

Pstable Semantics for Logic Programs with Possibilistic Ordered Disjunction

Roberto Confalonieri, Juan Carlos Nieves, and Javier Vázquez-Salceda*

Universitat Politècnica de Catalunya
Dept. Llenguatges i Sistemes Informàtics
C/ Jordi Girona Salgado 1-3
E - 08034 Barcelona
{confalonieri, jcnieves, jvazquez}@lsi.upc.edu

Abstract. In this paper we define the semantics for capturing possibilistic ordered disjunction programs based on pstable semantics. The pstable semantics, based on paraconsistent logic, allows to treat inconsistency programs. Moreover being closer to possibilistic inference it allows to extend the necessity-values of the clauses to be considered, causing a higher level of comparison at the moment of selecting preferred pstable models of a possibilistic ordered disjunction programs. We compare the possibilistic pstable semantics for ordered disjunction programs with the recently defined possibilistic answer set semantics for the same class of logic programs.

1 Introduction

Logic programs with ordered disjunction (LPODs) are extended logic programs based on answer set semantics with a new *ordered disjunction* logic connector \times which can capture preferences between literals. An ordering between the valid models of a LPOD can then be specified through some comparison criteria, which allow to compare answer sets based on the preferences rules' satisfaction degree [2]. Therefore they are classes of logic programs that fit well in problems such as configuration management, policy monitoring and user preference representation and reasoning [2]. However, in realistic scenarios preference rules can be associated with a degree of relevance (or uncertainty) which can affect the established preference order, preventing the achievement of a single preferred solution, and some of these cases can be handled by possibilistic semantics for ordered disjunction programs [3].

Nevertheless, the syntax of a logic program with ordered disjunction allows the writing of programs such as $b \times a \leftarrow \text{not } a$, which are considered inconsistent because, under answer set semantics, they do not have any answer sets. Inconsistency in some cases has shown to be a desired feature of logic programs, because sometime the loss of information is worse compared to an inconsistent program not having any model. In the case of ordered disjunction programs the

* The authors are named by alphabetic order.

elimination of an inconsistent rule by the answer set-based reduction defined for this class of programs [2], can either prevent the program to have any solution, or to reach an order between the answer sets. In this paper we propose a *pstable semantics for ordered disjunction programs*, which can handle inconsistent ordered disjunction programs in the sense of no existence of models, and we extend it for capturing *possibilistic ordered disjunction programs* in order to cope with the degree of uncertainty in the reasoning process.

The advantages of our approach are several. First, pstable semantics, being based on paraconsistent logic, is less sensible to inconsistency and allows to treat inconsistent programs [8]. This means that we can provide valid pstable models of an ordered disjunction program whereas the program does not have any answer sets due to the presence of inconsistent rules. Secondly, being closer to possibilistic logic inference, it allows to consider higher degrees of necessity-values associated to the possibilistic atoms in the reasoning process, and sometimes can guarantee a higher level of comparison at the moment of selecting preferred pstable models of a possibilistic ordered disjunction programs. In such cases, we are able to compute a single preferred possibilistic pstable model, whereas the ordered disjunction program based on possibilistic answer set semantics cannot.

The rest of the paper is organized as follows. In the next section we provide some basic definitions. Section 3 describes the pstable semantics for ordered disjunction programs. In Section 4 we propose a possibilistic extension of the pstable semantics and we show the benefits of possibilistic pstable semantics for ordered disjunction. In the last section we outline some conclusions and future work directions. Throughout the paper, we use some simple examples to explain our semantics.

2 Background

This section presents the reader with some basic definitions *w.r.t* extended logic programs, pstable semantics, possibilistic logic, and logic programs with ordered disjunction, which are the basis for our approach.

2.1 Extended Logic Programs

We consider extended logic programs which have two kinds of negation: strong negation \neg and default negation *not*. A signature \mathcal{L} is a finite set of elements that we call atoms, where atoms negated by \neg are called extended atoms. Intuitively, *not a* is true whenever there is no reason to believe *a*, whereas $\neg a$ requires a proof of the negated atom. In the following we use the concept of atom without paying attention if it is an extended atom or not. A *literal* is either an atom *a*, called *positive literal*, or the negation of an atom *not a*, called *negative literal*. Given a set of atoms $\{a_1, \dots, a_n\}$, we write *not* $\{a_1, \dots, a_n\}$ to denote the set of atoms $\{\text{not } a_1, \dots, \text{not } a_n\}$. An extended *normal* rule, *r*, is a rule of the form $a \leftarrow b_1, \dots, b_n, \text{not } b_{n+1}, \dots, \text{not } b_{n+m}$ where *a* and each of the b_i are atoms for $1 \leq i \leq n + m$. In a slight abuse of notation we will denote such a clause by the formula $a \leftarrow \mathcal{B}^+, \text{not } \mathcal{B}^-$ where the set $\{b_1, \dots, b_n\}$ will be denoted by \mathcal{B}^+ , and

the set $\{b_{n+1}, \dots, b_{n+m}\}$ will be denoted by \mathcal{B}^- . A *constraint* is an extended rule of the form: $\leftarrow \mathcal{B}^+, \text{not } \mathcal{B}^-$. We define an *extended logic normal program* P as a finite set of extended normal rules and constraints. If the body of a normal rule is empty, then the clause is known as a *fact* and can be denoted just by a . We write \mathcal{L}_P , to denote the set of atoms that appear in the rules of P . We denote by $HEAD(P)$ the set $\{a \mid a \leftarrow \mathcal{B}^+, \text{not } \mathcal{B}^- \in P\}$. We will manage the strong negation \neg in our logic programs as it is done in Answer Set Programming. Basically, each atom $\neg a$ is replaced by a new atom symbol a' which does not appear in the language of the program and we add the constraint $\leftarrow a, a'$ to the program [1]. For managing the constraints in our logic programs, we will replace each rule of the form $\leftarrow \mathcal{B}^+ \text{not } \mathcal{B}^-$ by the following set of clauses: $v \leftarrow \mathcal{B}^+, \text{not } \mathcal{B}^-$. $w \leftarrow \text{not } y, v$. $y \leftarrow \text{not } z, v$. $z \leftarrow \text{not } w, v$., where v, w, y, z are new atom symbols which do not appear in \mathcal{L}_P .¹

2.2 Pstable Semantics

Pstable semantics is a recently introduced logic programming semantics which is inspired in paraconsistent logic and is defined in terms of a single reduction.

Definition 1 (Pstable reduction [8])

Let P be a normal program and M a set of atoms. The pstable reduction of P is defined as $RED(P, M) := \{a \leftarrow \mathcal{B}^+, \text{not } (\mathcal{B}^- \cap M) \mid a \leftarrow \mathcal{B}^+, \text{not } \mathcal{B}^- \in P\}$.

By considering the reduction $RED(P, M)$, the pstable semantics for normal programs is defined as follows:

Definition 2 (Pstable semantics [8])

Let P be a normal program and M a set of atoms. M is a pstable model of P if $RED(P, M) \Vdash M$.²

The pstable semantics allow to treat inconsistent logic programs such as $a \leftarrow \text{not } a$. In fact by applying pstable semantics, we can see that $M = \{a\}$ is a pstable model of $a \leftarrow \text{not } a$, while under the answer set semantics we would not have obtained any answer sets [5].

2.3 Possibilistic Logic

A *necessity-valued formula* is a pair $(\varphi \alpha)$ where φ is a classical logic formula and $\alpha \in (0, 1]$ is a positive number. The pair $(\varphi \alpha)$ expresses that the formula φ is *certain* at least to the level α , *i.e.* $N(\varphi) \geq \alpha$, where N is a necessity measure modeling our possibly incomplete state knowledge [4]. α is not a probability (as it is in probability theory) but it induces a certainty (or confidence) scale. This value is determined by the expert providing the knowledge base. A necessity-valued knowledge base is then defined as a finite set (*i.e.* a conjunction) of necessity-valued formulae. Dubois *et al.* [4] introduced a formal system for necessity-valued logic which is based in the following axioms schemata (propositional case):

¹ This approach was suggested in [7].

² It is written $P \Vdash M$ when $P \vdash_C M$ (logic consequence in classic logic) and M is a classical 2-valued model of P .

- (A1) $(\varphi \rightarrow (\psi \rightarrow \varphi)) \vdash 1$
(A2) $((\varphi \rightarrow (\psi \rightarrow \xi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi))) \vdash 1$
(A3) $((\neg\varphi \rightarrow \neg\psi) \rightarrow ((\neg\varphi \rightarrow \psi) \rightarrow \varphi)) \vdash 1$

For the axioms above, the following inference rules are defined:

- (GMP) $(\varphi \alpha), (\varphi \rightarrow \psi \beta) \vdash (\psi \min\{\alpha, \beta\})$
(S) $(\varphi \alpha) \vdash (\varphi \beta)$ if $\beta \leq \alpha$

We denote by \vdash_{PL} the inference under Possibilistic Logic without paying attention if the necessity-valued formulæ are using either a totally ordered set or a partially ordered set for expressing the levels of uncertainty.

2.4 Logic Programs with Ordered Disjunction

Logic programs with ordered disjunction (LPODs) are extended logic programs which allow the use of an ordered disjunction connector \times in the head of rules to express preferences among its literals [2]. The rule $r = C_1 \times \dots \times C_n \leftarrow b_1, \dots, b_m, \text{ not } b_{m+1}, \dots, \text{ not } b_{m+k}$ states that if the body is satisfied then some C_i must be in the answer set, if possible C_1 , if not then C_2 , and so on, and at least one of them must be true. Each of the C_i can be seen as a preference the user is interested into according to a desired order. One interesting characteristic of LPODs is that they provide a mean to represent preferences among answer sets by considering the satisfaction degree [2].

Definition 3. [2] *Let M be an answer set of an ordered disjunction program P . Then M satisfies the rule $r = C_1 \times \dots \times C_n \leftarrow b_1, \dots, b_m, \text{ not } b_{m+1} \dots, \text{ not } b_{m+k}$*

- to degree 1 if $b_j \notin M$ for some j ($1 \leq j \leq m$), or $b_i \in M$ for some i ($m+1 \leq i \leq m+k$),
- to degree j ($1 \leq j \leq n$) if all $b_l \in M$ ($1 \leq l \leq m$), no $b_i \in M$ ($m+1 \leq i \leq m+k$), and $j = \min\{r \mid C_r \in M, 1 \leq r \leq n\}$.

The satisfaction degree of an answer set M w.r.t a rule, denoted by $deg_M(r)$, provides a ranking of the answer sets of a LPOD, and a preference order on the answer sets can be obtained using some proposed combination strategies. In [2], the authors have proposed three criteria for comparing answer sets, respectively *cardinality*, *inclusion* and *Pareto*. In this paper we keep the three criteria and extend them for comparing possibilistic pstable models.

Example 1. Let P be the following ordered disjunction program.

$$\mathbf{r}_1 : b \times a \leftarrow \text{not } a. \quad \mathbf{r}_2 : a \leftarrow b. \quad \mathbf{r}_3 : b \leftarrow a.$$

The program P does not have any answer sets as answer set semantics do not allow to treat inconsistent programs (rule r_1 is inconsistent). In the next section we show how, by defining the pstable semantics for ordered disjunction programs, we can compute pstable models for inconsistent programs.

3 Pstable Semantics for Ordered Disjunction Programs

We specify a pstable semantics for ordered disjunction programs through two reductions based on the pstable semantics of Section 2.2 and on the original reductions defined by Brewka [2].

Definition 4 (Reduction $RED(r, M)_\times$)

Let $r = C_1 \times \dots \times C_n \leftarrow \mathcal{B}^+$, not \mathcal{B}^- be an ordered disjunction clause and M be a set of atoms. The \times reduct $RED(r, M)_\times$ of r is defined as follows: $RED(r, M)_\times := \{C_i \leftarrow \mathcal{B}^+, \text{ not } (\mathcal{B}^- \cap M) \mid C_i \in M \text{ and } M \cap \{C_1, \dots, C_{i-1}\} = \emptyset\}$.

Definition 5 (Reduction $RED(P, M)_\times$)

Let P be an ordered disjunction program and M a set of atoms. The \times reduct $RED(P, M)_\times$ of P is defined as follows: $RED(P, M)_\times = \bigcup_{r \in N} RED(r, M)_\times$.

Based on the reductions, we can define the pstable semantics extension for ordered disjunction programs.

Definition 6 (Pstable semantics for LPODs)

Let P be an ordered disjunction program P and M a set of atoms. M is pstable model of P if $RED(P, M)_\times \Vdash M$.

To see the benefits of the pstable semantics for ordered disjunction programs let us consider the following example.

Example 2. Let us consider the ordered disjunction program P in Example 1 and the set of atoms $M = \{b, a\}$. We can see that the $RED(P, M)_\times$ is:

$$\mathbf{r}_1 : b \leftarrow \text{not } a. \quad \mathbf{r}_2 : a \leftarrow b. \quad \mathbf{r}_3 : b \leftarrow a.$$

Hence, the set $M = \{b, a\}$ is a pstable model of P , as $RED(P, M)_\times \vdash_c M$ and M is a 2-valued model of $RED(P, M)_\times$.

Despite applying pstable semantics, pstable models of ordered disjunction programs are comparable by using the comparison criteria of [2]. Furthermore, the semantics we define is a generalization of the original semantics of Brewka, as formulated in the following theorem.

Theorem 1. *Let P be an ordered disjunction program and M a set of atoms. If M is an answer set of P then M is a pstable model of P .*

4 Pstable Semantics for LPPODs

In this section we recall the syntax of possibilistic ordered disjunction program [3] and extend the pstable semantics for ordered disjunction programs defined in Section 3 with possibilistic logic.

4.1 Syntax of LPPODs

The syntax of a possibilistic ordered disjunction program is based on the syntax of ordered disjunction rules (Section 2.4) and of possibilistic logic (Section 2.3).

A *possibilistic atom* is a pair $p = (a, q) \in \mathcal{A} \times \mathcal{Q}$ where \mathcal{A} is a set of atoms and (\mathcal{Q}, \leq) a finite lattice.³ The projection $*$ for any possibilistic atom p is defined as: $p^* = a$. Given a set of possibilistic atoms M , the generalization of $*$ over M is defined as: $M^* = \{p^* \mid p \in M\}$. Given (\mathcal{Q}, \leq) , a possibilistic ordered disjunction rule r is of the form: $\alpha : C_1 \times \dots \times C_n \leftarrow \mathcal{B}^+, \text{ not } \mathcal{B}^-$, where $\alpha \in \mathcal{Q}$ and $C_1 \times \dots \times C_n \leftarrow \mathcal{B}^+, \text{ not } \mathcal{B}^-$ is an ordered disjunction rule as defined in Section 2.4 with $\mathcal{B}^+ = \{b_1, \dots, b_m\}$ and $\mathcal{B}^- = \{b_{m+1}, \dots, b_{m+k}\}$.

The projection $*$ for a possibilistic ordered disjunction rule r , is $r^* = C_1 \times \dots \times C_n \leftarrow \mathcal{B}^+, \text{ not } \mathcal{B}^-$. $n(r) = \alpha$ is a necessity degree representing the certainty level of the information described by r . A possibilistic constraint c is of the form: $\text{TOP}_{\mathcal{Q}} \leftarrow \mathcal{B}^+, \text{ not } \mathcal{B}^-$, where $\text{TOP}_{\mathcal{Q}}$ is the top of the lattice (\mathcal{Q}, \leq) and $\leftarrow \mathcal{B}^+, \text{ not } \mathcal{B}^-$ is a constraint. Observe that any possibilistic constraint must have the top of the lattice (\mathcal{Q}, \leq) . This restriction is motivated by the fact that, like constraints in standard Answer Set Programming, the purpose of the possibilistic constraint is to eliminate possibilistic models. Hence, it is assumed that there is no uncertainty about the information captured by a possibilistic constraint. As in possibilistic ordered disjunction rules, the projection $*$ for a possibilistic constraint c is $c^* = \leftarrow \mathcal{B}^+, \text{ not } \mathcal{B}^-$.

A Logic Program with Possibilistic Ordered Disjunction (LPPOD) is a tuple of the form $P := \langle (\mathcal{Q}, \leq), N \rangle$ such that N is a finite set of possibilistic ordered disjunction rules and possibilistic constraints. The generalization of $*$ over P is defined as follows: $P^* := \{r^* \mid r \in N\}$. Notice that P^* is an ordered disjunction logic program. Given a possibilistic ordered disjunction program $P := \langle (\mathcal{Q}, \leq), N \rangle$, we define the α -cut of P denoted by P_α as $P_\alpha := \{r \mid r \in P, n(r) \geq \alpha\}$.

4.2 Pstable Semantics for LPPODs

We propose a possibilistic logic semantics which is close to the proof theory of possibilistic logic and pstable semantics. As in the pstable semantics definition, our approach is based on a syntactic reduction. We will consider sets of possibilistic atoms as interpretations. Hence, before defining the possibilistic ordered disjunction logic programming semantics, we introduce basic operators between sets of possibilistic atoms and a relation of order between them [6].

Definition 7. *Given \mathcal{A} a finite set of atoms and (\mathcal{Q}, \leq) be a lattice, we consider $\mathcal{PS} = 2^{\mathcal{A} \times \mathcal{Q}}$ as the finite set of all the possibilistic atoms sets induced by \mathcal{A} and \mathcal{Q} . Let $A, B \in \mathcal{PS}$, hence we define the operators \sqcap , \sqcup and \sqsubseteq as follows:*

$$\begin{aligned} A \sqcap B &= \{(x, GLB\{q_1, q_2\}) \mid (x, q_1) \in A \wedge (x, q_2) \in B\} \\ A \sqcup B &= \{(x, q) \mid (x, q) \in A \text{ and } x \notin B^*\} \cup \{(x, q) \mid x \notin A^* \text{ and } (x, q) \in B\} \cup \\ &\quad \{(x, LUB\{q_1, q_2\}) \mid (x, q_1) \in A \text{ and } (x, q_2) \in B\}. \\ A \sqsubseteq B &\iff A^* \subseteq B^*, \text{ and } \forall x, q_1, q_2, (x, q_1) \in A \wedge (x, q_2) \in B \text{ then } q_1 \leq q_2. \end{aligned}$$

The following reductions are the pstable possibilistic extensions of the reductions in Section 3.

³ In the paper we will consider only finite lattices.

Definition 8 (Reduction $PRED(r, M)_\times$)

Let $r = \alpha : C_1 \times \dots \times C_n \leftarrow \mathcal{B}^+$, $not \mathcal{B}^-$ be a possibilistic ordered disjunction clause and M be a set of atoms. The \times -possibilistic reduct $PRED(r, M)_\times$ of r is defined as follows: $PRED(r, M)_\times := \{\alpha : C_i \leftarrow \mathcal{B}^+, not (\mathcal{B}^- \cap M) \mid C_i \in M \text{ and } M \cap \{C_1, \dots, C_{i-1}\} = \emptyset\}$.

Definition 9 (Reduction $PRED(P, M)_\times$)

Let $P = \langle (Q, \leq), N \rangle$ be a possibilistic ordered disjunction program and M be a set of atoms. The \times -possibilistic reduct $PRED(P, M)_\times$ of P is defined as follows: $PRED(P, M)_\times = \bigcup_{r \in N} PRED(r, M)_\times$.

Example 3. Let P be a possibilistic ordered disjunction program such that $\mathcal{Q} = (\{0, 0.1, \dots, 0.9, 1\}, \leq)$, \leq be the standard relation between rational numbers and $\alpha \in \mathcal{Q}$ associated to each rule:

$$r_1 = \mathbf{0.3} : a \times b. \quad r_2 = \mathbf{0.8} : b \times a \leftarrow not a.$$

Let $M_1 = \{(a, 0.8)\}$ and $M_2 = \{(b, 0.8)\}$ be sets of possibilistic atoms. Taking M_1 as M we can see that $PRED(P, M)_\times$ is:

$$r_1 = \mathbf{0.3} : a. \quad r_2 = \mathbf{0.8} : a \leftarrow not a.$$

By considering the reduction $PRED(P, M)_\times$, we define the possibilistic pstable semantics as follows, which allows to test whether M is a possibilistic pstable model of a possibilistic ordered disjunction program P .

Definition 10 (Possibilistic Pstable Semantics)

Let $P = \langle (Q, \leq), N \rangle$ be a possibilistic ordered disjunction program and M be a set of possibilistic atoms such that M^* is a pstable model of P^* . M is a possibilistic pstable model of P if and only if $PRED(P, M^*)_\times \vdash_{PL} M$ and $\nexists M' \in \mathcal{PS}$ such that $M' \neq M$, $PRED(P, M'^*)_\times \vdash_{PL} M'$ and $M \sqsubseteq M'$.

Example 4. Let us continue with Example 3. We have already reduced the program and now we want to check if M is a possibilistic pstable model of P . First of all it is easy to see that M^* is a pstable model of P^* . Hence, we have to construct a proof in possibilistic logic for the possibilistic atom $(a, 0.8)$.

Let us prove $(a, 0.8)$ from $PRED(P, M^*)_\times$.⁴

Premises from $PRED(P, M^*)_\times$	From 3 and 1 by S
1. a 0.3	4. a 0.8
2. $\sim a \rightarrow a$ 0.8	
From 2 by possibilistic logical equivalence	
3. $a \vee a$ 0.8	

Therefore, we can say that $PRED(P, M^*)_\times \vdash_{PL} M$ is true (and similarly it can be proved that $M_2 = \{(b, 0.8)\}$ is a pstable model of P). Notice that it does not exist a possibilistic set M' such that $M' \neq M$, $PRED(P, M'^*)_\times \vdash_{PL} M'$ and $M \sqsubseteq M'$, hence we can conclude that M is a possibilistic pstable model of P .

⁴ When we treat a logic program as a theory, each negative literal $not a$ is replaced by $\sim a$, such that \sim is regarded as the negation in classical logic.

From Definition 10, we can observe that there is an important condition *w.r.t* the definition of a *possibilistic pstable model* of a possibilistic ordered disjunction program.

Proposition 1. *Let $P = \langle (Q, \leq), N \rangle$ be a possibilistic ordered disjunction program and M be a set of possibilistic atoms. If M is a possibilistic pstable model of P then M^* is a pstable model of P^* .*

Please observe that the pstable semantics for possibilistic ordered disjunction programs is different from the semantics defined in [3] as there are programs where the possibilistic pstable models do not correspond with the possibilistic answer sets. Nevertheless we can identify a relationship between the two semantics.

Proposition 2. *Let P be a possibilistic ordered disjunction program and M be a set of possibilistic atoms. If M is a possibilistic answer set of P , then the following conditions hold: (a) M^* is a pstable model of P , (b) there exists a possibilistic pstable model M' of P such that $M \sqsubseteq M'$ and $M^* = M'^*$.*

The proposition states that, whenever a possibilistic ordered disjunction program P has a possibilistic answer set M , there exists a possibilistic pstable model M' such that the main differences between M and M' are the necessity-values of their elements.

Example 5. Let P be the possibilistic ordered disjunction program of Example 3. We want to show how, under pstable semantics, its possibilistic stable models are $M_1 = \{(a, 0.8)\}$ and $M_2 = \{(b, 0.8)\}$ while, under answer set semantics, its possibilistic answer sets are $M_1 = \{(a, 0.3)\}$ and $M_2 = \{(b, 0.3)\}$. For doing so let us consider the reductions $PRED(P, M)_\times$ (Definition 9) and P_\times^M (Definition 8 in [3]) and apply them using M_1 :

$$\begin{array}{ll} PRED(P, M_1)_\times: & P_\times^{M_1}: \\ r_1 = \mathbf{0.3} : a. & r_1 = \mathbf{0.3} : a. \\ r_2 = \mathbf{0.8} : a \leftarrow not\ a. & \end{array}$$

We can see that under answer set semantics we have to cut the inconsistent rule r_2 , which triggers a lowering of the necessity degree of the possibilistic answer set, whereas by pstable semantics r_2 is kept and a higher necessity-value is inferred. This result follows from the fact that pstable semantics is closer to possibilistic logic inference than answer set semantics and it can lead to a maximum value choice.

4.3 Preferred Possibilistic Pstable Models

To distinguish between preferred possibilistic pstable models, we take the definition and the notation of satisfaction degree of M *w.r.t* a rule r as $deg_M(r)$ (see Section 2.4). In the following we define three comparison criteria (as in [2]), adapted to possibilistic pstable models.

Definition 11. *Let M_1 and M_2 be possibilistic pstable models of a possibilistic ordered disjunction logic program P . M_1 is possibilistic cardinality-preferred to M_2 , ($M_1 >_{pc} M_2$) iff $\exists i$ such that $|M_1^{i,\alpha}(P)| > |M_2^{i,\alpha}(P)|$ and $\forall j < i$, $|M_1^{j,\alpha}(P)| = |M_2^{j,\alpha}(P)|$, where $\alpha = \min\{n(M_1), n(M_2)\}$.*

Definition 12. Let M_1 and M_2 be possibilistic pstable models of a possibilistic ordered disjunction logic program P . M_1 is possibilistic inclusion-preferred to M_2 , ($M_1 >_{pi} M_2$) iff $\exists k$ such that $M_2^{k,\alpha}(P) \subset M_1^{k,\alpha}(P)$ and $\forall j < k$, $M_1^{j,\alpha}(P) = M_2^{j,\alpha}(P)$, where $\alpha = \min\{n(M_1), n(M_2)\}$.

Definition 13. Let M_1 and M_2 be possibilistic pstable models of a possibilistic ordered disjunction logic program P . M_1 is possibilistic pareto-preferred to M_2 , ($M_1 >_{pp} M_2$) iff $\exists r \in P$ such that $\text{deg}_{M_1}(r) < \text{deg}_{M_2}(r)$, and $\nexists r' \in P$ such that $\text{deg}_{M_1}(r') > \text{deg}_{M_2}(r')$, and $n(r) \geq \min\{n(M_1), n(M_2)\}$.

Notice that, applying the pstable semantics, we are able to keep higher degree of necessity-values formulæ. Let us consider the following examples which compare pstable semantics and answer semantics whenever computing preferred pstable models and answer sets respectively. The first example shows how pstable semantics allow to compare more than answer set semantics, while the second example shows how sometimes pstable models are differently preferred than answer sets due to higher cuts produced in the comparison criteria.

Example 6. Let P be the following possibilistic ordered disjunction program:

$$r_1 = \mathbf{0.5} : a \times b \leftarrow \text{not } c. \quad r_2 = \mathbf{0.5} : b \times a \leftarrow \text{not } d. \quad r_3 = \mathbf{0.8} : b \times a \leftarrow \text{not } a.$$

We can first see that $M_1^{pas} = \{(a, 0.5)\}$ and $M_2^{pas} = \{(b, 0.5)\}$ are possibilistic answer sets of P and $M_1^{pstable} = \{(a, 0.8)\}$ and $M_2^{pstable} = \{(b, 0.8)\}$ are possibilistic pstable models of P . We can notice that under answer set semantics no cut is produced in the comparison criteria and M_1^{pas} and M_2^{pas} are not comparable, while considering pstable semantics, the possibilistic necessity-value 0.8 derived from the pstable models triggers a cut of r_1 and r_2 . Hence, we can conclude that $M_2^{pstable}$ is preferred to $M_1^{pstable}$ according to all the criteria.

Example 7. Let P be the possibilistic ordered disjunction program of Example 6 with a different necessity-value in rule r_1 :

$$r_1 = \mathbf{0.3} : a \times b \leftarrow \text{not } c. \quad r_2 = \mathbf{0.5} : b \times a \leftarrow \text{not } d. \quad r_3 = \mathbf{0.8} : b \times a \leftarrow \text{not } a.$$

As in Example 6 it can be proved that $M_1^{pas} = \{(a, 0.5)\}$ and $M_2^{pas} = \{(b, 0.5)\}$ are possibilistic answer sets of P and $M_1^{pstable} = \{(a, 0.8)\}$ and $M_2^{pstable} = \{(b, 0.8)\}$ are possibilistic pstable models of P . However in this case the necessity-values allow to prefer M_1^{pas} to M_2^{pas} under answer set semantics, while considering pstable semantics the preference relation is inverted and $M_2^{pstable}$ is preferred to $M_1^{pstable}$. This can be justified by the fact that the pstable semantics allow to reach a higher degree of the necessity-values in the possibilistic pstable models and a higher cut is produced.

5 Conclusions

In this paper we have applied pstable semantics to ordered disjunction programs and extended it for capturing possibilistic ordered disjunction programs. The main advantages shown by our approach are essentially that: (a) using pstable

semantics (which is based on paraconsistent logic) it is possible to treat inconsistent ordered disjunction rules, thus inconsistent ordered disjunction programs can have a solution, while under answer set semantics they may not have; (b) as pstable semantics is closer to possibilistic logic inference, we are able to reach a higher degree of the necessity-value of possibilistic pstable models as it allows to take the maximum; and finally it allows to compare more models of a program because a higher cut is produced and we have been able to compare possibilistic pstable models whereas possibilistic answer sets are not comparable. Moreover, the semantics used in our approach is a generalization of the ordered disjunction program semantics in [2], as any answer sets of an ordered disjunction program, are pstable models as well (Theorem 1).

The two extensions we defined are computable, and as future work we aim to implement them. In parallel, we are exploring the use of ordered disjunction programs with our semantics in a realistic application to represent and reason about user preferences in the presence of uncertainty to enhance the Web service selection process [3].

Acknowledgements. This work is supported in part by the European Commission ALIVE project (FP7-215890). Javier Vázquez-Salceda is partially supported by the Ramón y Cajal program of the Spanish Ministry of Education and Science. We are grateful to anonymous referees for their useful comments.

References

1. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press, Cambridge (2003)
2. Brewka, G., Niemelä, I., Syrjänen, T.: Logic Programs with Ordered Disjunction. *Computational Intelligence* 20(2), 333–357 (2004)
3. Confalonieri, R., Nieves, J.C., Vázquez-Salceda, J.: Logic Programs with Possibilistic Ordered Disjunction. Research Report LSI-09-19-R, UPC - LSI (2009)
4. Dubois, D., Lang, J., Prade, H.: Possibilistic Logic. In: *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 3, pp. 439–513. Oxford University Press, Oxford (1994)
5. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing* 9(3/4), 365–386 (1991)
6. Nicolas, P., Garcia, L., Stéphan, I., Lafèvre, C.: Possibilistic Uncertainty Handling for Answer Set Programming. *Annals of Mathematics and Artificial Intelligence* 47(1-2), 139–181 (2006)
7. Osorio, M., Arrazola Ramírez, J.R., Carballido, J.L.: Logical Weak Completions of Paraconsistent Logics. *Journal of Logic and Computation* 18(6), 913–940 (2008)
8. Osorio, M., Pérez, J.A.N., Ramírez, J.R.A., Macías, V.B.: Logics with Common Weak Completions. *Journal of Logic and Computation* 16(6), 867–890 (2006)