1,3}$ is relevant only for M_3 (as $M_3 \models \mathtt{fish} \wedge (\mathtt{white} \vee \mathtt{beer}) \wedge \mathtt{not\,beef}$), ϱ_3 is relevant only for M_6 (as $M_6 \models \mathtt{beef} \wedge (\mathtt{white} \vee \mathtt{beer}) \wedge \mathtt{not\,fish}$), and these conflicting preferences belong to $\Phi_3^*[0]$. □

Fact 1 *Let $\langle P, \Phi \rangle$ be a prioritized program without conflicting preferences and $\langle \Phi^*[0], \Phi^*[1], ..., \Phi^*[n] \rangle$ be the stratification of Φ^*. Then, any two models M_1 and M_2 are comparable on $\Phi^*[i]$, $i \in [0..n]$.* □

Proof. The proof of the above fact follows directly from Definition 7. □

On the basis of Definition 7 the declarative semantics of prioritized logic programs can now be provided. This new semantics, denoted with $\mathcal{PAS}$ (Preferred Answer Sets), is given by preferred stable models as follows:

Definition 8 *Preference between Answer Sets.* Given a prioritized program $\langle \mathcal{P}, \Phi \rangle$, the relation $\sqsupseteq$ is defined over the stable models of $\mathcal{P}$ as follows. For any pair of stable models M_1 and M_2 of $\mathcal{P}$, being $\langle \Phi^*[0], \Phi^*[1], ..., \Phi^*[n] \rangle$ be the stratification of Φ^*, $M_1 \sqsupseteq M_2$ if

1. $\exists \varrho_1 : (e_1 > e_2) \leftarrow \mathit{body}_1 \in \Phi^*[i]$ such that $e_1 \in M_1 - M_2$, $e_2 \in M_2 - M_1$, such that M_1 and M_2 are comparable on $\Phi^*[i]$, and ϱ_1 is relevant for M_1 and M_2, and

2. $\nexists \varrho_2 : (e_3 > e_4) \leftarrow body_2 \in \Phi^*[j]$, such that $j < i$, $e_3 \in M_2 - M_1$, $e_4 \in M_1 - M_2$ and ϱ_2 is relevant for M_1 and M_2.

Moreover, M_1 is strictly preferable to M_2 ($M_1 \sqsupset M_2$) if $M_1 \sqsupseteq M_2$ and $M_2 \not\sqsupseteq M_1$. □

Note also that the relation $\sqsupset$ could be defined directly by replacing the condition $j < i$ with $j \leq i$ in Item 2 of the above definition.

Definition 9 *Preferred Answer Sets.* An interpretation M is a *preferred* stable model for a prioritized program $\langle \mathcal{P}, \Phi \rangle$ if M is a stable model of $\mathcal{P}$ and there does not exist a stable model N such that $N \sqsupset M$. The set of preferred stable models for $\langle \mathcal{P}, \Phi \rangle$ will be denoted by $\mathcal{PAS}(\langle \mathcal{P}, \Phi \rangle)$. □

Note that Definition 8 introduces preferences between pairs of models by also considering additional information gained from the structure of preference rules.

Example 6 Consider the prioritized program $\langle P_3, \Phi_3 \rangle$ of Example 3 and the stratification $\langle \Phi_3^*[0], \Phi_3^*[1] \rangle$ presented in Example 4. We have that

- all models are comparable on $\Phi^*[0]$ by definition and
  - owing to $\varrho_{1,1}, \varrho_{1,2}, \varrho_{1,3}$, $M_1 \sqsupset M_2 \sqsupset M_3$;
  - owing to ϱ_2, ϱ_3, $M_5 \sqsupset M_4$ and $M_6 \sqsupset M_4$;
- as $\varrho_{1,3}$ and ϱ_3 are conflicting, models M_1 and M_3, for which $\varrho_{1,3}$ is relevant, cannot be compared in $\Phi^*[1]$ with models M_4 and M_6, for which ϱ_3 is relevant.

Therefore, as discussed in Example 5, M_3 and M_6 are not comparable on $\Phi_3^*[1]$ and, consequently, the preferred models are: M_1, M_5 and M_6. □

In the previous example the stable model M_6 is considered as good as M_5 because both have `beef` as main dish, the best choice of drink (`red` wine and `beer`, respectively) and the same dessert (`ice-cream`). Observe that both $\mathcal{ASO}$ and $\mathcal{PSM}$ semantics discard M_6. As already stated, the $\mathcal{ASO}$ semantics deduces that M_1 and M_5 are preferable to M_6 owing to ϱ_3, while the $\mathcal{PSM}$ semantics states that M_1 is preferable to M_3 and M_3 is preferable to M_6, owing to ϱ_1, ϱ_3.

Example 7 Let $\langle \mathcal{P}_3, \Phi_3 \rangle$ be the program of Example 3, consider the program $\langle \mathcal{P}_3, \widehat{\Phi}_3 \rangle$ where $\widehat{\Phi}_3$ is derived from Φ_3 by replacing ϱ_4 with

$$\varrho_4' : \texttt{pie} > \texttt{ice-cream} \leftarrow$$

The new ground preference program has the unique level $\widehat{\Phi}_3^*[0] = \{\varrho_{1,1}, \varrho_{1,2}, \varrho_{1,3}, \varrho_2, \varrho_3, \varrho_4'\}$, where $\varrho_{1,1}, \varrho_{1,2}$ and $\varrho_{1,3}$ are derived from ϱ_1, as shown in Example 4. Due to ϱ_4', the following relations also hold: $M_i \sqsupset M_j$ for $i \in \{1, 2, 3\}$ and $j \in \{4, 5, 6\}$. Therefore, M_1 is the unique preferred model. The same result is obtained by both $\mathcal{PSM}$ and $\mathcal{ASO}$ semantics. □

4. Complexity

This section provides some results concerning the computational complexity of computing preferred stable models and answering queries under $\mathcal{PAS}$ semantics. We consider here data complexity where the input domain $U_{\mathcal{P}}$ consists of the Herbrand universe (we assume that all constants occurring in Φ also occur in $\mathcal{P}$). Clearly, the size of the Herbrand base $\mathcal{B}_{\mathcal{P}}$ as well as the sizes of $ground(\mathcal{P})$ and $ground(\Phi)$ are polynomial in the size of $U_{\mathcal{P}}$.

The following results demonstrate that allowing preferences among atoms in the semantics proposed here increases the complexity and expressivity of the language by one level in the polynomial hierarchy. Thus the use of additional information does not increase the computational complexity of the proposed approach with respect to the above mentioned techniques (Brewka et al., 2003; Sakama & Inoue, 2000).

Proposition 1 *Let $\Pi = \langle \mathcal{P}, \Phi \rangle$ be a prioritized program, then there exists a program $\hat{\Pi} = \langle \hat{\mathcal{P}}, \hat{\Phi} \rangle$ equivalent to $\langle ground(\mathcal{P}), \Phi^* \rangle$ such that i) the stratification of $\hat{\Phi}$ can be computed in polynomial time, and ii) $\langle \hat{\mathcal{P}}, \hat{\Phi} \rangle$ can be derived from $\langle ground(\mathcal{P}), \Phi^* \rangle$ in polynomial time.*

Proof. We start by considering the program $\Pi' = \langle ground(\mathcal{P}), \Phi' \rangle$, where Φ' is the binary ground version of Φ. The size of the graphs $G_{\Pi'}$ and $G^C_{\Pi'}$ is polynomial in the size of Π' and can be computed in polynomial time. As the assignment of levels to nodes in $G^C_{\Pi'}$ can be done in polynomial time, the assignment of levels to atoms and rules in Φ' can also be done in polynomial time.

Let $\langle \Phi'[0], ..., \Phi'[n] \rangle$ be the stratification of Φ'. We generate a new ground prioritized program $\langle \hat{\mathcal{P}}, \hat{\Phi} \rangle$ which is equivalent to $\langle ground(\mathcal{P}), \Phi^* \rangle$ and such that the size of $\hat{\Phi}$ is polynomial in the size of Φ'.

Initially, assign to each ground atom a_i appearing in the head of a preference rule in Φ' a unique index i. Let $a_1, ..., a_p$ be the (indexed) atoms appearing in the head of rules in Φ', $\hat{\mathcal{P}}$ denotes the program $ground(\mathcal{P}) \cup \mathcal{P}_{\Phi}$ where

$$\begin{aligned} \mathcal{P}_{\Phi} = \ & \{ b(i,j,l) \leftarrow body_{i,j} \mid a_i > a_j \leftarrow body_{i,j} \in \Phi'[l] \ \} \cup \\ & \{ b(i,j,l) \leftarrow b(i,k,l_1), b(k,j,l_2), l = max(l_1,l_2) \mid i,j,k \in [1..p], l_1, l_2 \in [1..n] \} \end{aligned}$$

and b is a new predicate symbol. Then, $\hat{\Phi}$ denotes the new set of ground preference rules defined as follows:

$$\hat{\Phi} = \{ a_i > a_j \leftarrow b(i,j,l) \mid i,j \in [1..p] \wedge l \in [0..n] \ \}$$

The stratification of $\hat{\Phi}$ is obtained by associating to each stratum l the preference rules whose body atom has the value of the level attribute equal to l, that is $\hat{\Phi}[l] = \{ a_i > a_j \leftarrow b(i,j,l) \mid a_i > a_j \leftarrow b(i,j,l) \in \hat{\Phi} \}$.

In order to show the equivalence between $\langle ground(\mathcal{P}), \Phi^* \rangle$ and $\langle \hat{\mathcal{P}}, \hat{\Phi} \rangle$, observe that the set of stable models of $ground(\mathcal{P})$ and $\hat{\mathcal{P}}$ are 'equivalent', i.e. for each $M \in \mathcal{SM}(\hat{\mathcal{P}})$ there is a stable model $N \in \mathcal{SM}(ground(\mathcal{P}))$ such that $N = M - \{ b(i,j,l) \mid b(i,j,l) \in M \}$ and for each $N \in \mathcal{SM}(ground(\mathcal{P}))$ there is a stable model $M \in \mathcal{SM}(\hat{\mathcal{P}})$ such that $N = M - \{ b(i,j,l) \mid b(i,j,l) \in M \}$, as the rules in $ground(\mathcal{P})$ do not contain atoms of the form $b(i,j,l)$ in their bodies.

Moreover, let N be a stable model of $ground(\mathcal{P})$ and M be the corresponding stable model of $\hat{\mathcal{P}}$ ($N \subseteq M$), for each ground preference rule $a_i > a_j \leftarrow body_{i,j}$ in $\Phi^*[l]$ whose body is true in N, there is a ground rule $a_i > a_j \leftarrow b(i,j,l)$ in $\hat{\Phi}[l]$ whose body is also true in M and vice versa. Therefore the two sets $\hat{\Phi}[l]$ and $\Phi^*[l]$ are equivalent, for all $l \in [1..n]$.

Clearly, the program $\langle \hat{\mathcal{P}}, \hat{\Phi} \rangle$ is derived from $\langle ground(\mathcal{P}), \Phi^* \rangle$ in polynomial time. $\Box$

In the following, for the sake of simplicity of presentation, we continue to refer to the program $\langle \mathcal{P}, \Phi \rangle$ and to the stratification of Φ^*.

Proposition 2 *Let $\langle \mathcal{P}, \Phi \rangle$ be a prioritized program, M_1 and M_2 two stable models for $\mathcal{P}$, and $\langle \Phi^*[0], \Phi^*[1], ..., \Phi^*[n] \rangle$ a stratification of Φ^*. The problem of checking whether, for a given $k \leq n$, M_1 and M_2 are comparable on $\Phi^*[0], ..., \Phi^*[k]$ can be solved in polynomial time.*

Proof. Obviously M_1 and M_2 are comparable in $\Phi^*[0]$. Assuming that M_1 and M_2 are comparable for a given level $j < k$, M_1 and M_2 are comparable for the level $j+1$ if there are no two conflicting preference rules $\rho_1 = a > b \leftarrow body_1$ and $\rho_2 = b > a \leftarrow body_2$ in $\Phi^*[j]$ such that $M_1 \models body_1 \wedge (a \vee b) \wedge not\ body_2$ and $M_2 \models body_2 \wedge (a \vee b) \wedge not\ body_1$. This check can be done in polynomial time as the number of rules in $\Phi^*[j]$ is polynomial in the size of $U_{\mathcal{P}}$. Moreover, as the maximum value of k is bounded by the size of Φ^* (which is bounded by the set of atoms in $\mathcal{B}_{\mathcal{P}}$), the global complexity is also polynomial. $\Box$

Corollary 1 *Let $\langle \mathcal{P}, \Phi \rangle$ be a prioritized program, M_1 and M_2 two interpretations for $\mathcal{P}$. The problem of checking whether $M_1 \sqsupset M_2$ can be solved in polynomial time.*

Proof. Straightforward from Definition 8 and Proposition 2. $\Box$

Lemma 1 *Let $\langle \mathcal{P}, \Phi \rangle$ be a prioritized program and M an interpretation for $\mathcal{P}$. The problem of deciding whether M is a preferred stable model for $\langle \mathcal{P}, \Phi \rangle$ is in Π_2^p.*

Proof. Consider the complementary problem of deciding whether M is not a preferred stable model for $\langle \mathcal{P}, \Phi \rangle$. In such a case it is sufficient to first check if M is a stable model. If M is a stable model it is sufficient to guess an interpretation N and to check that i) N is a stable model for $\mathcal{P}$ and ii) $N \sqsupset M$. The check on part i) (as well as the check that M is a stable model) can be done by means of a $\mathcal{NP}$ oracle as the problem of deciding whether an interpretation is a stable model for a disjunctive program is $co\mathcal{NP}$-complete, whereas the check on part ii) can be done in polynomial time (see Corollary 1). Therefore, the complexity of the complementary problem is $\mathcal{NP}^{\mathcal{NP}}$ and, consequently, the complexity of the original problem is $co\mathcal{NP}^{\mathcal{NP}}$. $\Box$

Theorem 1 *Let $\langle \mathcal{P}, \Phi \rangle$ be a prioritized program. Then*

1. *Deciding whether a ground atom G is true in some preferred stable models of $\langle \mathcal{P}, \Phi \rangle$ is Σ_3^p-complete;*

2. *Deciding whether a ground atom G is true in all preferred stable models of $\langle \mathcal{P}, \Phi \rangle$ is Π_3^p-complete.*

Proof. *Membership:* We first demonstrate that deciding whether G is true in some preferred stable model of $\langle \mathcal{P}, \Phi \rangle$ is in Σ_3^p. This result suffices to prove that the complementary problem, consisting in deciding whether A is true in all preferred stable models, is Π_3^p-complete.

To show the membership it is sufficient to guess an interpretation M containing G and to check whether M is a preferred stable model. From Lemma 1 the problem of deciding whether M is a preferred stable model is in Π_2^p and can be solved by means of a Σ_2^p oracle. Therefore, deciding whether there exists a preferred stable model containing G is in $\mathcal{NP}^{\Sigma_2^p} = \Sigma_3^p$.

Hardness: Given an abductive logic program consisting of a disjunctive program $\mathcal{P}$ and a set of abducibles (positive) atoms $\mathcal{A}$, the ground abductive logic program derived prom $\mathcal{P}$ and $\mathcal{A}$ is

$$\begin{array}{rlll} \Gamma = ground(\mathcal{P}) & \cup & \{\ g'(t) \leftarrow not\ g(t) & \mid g(t) \in ground(\mathcal{A})\ \} \\ & \cup & \{\ g'(t) \oplus g(t) \leftarrow & \mid g(t) \in ground(\mathcal{A})\ \} \end{array}$$

Let

$$\Phi = \{\ g'(t) > g(t) \leftarrow\ \mid g(t) \in ground(\mathcal{A})\ \}$$

$\langle \Gamma, \Phi \rangle$ denotes the prioritized program derived from $\mathcal{P}$ and $\mathcal{A}$.

For any two stable models $M, N \in \mathcal{SM}(\Gamma)$, $M \sqsupseteq N$ with respect to Φ means that a preference $p'(u) > p(u) \leftarrow\ \in \Phi$ such that $p'(u) \in M$ and $p(u) \in N$ exists and no preference $q'(v) > q(v) \leftarrow\ \in \Phi$ such that $q'(v) \in N$ and $q(v) \in M$ exists. This implies that $M \cap ground(\mathcal{A}) \subseteq N \cap ground(\mathcal{A})$ and, consequently, that preferred stable models are $\mathcal{A}$-minimal.

Therefore, the problem of deciding whether $\langle \Gamma, \Phi \rangle$ has an $\mathcal{A}$-minimal explanation for a goal G is equivalent to deciding whether $\langle \Gamma, \Phi \rangle$ has a preferred stable model containing G. Consequently, as the problem of deciding whether a $\mathcal{A}$-minimal explanation S for G exists is Σ_3^p-complete, the problem of deciding whether a preferred stable model M for $\langle \Gamma, \Phi \rangle$ containing G exits is also Σ_3^p-hard, whereas the problem of deciding whether all preferred stable models of $\langle \mathcal{P}, \Phi \rangle$ contain G is Π_3^p-hard. $\Box$

Corollary 2 *Let $\langle \mathcal{P}, \Phi \rangle$ be a disjunction-free, prioritized program. Then deciding whether a ground atom is true in some (all) preferred stable models of $\langle \mathcal{P}, \Phi \rangle$ is Σ_2^p-complete (Π_2^p-complete).*

Proof. The complexity is one level lower as the problem of deciding whether an interpretation M is a stable model for a disjunction-free program is polynomial. $\Box$

5. Analysis And Comparison

This section compares the semantics introduced here with the $\mathcal{PSM}$ and *($\mathcal{R}$)$\mathcal{ASO}$* semantics and briefly discusses other recently proposed semantics.

The $\mathcal{PSM}$ semantics is very elegant and compares pairs of models on the basis of their common preferences and not on the basis of their degree of satisfaction. It does not consider the natural ordering between preference rules and, in some cases, as in Example 1 and 3, compares (and consequently discards) models which in the $\mathcal{PAS}$ approach are not comparable. An interesting feature of the $\mathcal{PSM}$ technique is the application of transitive property in order to derive additional preference relations among problem solutions so that new, not immediately visible, preference relations are captured. However, as the test of the transitive property cannot be performed by a direct comparison of two models, this lies in a more complex implementation.

The $(\mathcal{R})\mathcal{ASO}$ technique is a very powerful tool as it determines the preferred models by evaluating the degree of satisfaction of all preference rules. Thus, it compares two models even in the absence of common preferences; and the preference relation between the two models can be established directly. In more detail, the $\mathcal{RASO}$ technique considers the structure of preference rules by associating a degree of satisfaction to choice options and introduces a natural ordering among preferences. As in the case of $\mathcal{PSM}$ semantics, the $(\mathcal{R})\mathcal{ASO}$ semantics also compares and, consequently, discards models which are not comparable using the $\mathcal{PAS}$ technique. For instance, for the program $\langle P_1, \Phi_1 \rangle$, presented in the Introduction, $\mathcal{RASO}$ discards M_1, having the second best option of drink, even if this is the unique possible choice in the presence of `fish`.

More specifically, the preference relation $\sqsupseteq$ used in the $\mathcal{PSM}$ approach is a *preorder* relation as it is reflexive and transitive; it determines *equivalent* answer set classes and establishes the partial preference order among the above mentioned classes. Consequently the preferred answer sets are those appearing in the preferred classes. It should be noted that $\mathcal{PSM}$ semantics *requires* the use of the transitive property in order to derive, on the basis of relations obtained by the direct comparison of pairs of models, new preference relations. On the contrary, the $(\mathcal{R})\mathcal{ASO}$ semantics uses a strict preference relation which is just asymmetric and it does not require the application of the transitive property to compare solutions.

The $\mathcal{PAS}$ semantics, proposed here, compares two solutions on the basis of their common preferences by introducing the concept of comparable models and by considering a refinement of the natural order among preference choices. Thus, it can be seen as an extension of $\mathcal{PSM}$ semantics that also uses additional information derived from the structure of preference rules, but which instead of comparing models transitively, compares models by considering the transitive closure of the (ground) preference rules.

A novelty of $\mathcal{PAS}$ semantics is the consideration of the structural information of preference rules. It introduces the concept of comparable models in order to avoid comparing models which (in our opinion) should not be compared as they are associated with alternative decisions. Moreover, it proposes a refinement of the natural order among preferences in order to define the order of choices. $\mathcal{RASO}$ semantics establishes the relational order among atoms appearing in Φ by means of the corresponding dependency graph $G(\Phi)$ and cannot treat the case of "mutually dependent" choices. The stratification algorithm, proposed in this paper, overcomes this problem by considering the collapsed graph G_{Π}^{C}, which is acyclic by construction and is not sensitive to "syntactic" changes. Moreover, in $\mathcal{RASO}$ semantics the stratification of preference rules is established by considering their head atoms; whereas in $\mathcal{PAS}$ semantics levels are assigned to rules on the basis of the body atoms, following the intuition that they describe the contexts of choices. Thus, the stratification proposed here always assigns preference facts to the first level because the level of a rule is fixed by looking at the level of body atoms.

Some of the advantages of the adopted approach are clarified by the following example.

Example 8 The problem defined by means of the prioritized program $\langle P_8, \Phi_8 \rangle$ consists in selecting the colors of the trousers and the shirt, having only $\mathtt{black}$ or $\mathtt{blue}$ trousers ($\mathtt{r}_1$) and $\mathtt{white}, \mathtt{yellow}$ or $\mathtt{red}$ shirts ($\mathtt{r}_2$) available. The fashion consultant suggests that $\mathtt{blue}$ trousers are better than $\mathtt{black}$ ones (ϱ_1); a $\mathtt{white}$ shirt is better than a $\mathtt{yellow}$ shirt (ϱ_2); and in the case of $\mathtt{black}$ trousers a $\mathtt{white}$ shirt is preferred to a $\mathtt{red}$ one (ϱ_3). Moreover, $\mathtt{blue}$ trousers do not go with a $\mathtt{white}$ shirt ($\mathtt{c}_1$) and a $\mathtt{red}$ shirt does not go with $\mathtt{blue}$ trousers ($\mathtt{c}_2$).

$$
\begin{array}{llll}
P_8: & \mathtt{r}_1: & \mathtt{black} \oplus \mathtt{blue} \leftarrow & \\
 & \mathtt{r}_2: & \mathtt{white} \oplus \mathtt{yellow} \oplus \mathtt{red} \leftarrow & \\
 & \mathtt{c}_1: & \leftarrow \mathtt{blue}, \mathtt{white} & \\
 & \mathtt{c}_2: & \leftarrow \mathtt{red}, \mathtt{blue} &
\end{array}
\qquad
\begin{array}{lll}
\Phi_8: & \varrho_1: & \mathtt{blue} > \mathtt{black} \leftarrow \\
 & \varrho_2: & \mathtt{white} > \mathtt{yellow} \leftarrow \\
 & \varrho_3: & \mathtt{white} > \mathtt{red} \leftarrow \mathtt{black}
\end{array}
$$

The program P_8 has four stable models: $M_1 = \{\mathtt{black}, \mathtt{white}\}$, $M_2 = \{\mathtt{black}, \mathtt{yellow}\}$, $M_3 = \{\mathtt{blue}, \mathtt{yellow}\}$ and $M_4 = \{\mathtt{black}, \mathtt{red}\}$. In order to define the stratification of preference rules, both $\mathcal{RASO}$ and $\mathcal{PAS}$ semantics firstly assign the level to atoms: first level to $\mathtt{blue}$, $\mathtt{black}$ and $\mathtt{yellow}$ and second level to $\mathtt{white}$ and $\mathtt{red}$. In the second step the $\mathcal{RASO}$ approach, by considering the maximum level of head atoms, assigns ϱ_1 to the first level and ϱ_2 and ϱ_3 to the second level, whereas $\mathcal{PAS}$ defines the level of preferences on the basis of body atoms and assigns ϱ_1 and ϱ_2 to the first level and ϱ_3 to the second level. Note that in this case the order of ϱ_2 is relevant for determining the preferred models. In fact, $\mathcal{RASO}$ gives only M_3, while $\mathcal{PAS}$ returns M_1 and M_3 as preferred models. $\Box$

A formal comparison of the three semantics can be carried out only for the class of programs where the specific definition of stratification is not significant. Moreover, as the $\mathcal{PSM}$ semantics is defined only for prioritized programs $\langle \mathcal{P}, \Phi \rangle$ where Φ consists only of binary facts (preference rules are rewritten into preference facts), in the following the comparison is carried out by considering programs whose preference rules consist of only facts. For such a class of programs the closure Φ^* used in the $\mathcal{PSM}$ and $\mathcal{PAS}$ semantics coincide.

Given a prioritized program $\langle \mathcal{P}, \Phi \rangle$ we denote with $G_{\mathcal{SEM}} = (V, E_{\mathcal{SEM}})$ a preference graph on the stable models of $\mathcal{P}$, where $V = \mathcal{SM}(\mathcal{P})$ denotes the set of stable models of $\mathcal{P}$ and $E_{\mathcal{SEM}}$ denotes the preference relation $\sqsupset$ defined by the semantics $\mathcal{SEM} \in \{\mathcal{PSM}, \mathcal{ASO}, \mathcal{PAS}\}$. In particular, $E_{\mathcal{SEM}}$ consists of the arcs (M_i, M_j) such that $M_i \sqsupset M_j$ holds in the $\mathcal{SEM}$ semantics. Therefore, the comparison of the different semantics can be performed by analyzing the corresponding preference graphs. A stable model M_i is preferred under the $\mathcal{SEM}$ semantics if there is no arc (M_j, M_i) in $G_{\mathcal{SEM}}$.

The following example shows the relation between the $\mathcal{PSM}$ and $\mathcal{PAS}$ semantics.

Example 9 Consider the prioritized program $\langle \mathcal{P}_9, \Phi_9 \rangle$ below

$$
\begin{array}{ll}
\mathcal{P}_9: & \mathtt{fish} \oplus \mathtt{beef} \oplus \mathtt{pork} \oplus \mathtt{chicken} \leftarrow \\
 & \mathtt{white} \leftarrow \mathtt{fish} \\
 & \mathtt{red} \leftarrow \mathtt{beef} \\
 & \mathtt{beer} \leftarrow \mathtt{pork} \\
 & \mathtt{water} \leftarrow \mathtt{chicken}
\end{array}
\qquad
\begin{array}{ll}
\Phi_9: & \mathtt{fish} > \mathtt{beef} \leftarrow \\
 & \mathtt{chicken} > \mathtt{pork} \leftarrow \\
 & \mathtt{red} > \mathtt{white} \leftarrow \\
 & \mathtt{red} > \mathtt{water} \leftarrow \\
 & \mathtt{beer} > \mathtt{water} \leftarrow
\end{array}
$$

The program has four stable models: $M_1 = \{\texttt{fish}, \texttt{white}\}$, $M_2 = \{\texttt{beef}, \texttt{red}\}$, $M_3 = \{\texttt{pork}, \texttt{beer}\}$ and $M_4 = \{\texttt{chicken}, \texttt{water}\}$, whereas the direct preference relations, for the $\mathcal{PSM}$ semantics, are as follows: $M_1 \sqsupseteq M_2$, $M_2 \sqsupseteq M_1$, $M_3 \sqsupseteq M_4$, $M_4 \sqsupseteq M_3$ and $M_2 \sqsupseteq M_4$.

Consequently, we have that the graph $G_{\mathcal{PSM}}$ consists of four nodes (M_1, M_2, M_3 and M_4) and four arcs: $M_1 \sqsupset M_3$, $M_1 \sqsupset M_4$, $M_2 \sqsupset M_3$ and $M_2 \sqsupset M_4$. Therefore, the preferred models are M_1 and M_2.

Regarding the $\mathcal{PAS}$ semantics, we have that only the relation $M_2 \sqsupset M_4$ holds and, thus, there are three preferred stable models, namely M_1 and M_2 and M_3. □

Theorem 2 *For any prioritized program $\langle \mathcal{P}, \Phi \rangle$ such that Φ consists in preference facts, $\mathcal{PSM}(\langle \mathcal{P}, \Phi \rangle) \subseteq \mathcal{PAS}(\langle \mathcal{P}, \Phi \rangle)$.*

Proof. Consider the graphs $G_{\mathcal{PSM}} = (V, E_{\mathcal{PSM}})$ and $G_{\mathcal{PAS}} = (V, E_{\mathcal{PAS}})$. Both graphs are acyclic and $E_{\mathcal{PAS}} \subseteq E_{\mathcal{PSM}}$. As the two graphs are acyclic, by adding edges which do not create cycles, the number of nodes without incoming edges decreases. Therefore, the set of nodes without incoming edges in $G_{\mathcal{PAS}}$ contains all nodes without incoming edges in $G_{\mathcal{PSM}}$ and, consequently, $\mathcal{PSM}(\langle \mathcal{P}, \Phi \rangle) \subseteq \mathcal{PAS}(\langle \mathcal{P}, \Phi \rangle)$. □

We now analyze the relation between $\mathcal{ASO}$ and $\mathcal{PAS}$ semantics. First of all note that, as observed in the Introduction, the $\mathcal{ASO}$ semantics is sensitive to syntax changes. For instance, the prioritized program

$$\texttt{a} \oplus \texttt{b} \oplus \texttt{c} \leftarrow \qquad\qquad \begin{array}{l} \rho_1 = \texttt{a} > \texttt{b} > \texttt{c} \leftarrow \\ \rho_2 = \texttt{b} > \texttt{a} \leftarrow \end{array}$$

has two preferred stable models: $M_1 = \{\texttt{a}\}$ and $M_2 = \{\texttt{b}\}$. However, for the program below

$$\texttt{a} \oplus \texttt{b} \oplus \texttt{c} \leftarrow \qquad\qquad \begin{array}{l} \rho_1' = \texttt{a} > \texttt{b} \leftarrow \\ \rho_1'' = \texttt{b} > \texttt{c} \leftarrow \\ \rho_2 = \texttt{b} > \texttt{a} \leftarrow \end{array}$$

derived from the rewriting of the rule ρ_1, $M_3 = \{\texttt{c}\}$ is also a preferred model.

Thus, we consider a special class of constraints which is not sensitive to "syntactic" changes. Since every (ground) prioritized program $\Pi = \langle \mathcal{P}, \Phi \rangle$, such that Φ^* can be partitioned into n strata, with $n > 1$, can be rewritten into a program $\hat{\Pi} = \langle \hat{\mathcal{P}}, \hat{\Phi} \rangle$, such that $\hat{\Pi}$ and Π are equivalent under the $\mathcal{PAS}$ semantics (as shown in the proof of Proposition 1), and may not be equivalent under the $\mathcal{RASO}$ semantics (as all rules in $\hat{\Phi}$ belong to the unique stratum 0), we continue to consider programs $\Pi = \langle \mathcal{P}, \Phi \rangle$ where Φ^* consists of a single stratum and, in particular, of facts.

Given a prioritized program $\langle \mathcal{P}, \Phi \rangle$ such that Φ consists only of facts, we denote with

$$\Phi^+ = \{ a_1 > \cdots > a_n \leftarrow \ \mid a_i > a_{i+1} \leftarrow \in \Phi^* \text{ for } i \in [1..n\text{-}1] \wedge n \text{ is maximum } \}$$

the set of preference rules which can be obtained from the "merging" of ground preference facts.

Lemma 2 *Let $\langle \mathcal{P}, \Phi \rangle$ be a prioritized program such that Φ consists only of preference facts and $ground(\Phi) = \Phi^+$. Then, $\mathcal{ASO}(\langle \mathcal{P}, \Phi \rangle) \subseteq \mathcal{PAS}(\langle \mathcal{P}, \Phi \rangle)$.*

Proof. Consider the two graphs $G_{\mathcal{ASO}} = (V, E_{\mathcal{ASO}})$ and $G_{\mathcal{PAS}} = (V, E_{\mathcal{PAS}})$. $(M_1, M_2) \in E_{\mathcal{PAS}}$ means that $M_1 \sqsupset M_2$, i.e. that

i) there is a ground rule $\varrho_1 : e_1 > e_2 \leftarrow \; \in \Phi^*$ such that $e_1 \in M_1 - M_2$, $e_2 \in M_2 - M_1$, and

ii) there is no ground rule $\varrho_2 : e_3 > e_4 \leftarrow \; \in \Phi^*$, such that $e_3 \in M_2 - M_1$ and $e_4 \in M_1 - M_2$.

This implies that if $ground(\Phi) = \Phi^+$

i) there must be a ground rule $\sigma_1 : \cdots > e_1 > \cdots > e_2 > \cdots \leftarrow \; \in \Phi^+$ such that $e_1 \in M_1 - M_2$, $e_2 \in M_2 - M_1$, and

ii) there must not be ground rule $\sigma_2 : \cdots > e_3 > \cdots > e_4 > \cdots \leftarrow \; \in \Phi^+$, such that $e_3 \in M_2 - M_1$ and $e_4 \in M_1 - M_2$.

Therefore, $M_1 \sqsupset M_2$ also with respect to the $\mathcal{ASO}$ semantics, and the graph $G_{\mathcal{ASO}}$ contains an arc (M_1, M_2). Consequently, as $E_{\mathcal{PAS}} \subseteq E_{\mathcal{ASO}}$, $\mathcal{ASO}(\langle \mathcal{P}, \Phi^+ \rangle) \subseteq \mathcal{PAS}(\langle \mathcal{P}, \Phi^+ \rangle)$. □

To find a tight relation between the two semantics, we consider a further restriction of Φ^+ which is obtained by deleting atoms which do not appear in any model from the ground preference rules:

$$\widehat{\Phi}^+ = \{a_1 > \cdots > a_n \leftarrow \; \mid \; \begin{array}{l} a_i > a_{i+1} \leftarrow \in \Phi^* \text{ for } i \in [1..n\text{-}1] \wedge n \text{ is maximum } \wedge \\ \exists M \in \mathcal{SM}(\mathcal{P}) \text{ s.t. } a_i, a_{i+1} \in M \} \end{array}$$

Theorem 3 *Let $\langle \mathcal{P}, \Phi \rangle$ be a prioritized program such that Φ consists only of preference facts and $ground(\Phi) = \widehat{\Phi}^+$. Then, $\mathcal{ASO}(\langle \mathcal{P}, \Phi \rangle) = \mathcal{PAS}(\langle \mathcal{P}, \Phi \rangle)$.*

Proof. $\mathcal{ASO}(\langle \mathcal{P}, \Phi \rangle) \subseteq \mathcal{PAS}(\langle \mathcal{P}, \Phi \rangle)$ derives from Lemma 2, as $\widehat{\Phi}^+$ is derived from Φ^+ by deleting nodes which do not appear in any model and do not influence the relation $\sqsupset$ in the $\mathcal{ASO}$ semantics.

To show that $\mathcal{ASO}(\langle \mathcal{P}, \Phi \rangle) \supseteq \mathcal{PAS}(\langle \mathcal{P}, \Phi \rangle)$ consider the relation in the $\mathcal{ASO}$ semantics. $M_1 \sqsupset M_2$ means that

1. there is a ground rule $\sigma_1 : \cdots > e_1 > \cdots > e_2 > \cdots \leftarrow \; \in \widehat{\Phi}^+$ such that $e_1 \in M_1 - M_2$, $e_2 \in M_2 - M_1$, and

2. there is no ground rule $\sigma_2 : \cdots > e_3 > \cdots > e_4 > \cdots \leftarrow \; \in \Phi^+$, such that $e_3 \in M_2 - M_1$ and $e_4 \in M_1 - M_2$.

This implies that if $ground(\Phi) = \widehat{\Phi}^+$

1. there must be a ground rule $\varrho_1 : e_1 > e_2 \leftarrow \; \in \Phi^*$ such that $e_1 \in M_1 - M_2$, $e_2 \in M_2 - M_1$, and

2. there is no ground rule $\varrho_2 : e_3 > e_4 \leftarrow \; \in \Phi^*$, such that $e_3 \in M_2 - M_1$ and $e_4 \in M_1 - M_2$.

Therefore, $M_1 \sqsupset M_2$ also holds with respect to the $\mathcal{PAS}$ semantics. Consequently, as $E_{\mathcal{ASO}} \subseteq R_{\mathcal{PAS}}$, $\mathcal{ASO}(\langle \mathcal{P}, \Phi^+ \rangle) \supseteq \mathcal{PAS}(\langle \mathcal{P}, \Phi^+ \rangle)$. □

An extension of the $\mathcal{ASO}$ semantics has been proposed by Brewka (2004) and Brewka, Niemela, and Truszczynski (2005). In more detail, Brewka (2004) provided a preference description language, allowing to express complex preferences by combining qualitative and quantitative penalty based preferences, whereas Brewka et al. (2005) proposed a framework to specify problem solutions (outcomes) and preferences among them. The latter proposal combines ideas from answer-set programming, answer-set optimization and CP-nets (Boutilier, Brafman, Domshlak, Hoos, & Poole, 2004). The semantics that we have proposed in this paper is different from both those proposed by Brewka (2004) and Brewka et al. (2005), as in some cases it returns different results (see Examples 2 and 4).

6. Other Approaches

Besides the approaches managing preferences among atoms, some other works proposed in the literature specify preferences among rules.

Early proposals expressing preferences on rules focus on Default Logic (Brewka & Eiter, 2000; Delgrande et al., 2000b; Rintanen, 1998), whereas more recently the emphasis has been given to logic programs. In this regard, different proposals have been developed for representing and reasoning about user preferences such as ordered logic programs (Delgrande et al., 2000a; Van Nieuwenborgh & Vermeir, 2002, 2004) and preferred answer sets of extended logic programs (Brewka & Eiter, 1999). Most of the approaches propose an extension of Gelfond and Lifschitz's extended logic programming by adding preference information (Delgrande et al., 2003; Wang et al., 2000; Zhang & Foo, 1997). Other proposals attempt to extend the well founded semantics to logic programs with preferences (Brewka, 1996; Schauba & Wang, 2001), and an extension of van-Gelder's alternating fixpoint theory for logic programs with priorities has been proposed by Wang et al. (2000).

Gelfond and Son (1997) have proposed a methodology of reasoning with prioritized default in the language of logic programming under answer set semantics. This approach enables the specification of preferences among rules and allows the definition of a set of default rules which must be satisfied as well as a second set of default rules which could be ignored.

Ordered logic programs have been introduced by Delgrande et al. (2000a) as extended logic programs whose rules are subject to a strict partial order with both static and dynamic preferences. This approach is fully prescriptive as it enforces the ordering information during the construction of the answer set. The original program is transformed into a second extended logic program in which preferences are taken into account in the sense that the answer sets obtained by evaluating the transformed theory correspond to the preferred answer sets of the original theory.

Another methodology in which logic programs containing preferences on the set of rules can be translated into logic programs under stable model semantics has been proposed by Delgrande et al. (2003).

7. Conclusions

In this paper the case of preferences involving atoms in logic programming has been studied. In particular, the behavior of the technique proposed by Sakama and Inoue (2000) and Brewka et al. (2003) has been analyzed and a semantics, interpreting each preference rule as a tool for representing a choice over alternative options, has been proposed. Specifically, the proposed approach extends the $\mathcal{PSM}$ semantics by considering a refinement of the natural order among preferences and introduces the concept of comparable models. Preferences and logic programs are examined together in order to determine the choice order and the sets of models which can be compared.

The new semantics has been compared with the $\mathcal{PSM}$ and the $\mathcal{ASO}$ semantics. Complexity analysis has also been performed showing that the use of additional information, regarding the preference order and the sets of non comparable models, does not increase the complexity of computing preferred stable models. Although the semantics presented here has the same complexity as other approaches proposed in the literature, the advantage lies in the fact that it seems to better capture the intuitive meaning of prioritized programs by also considering the structural information of preference rules.

Prioritized reasoning in logic programming under the $\mathcal{PAS}$ semantics can be easily implemented on the top of deductive systems based on stable model semantics such as DeRes, DLV, Smodels (Cholewinski, Marek, & Truszczynski, 1996; Leone, Pfeifer, Faber, Calimeri, & Dell'Armi, 2002; Syrjanen & Niemela, 2001). An architecture and a system prototype implementing prioritized reasoning (with different semantics) on the top of the DLV system has been presented by Caroprese, Trubitsyna, and Zumpano (2007).

Acknowledgments

A preliminary version of the papers has been presented by Greco, Trubitsyna, and Zumpano (2006). The authors would like to thank the anonymous referees for their useful suggestions and Filippo Furfaro for his comments.

References

Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21, 135-191.

Brewka, G. (1996). Well-Founded Semantics for Extended Logic Programs with Dynamic Preferences. *Journal of Artificial Intelligence Research*, 4, 19-36.

Brewka, G., Eiter, T. (1999). Preferred Answer Sets for Extended Logic Programs. *Artificial Intelligence*, 109(1-2), 297-356.

Brewka, G., Eiter, T. (2000). Prioritizing Default Logic. *Intellectics and Computational Logic*, Kluwer, 27-45.

Brewka, G. (2002). Logic programming with ordered disjunction. *Proceedings 18th National Conference on Artificial Intelligence (AAAI/IAAI)*, 100-105.

Brewka, G., Niemela, I., Truszczynski, M. (2003). Answer Set Optimization. *Proceedings 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 867-872.

Brewka, G. (2004). Complex Preferences for Answer Set Optimization, *Proceedings 9th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 213-223.

Brewka, G., Niemela, I., Truszczynski, M. (2005). Prioritized Component Systems. *Proceedings 20th National Conference on Artificial Intelligence (AAAI)*, 596-601.

Caroprese, L., Trubitsyna, I., Zumpano, E. (2007). Implementing Prioritized Reasoning in Logic Programming. *Proceedings International Conference on Enterprice Information Systems (ICEIS)*, 94-100.

Cholewinski, P., Marek, V. W., Truszczynski, M. (1996). Default Reasoning System DeReS. *Proceedings 5th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 518-528.

Delgrande, J., P., Schaub, T., Tompits, H. (2000). Logic Programs with Compiled Preferences. *Proceedings 14th European Conference on Artificial Intelligence (ECAI)*, 464-468.

Delgrande, J., P., Schaub, T., Tompits, H. (2000). A Compilation of Brewka and Eiter's Approach to Prioritization. *Proceedings European Workshop on Logics in Artificial Intelligence (JELIA)*, 376-390.

Delgrande, J., P., Schaub, T., Tompits, H. (2003). A Framework for Compiling Preferences in Logic Programs. *Theory and Practice of Logic Programming*, 3(2), 129-187.

Eiter, T., Gottlob, G., Leone, N. (1997). Abduction from Logic Programs: Semantics and Complexity. *Theoretical Computer Science* 189(1-2), 129–177.

Eiter, T., Gottlob, G., Mannila, H. (1997). Disjunctive Datalog. *ACM Transaction On Database Systems*, 22(3), 364–418, 1997.

Gelfond, M., Lifschitz, V. (1988). The Stable Model Semantics for Logic Programming, *Proceedings International Conference on Logic Programming (ICLP)*, 1070–1080.

Gelfond, M., Lifschitz, V. (1991). Classical Negation in Logic Programs and Disjunctive Databases, *New Generation Computing*, 9, 365–385.

Gelfond, M., Son, T.C. (1997). Reasoning with prioritized defaults. *Proc. 3rd International Workshop on Logic Programming and Knowledge Representation (LPKR)*, 164-223.

Greco, S., Trubitsyna, I., Zumpano, E. (2006). On the Semantics of Logic Programs with Preferences. *Proceedings 10th European Conference on Logics in Artificial Intelligence (JELIA)*, 203-215.

Grell, S., Konczak, K., Schaub, T. (2005). nomore<: A System for Computing Preferred Answer Sets. *Proceedings 8th International. Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, 394-398.

Janhunen, T., Niemela, I., Simons, P., You, J.-H. (2000). Unfolding partiality and disjunctions in stable model semantics, *Proceedings 7th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 411-419.

Inoue, K., Sakama, S. (1998). Negation as Failure in the Head. *Journal of Logic Programming*, 35(1), 39-78.

Kakas, A. C., Kowalski, R. A., Toni, F. (1992). Abductive Logic Programming. *Journal of Logic anc Computation*, 2(6), 719-770.

Leone, N., Pfeifer, G., Faber, W., Calimeri, F., Dell'Armi, T., Eiter, T., Gottlob, G., Ianni, G., Ielpa, G., Koch, K., Perri, S., Polleres, A. (2002). The DLV System. *Proceedings 8th European Conference on Logics in Artificial Intelligence (JELIA)*, 537-540, 2002.

Minker, J. (1982). On Indefinite Data Bases and the Closed World Assumption, *Proc. 6-th Conf. on Automated Deduction*, 292-308, 1982.

Papadimitriou, C. H. (1994). *Computational Complexity*. Addison-Wesley.

Rintanen J. (1998). Complexity of Prioritized Default Logics, *Journal of Artificial Intelligence Research*, 9, 423-461.

Sakama, C., Inoue, K. (2000). Priorized logic programming and its application to commonsense reasoning. *Artificial Intelligence*, *123*, 185-222.

Schaub, T., Wang , K. (2001). A Comparative Study of Logic Programs with Preference. *Proceedings 17th International Joint Conference on Artificial Intelligence (IJCAI)*, 597-602.

Syrjanen, T., and Niemela, I. (2001). The Smodels System. *Proceedings International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, 434-438.

Van Nieuwenborgh, D., Vermeir, D. (2002). Preferred Answer Sets for Ordered Logic Programs. *Proceedings 10th European Conference on Logics in Artificial Intelligence (JELIA)*, 432-443.

Van Nieuwenborgh, D., Vermeir, D. (2002). Ordered Diagnosis, *Proceedings 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, 244-258.

Van Nieuwenborgh, D., Heymans, S., Vermeir, D. (2004). On Programs with Linearly Ordered Multiple Preferences. *Proceedings International Conference on Logic Programming (ICLP)*, 180-194.

Wakaki, T., Inoue, K., Sakama, C., Nitta, K. (2003). Computing Preferred Answer Sets in Answer Set Programming. *Proceedings 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, 259-273.

Wakaki, T., Inoue, K., Sakama, C., Nitta, K. (2004). The PLP System. *Proceedings 9th European Conference on Logics in Artificial Intelligence (JELIA)*, 706-709.

Wang, K., Zhou, L., Lin, F. (2000). Alternating Fixpoint Theory for Logic Programs with Priority. *Proceedings First International Conference on Computational Logic*, 164-178.

Zhang, Y., Foo, N. (1997). Answer sets for prioritized logic programs. *Proceedings International Logic Programming Symposium (ILPS)*, 69-83.