Logic Programs with Ordered Disjunction: First-Order Semantics and Expressiveness

Vernon Asuncion[1,2], **Yan Zhang**[1], **Heng Zhang**[1]
[1] School of Computing, Engineering and Mathematics, University of Western Sydney, Australia
[2] Department of Computer Science and Engineering, Qatar University, Qatar

Abstract

Logic programs with ordered disjunction (LPODs) (Brewka 2002) generalize normal logic programs by combining alternative and ranked options in the heads of rules. It has been showed that LPODs are useful in a number of areas including game theory, policy languages, planning and argumentations. In this paper, we extend propositional LPODs to the first-order case, where a classical second-order formula is defined to capture the stable model semantics of the underlying first-order LPODs. We then develop a progression semantics that is equivalent to the stable model semantics but naturally represents the reasoning procedure of LPODs. We show that on finite structures, every LPOD can be translated to a first-order sentence, which provides a basis for computing stable models of LPODs. We further study the complexity and expressiveness of LPODs and prove that almost positive LPODs precisely capture first-order normal logic programs, which indicates that ordered disjunction itself and constraints are sufficient to represent negation as failure.

Introduction

Logic programs with ordered disjunction (LPODs) generalize normal logic programs by combining features of qualitative choice logic so that alternative and ranked options may be explicitly expressed in the heads of rules (Brewka 2002; Brewka, Benferhat, and Berre 2002). A LPOD contains a finite set of rules of the form

$$\alpha_1 \times \cdots \times \alpha_k \quad \leftarrow \quad \beta_1, \cdots, \beta_l, \mathsf{not}\,\gamma_1, \cdots, \mathsf{not}\,\gamma_m, \quad (1)$$

where $\alpha_i, \beta_j, \gamma_h$ ($1 \leq i \leq k, 1 \leq j \leq l, 1 \leq h \leq m$) are propositional atoms. Intuitively, rule (1) says that when the body is satisfied, then whenever it is possible, derive α_1, otherwise if possible, derive α_2, and so on. The semantics of an LPOD is defined in terms of the stable models of so-called *split programs* of the underlying LPOD.

Let us consider a simple program Π_1 from (Brewka 2002):

$$\begin{aligned} A \times B &\leftarrow \mathsf{not}\,C, \\ B \times C &\leftarrow \mathsf{not}\,D. \end{aligned}$$

Π_1 has the following four split programs:

$$\begin{aligned} A &\leftarrow \mathsf{not}\,C & A &\leftarrow \mathsf{not}\,C \\ B &\leftarrow \mathsf{not}\,D & C &\leftarrow \mathsf{not}\,D, \mathsf{not}\,B \end{aligned}$$

$$\begin{aligned} B &\leftarrow \mathsf{not}\,C, \mathsf{not}\,A & B &\leftarrow \mathsf{not}\,C, \mathsf{not}\,A \\ B &\leftarrow \mathsf{not}\,D & C &\leftarrow \mathsf{not}\,D, \mathsf{not}\,B. \end{aligned}$$

Then the class of stable models of Π consists of all stable models of these four split programs, which is $\{\{A, B\}, \{B\}, \{C\}\}$. Then by integrating proper preference relation among these stable models, the preferred stable models can be obtained for an LPOD.

There have been several extensions of LPODs in recent years: Karger *et al* (2008) extended LPODs by allowing both ordered and unordered disjunction in the heads of rules; Confalonieri *et al* (2010) recently defined a possibilistic semantics for LPODs in order to handle uncertainty; and Cabalar (2011) also proposed a direct translation from LPODs to normal logic programs via the logic of Here-and-There. It has been argued that LPODs provide a natural way to deal with preference in reasoning that are useful in various applications such as game theory, policy languages, planning and argumentations (Brewka 2002; Cabalar 2011; Confalonieri et al. 2010).

On the other hand, in recent years, Answer Set Programming (ASP) has been generalized to arbitrary first-order sentences (Ferraris, Lee, and Lifschitz 2011). One challenging research along this direction is to establish proper logical and computational foundations for promoting useful functionalities in existing ASP paradigm to the first-order level. A number of topics in this aspect have been investigated and relevant properties revealed, e.g., (Asuncion et al. 2012; Asuncion, Zhang, and Zhou 2013; Lee and Meng 2011; Babb and Lee 2012). One major advantage of first-order ASP is that it provides a succinct declarative language, in which the underlying problem constraints (rules) may be completely separated from concrete problem instances, and hence more flexible for problem representation and modeling (Lin and Zhou 2011).

In this paper, we study the semantics and expressiveness of LPODs on the first-order level. We make the following main contributions towards this topic:

1. Following the style of general stable model semantics

(Ferraris, Lee, and Lifschitz 2011), we define the stable model semantics for first-order LPODs via a classical second-order sentence, and show that this semantics is a correct lifting of Brewka's original LPODs semantics to the first-order level. We then develop a progression semantics that is equivalent to the second-order based semantics but naturally represents the reasoning procedure of LPODs.

2. We propose a translation from LPODs to first-order sentences on finite structures. We argue that this translation actually may be used as a computational basis for developing an LPOD solver.
3. We address the complexity and expressiveness issues of LPODs. We show that LPODs data complexity is NP-complete, that remains true even for positive LPODs. We also prove that almost positive LPODs precisely capture the full class of first-order normal logic programs, which indicates that ordered disjunction itself and constraints are sufficient to represent negation as failure.
4. We further consider preferred LPODs, and provide a logic characterization of such preference semantics, which again, generalizes the corresponding preferred propositional LPODs.

The rest of this paper is organized as follows. In section 2 we focus on the syntax and semantics of first-order LPODs. In section 3 we then present a progression semantics for first-order LPODs, which we believe is more intuitive to reflect the underlying derivation process embedded in LPODs. Considering to compute stable models of an LPOD, in section 4 we propose a translation from LPODs to first-order sentences on finite structures, so that the problem of computing stable models of an LPOD can be viewed as a SAT solving problem. We study the complexity and expressiveness issues in section 5. In section 6 we define the preferred stable models for LPODs and provide a logic formulation for that. Finally, we conclude the paper with some remarks in section 7.

First-order LPODs: Syntax and Semantics

We consider a second-order logic but without function symbols (i.e., functions of arities not greater than 0). A *signature* τ is a set of symbols of the form $\{c_1, \ldots, c_m, P_1, \ldots, P_n\}$ such that c_i (for $1 \leq i \leq m$) are *constant* symbols and P_j (for $1 \leq j \leq n$) are *predicate* symbols. A *structure* $\mathcal{M}$ of signature τ (or a τ-structure) is a tuple of the form

$$(M, c_1^{\mathcal{M}}, \ldots, c_m^{\mathcal{M}}, P_1^{\mathcal{M}}, \ldots, P_n^{\mathcal{M}}), \tag{2}$$

where M is the *domain* of $\mathcal{M}$, which we will mostly refer to as $Dom(\mathcal{M})$; $c_i^{\mathcal{M}}$ and $P_j^{\mathcal{M}}$ are the respective interpretations of the constant and predicate symbols.

Let $\mathcal{M}$ be a τ-structure and $\mathcal{M}'$ a τ'-structure such that $\tau' \subseteq \tau$ and where (1) $Dom(\mathcal{M}') = Dom(\mathcal{M})$; (2) $c^{\mathcal{M}'} = c^{\mathcal{M}}$ for each constant symbol c of τ'; and (3) $P^{\mathcal{M}'} = P^{\mathcal{M}}$ for each predicate symbol P of τ'. We refer $\mathcal{M}'$ as the *restriction* of $\mathcal{M}$ to the signature of τ', which we denote by $\mathcal{M}\upharpoonright_{\tau'}$, i.e., $\mathcal{M}' = \mathcal{M}\upharpoonright_{\tau'}$. Symmetrically, $\mathcal{M}$ would be called an *expansion* of $\mathcal{M}'$ to the signature τ.

The Syntax

An *ordered disjunction rule* is a construct of the form:

$$\alpha_1 \times \ldots \times \alpha_k \;\leftarrow\; \beta_1, \ldots, \beta_l, \mathsf{not}\,\gamma_1, \ldots, \mathsf{not}\,\gamma_m, \tag{3}$$

where:

- Each α_i (for $1 \leq i \leq k$) is a standard atom $P(\mathbf{x})$ for some predicate P and tuple of terms (variables or constants) $\mathbf{x}$. If $k = 0$, then (3) is called a *constraint*.
- Each β_i $(1 \leq i \leq l)$ and γ_i $(1 \leq i \leq m)$ are either a standard or an equality atom.
- By $Head(r)$, we denote the ordered expression $\alpha_1 \times \ldots \times \alpha_k$, that corresponds to the head of r. We also denote $Head(r)_i = \alpha_i$.
- As usual, by $Body(r)$, we denote the set of literals $\{\beta_1, \ldots, \beta_l, \mathsf{not}\,\gamma_1, \ldots, \mathsf{not}\,\gamma_m\}$.
- Similarly, by $Pos(r)$ and $Neg(r)$, we denote the sets of atoms $\{\beta_1, \ldots, \beta_l\}$ and $\{\gamma_1, \ldots, \gamma_m\}$, respectively.

A *first-order logic program with ordered disjunction*, or simply called (first-order) LPOD, Π, is a finite set of ordered disjunction rules of the form (3). If for all rules $r \in \Pi$ we have that $k \leq 1$ in $Head(r) = \alpha_1 \times \ldots \times \alpha_k$, then Π becomes a normal logic program. It is obvious that this first-order LPOD syntax naturally extends Brewka's original propositional logic programs with ordered disjunction (Brewka 2002) to the first-order case.

For a given LPOD Π, a predicate is called *intensional* if it occurs at least once in the head of some rule in Π. All other predicates are called *extensional*. For convenience, we denote by $\tau(\Pi)$, $\mathcal{P}_{ext}(\Pi)$ and $\mathcal{P}_{int}(\Pi)$, the signature of Π, the set of extensional predicates and the set of intensional predicates, respectively. Furthermore, we also use $\tau_{ext}(\Pi)$ and $\tau_{int}(\Pi)$ to denote the sub-signatures of $\tau(\Pi)$ which only contain extensional and intensional predicates respectively, together with all constant symbols of $\tau(\Pi)$. When there is no confusion in the context, we often omit the parameter Π in these notions.

The Stable Model Semantics

For a rule of the form (3), we denote by $\widehat{Body(r)}$, the formula $\beta_1 \wedge \cdots \wedge \beta_l \wedge \neg\gamma_1 \wedge \cdots \wedge \neg\gamma_m$, and $Head(r)^{FO}$, the following formula

$$\alpha_1 \vee (\neg\alpha_1 \wedge \alpha_2) \vee (\neg\alpha_1 \wedge \neg\alpha_2 \wedge \alpha_3) \vee \ldots \vee (\neg\alpha_1 \wedge \neg\alpha_2 \wedge \ldots \wedge \neg\alpha_{k-1} \wedge \alpha_k).$$

Note that Cabalar (2011) has showed the strong equivalence of $\alpha_1 \times \cdots \times \alpha_k$ and $Head(r)^{FO}$ for the propositional case, that can be easily extended to our first-order case. Then for an LPOD Π, by $\widehat{\Pi}$, we denote the conjunction

$$\bigwedge_{r \in \Pi} (\widehat{Body(r)} \rightarrow Head(r)^{FO}). \tag{4}$$

Definition 1 *Let Π be an LPOD with tuple of distinct (intensional) predicates $\mathbf{P} = P_1 P_2 \ldots P_n$ and $\mathcal{M}$ a $\tau(\Pi)$-structure. Then we say that $\mathcal{M}$ is a* stable model *of Π iff it is a model of the following second-order sentence:*

$$\mathrm{SM}_{\mathbf{P}}(\Pi) = \widehat{\Pi} \wedge \neg\exists \mathbf{U}(\mathbf{U} < \mathbf{P} \wedge \widehat{\Pi}(\mathbf{U})^*), \tag{5}$$

where:

- $\mathbf{U} = U_1U_2 \ldots U_n$ *is a fresh tuple of distinct predicates matching* $\mathbf{P}$*;*
- $\mathbf{U} < \mathbf{P}$ *denotes the formula*

$$\bigwedge_{1\leq i\leq n} \forall\mathbf{x}(U_i(\mathbf{x}) \to P_i(\mathbf{x})) \wedge \neg \bigwedge_{1\leq i\leq n} \forall\mathbf{x}(P_i(\mathbf{x}) \to U_i(\mathbf{x}));$$

- $\widehat{\Pi}(\mathbf{U})^*$ *denotes the formula*

$$\bigwedge_{\substack{r\in\Pi,\\ r \text{ of form the (3)}}} \forall\mathbf{x}_r(\, \beta_1^* \wedge \ldots \wedge \beta_l^* \wedge \neg\gamma_1 \wedge \ldots \wedge \neg\gamma_m \to (\alpha_1 \times \ldots \times \alpha_k)^* \,),$$

where $(\alpha_1 \times \ldots \times \alpha_k)^*$ *denotes the formula:*

$$\alpha_1^* \vee (\neg\alpha_1 \wedge \alpha_2^*) \vee (\neg\alpha_1 \wedge \neg\alpha_2 \wedge \alpha_3^*) \vee \ldots \vee (\neg\alpha_1 \wedge \neg\alpha_2 \wedge \ldots \wedge \neg\alpha_{k-1} \wedge \alpha_k^*), \quad (6)$$

and such that $P_i(\mathbf{x})^* = U_i(\mathbf{x})$ *for any atomic formula* $P_i(\mathbf{x})$ *with* $1 \leq i \leq n$*, and* $Q(\mathbf{y})^* = Q(\mathbf{y})$ *otherwise.*

It is not difficult to see that Definition 1 is a counterpart of the second-order definition for the general stable model semantics (Ferraris, Lee, and Lifschitz 2011), with the additional operation (6). In the following, we will show that Definition 1 is also a generalization of the semantics for the original propositional LPODs, and hence, our first-order LPOD stable model semantics is a proper uplifting from the propositional level.

Let τ be a signature of Π with the sets $\mathcal{P}_{ext}$ and $\mathcal{P}_{int}$ of extensional and intensional predicates respectively, and let $\mathcal{M}$ be a τ-structure. Then we define $I_\mathcal{M}$ as follows:

$$I_\mathcal{M} := \{P(\mathbf{a}) \mid P \in (\mathcal{P}_{ext} \cup \mathcal{P}_{int}) \text{ and } \mathbf{a} \in P^\mathcal{M}\},$$

i.e., the set of propositional atoms corresponding to the extents of the predicates of τ under $\mathcal{M}$. In addition, by $Ext_\mathcal{M}$, define the set of rules (which are facts) as follows:

$$Ext_\mathcal{M} := \{Q(\mathbf{a}) \ \leftarrow \ \mid Q \in \mathcal{P}_{ext} \text{ and } \mathbf{a} \in Q^\mathcal{M}\},$$

i.e., the set of "facts" corresponding to the extensional database under $\mathcal{M}$.

In addition to the two previous notions, we now introduce the notion of a grounded program. Let Π be an LPOD and $\mathcal{M}$ a $\tau(\Pi)$-structure with domain denoted by $Dom(\mathcal{M})$. Then by $\Pi\mid_{Dom(\mathcal{M})}$, let us denote the propositional program obtained from Π by (1) simultaneously replacing the variables by domain elements from $Dom(\mathcal{M})$ in all possible ways, and the constant symbols by their corresponding interpretation in $Dom(\mathcal{M})$ under $\mathcal{M}$; and (2) removing all rules that contain false instantiations of equality predicates and deleting those true instantiations of equalities from the remaining rules.

Before we present the following result, we also need a formal definition of split programs for a given propositional LPOD. Let Π be a propositional LPOD consisting of rules of the form of (3), but all those α's, β's and γ's are propositional atoms. Then a *split program* of Π is a normal logic program Π' obtained from Π by replacing each rule of (3) in Π, for some $i \leq k$, by

$$\alpha_i \leftarrow \beta_1, \ldots, \beta_l, \mathsf{not}\,\gamma_1, \ldots, \mathsf{not}\,\gamma_m, \mathsf{not}\,\alpha_1, \cdots, \mathsf{not}\,\alpha_{i-1}. \quad (7)$$

Theorem 1 *Let* Π *be an LPOD and* $\mathcal{M}$ *a* $\tau(\Pi)$*-structure. Then* $\mathcal{M}$ *is a stable model of* Π *iff* $I_\mathcal{M}$ *is a stable model of some split program of* $\Pi|_{Dom(\mathcal{M})} \cup Ext_\mathcal{M}$.

Progression Semantics: An Alternative

According to Definition 1, the semantics of first-order LPODs is defined via a second-order sentence, which hardly reveals the rule based reasoning feature of the underlying LPOD. In this section, we provide an alternative semantics for FO LPODs - the progression semantics, which is an extension of Zhang and Zhou's progression semantics for first-order normal logic program (Zhang and Zhou 2010). As we will see, the important feature of such semantics is that it naturally represents the reasoning procedure through the progression stages.

Definition 2 *Let* Π *be a FO LPOD and* $\mathcal{M}$ *a* $\tau(\Pi)$*-structure*

$$(Dom(\mathcal{M}), c_1^\mathcal{M}, \ldots, c_r^\mathcal{M}, Q_1^\mathcal{M}, \ldots, Q_s^\mathcal{M}, P_1^\mathcal{M}, \ldots, P_n^\mathcal{M})$$

such that c_i *(for* $1 \leq i \leq r$*),* Q_i *(for* $1 \leq i \leq s$*), and* P_i *(for* $1 \leq i \leq n$*) are its constant symbols, extensional and intensional predicate symbols, respectively. We define the* $\tau(\Pi)$*-structure* $\mathcal{M}^{t+1}(\Pi)$ *inductively as follows:*

$$\mathcal{M}^0(\Pi) = (Dom(\mathcal{M}), c_1^{\mathcal{M}^0(\Pi)}, \ldots, c_r^{\mathcal{M}^0(\Pi)}, Q_1^{\mathcal{M}^0(\Pi)}, \ldots, Q_s^{\mathcal{M}^0(\Pi)}, P_1^{\mathcal{M}^0(\Pi)}, \ldots, P_n^{\mathcal{M}^0(\Pi)}), \quad (8)$$

where $c_i^{\mathcal{M}^0(\Pi)} = c_1^\mathcal{M}$ *(for* $1 \leq i \leq r$*),* $Q_i^{\mathcal{M}^0(\Pi)} = Q_i^\mathcal{M}$ *(for* $1 \leq i \leq s$*), and* $P_i^{\mathcal{M}^0(\Pi)} = \emptyset$ *(for* $1 \leq i \leq n$*);*

$$\mathcal{M}^{t+1}(\Pi) = \mathcal{M}^t(\Pi) \cup \{\alpha_i\eta \mid \exists r \in \Pi \textit{ with } Head(r) = \alpha_1 \times \ldots \times \alpha_i \times \ldots \times \alpha_k \textit{ and } \eta \textit{ an assignment such that:}$$
$$\textit{1. } Pos(r)\eta \subseteq \mathcal{M}^t(\Pi) \textit{ and } Neg(r)\eta \cap \mathcal{M} = \emptyset;$$
$$\textit{2. } i \textit{ (for } 1 \leq i \leq k \textit{) is the largest } i \textit{ such that } \{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} \cap \mathcal{M} = \emptyset\,\}. \quad (9)$$

Then let $\mathcal{M}^\infty(\Pi) = \bigcup_{0\leq t\leq\infty} \mathcal{M}^t(\Pi)$.

Since $\mathcal{M}^t(\Pi) \subseteq \mathcal{M}^{t+1}(\Pi)$, we know that the sequence $\mathcal{M}^0, \mathcal{M}^1, \cdots$ is monotonic and $\mathcal{M}^t(\Pi)$ always converges to its fixpoint. Also note that when $Head(r) = \alpha_1$ (i.e., r is a normal rule), then we will have in Item 2 in (9) that $i = 1$ will automatically be the largest i such that $\{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} \cap \mathcal{M} = \emptyset$. From this fact, the following proposition shows that the progression characterization captures the stable models when Π is a normal logic program.

Proposition 1 *Let* Π *be a normal logic program and* $\mathcal{M}$ *a* $\tau(\Pi)$*-structure. Then* $\mathcal{M}$ *is a stable model of* Π *iff* $\mathcal{M}^\infty(\Pi) = \mathcal{M}$.

Proof: If Π is a normal program, then we will have by default in Item 2 in (9) of Definition 2 that $i = 1$ will be the largest i such that $\{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} \cap \mathcal{M} = \emptyset$, since $\{\alpha_1, \ldots, \alpha_{i-1}\} = \emptyset$ in this case. Therefore, we can omit Item 2 in (9) so that $\mathcal{M}^{t+1}(\Pi)$ will be as originally defined in (Zhang and Zhou 2010) for normal programs. □

Theorem 2 shows that the progression semantics coincides with the second-order stable model semantics of LPODs as defined in Definition 1. We first present the following lemma which is needed in the proof of Theorem 2.

Lemma 1 *If $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$ then $\mathcal{M} \models \widehat{\Pi}$.*

Theorem 2 *Let Π be an LPOD and $\mathcal{M}$ a $\tau(\Pi)$-structure. Then $\mathcal{M}$ is a stable model of Π iff $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$.*

Proof: ($\Rightarrow$) First we show that $\mathcal{M}^{\infty} \subseteq \mathcal{M}$ by showing $\mathcal{M}^{t} \subseteq \mathcal{M}$ holds for all $t \geq 0$ by induction. Clearly, $\mathcal{M}^{0}(\Pi) \subseteq \mathcal{M}$ holds by the definition of $\mathcal{M}^{0}(\Pi)$ since all the intensional relations are set to empty. Now assume that $\mathcal{M}^{t'}(\Pi) \subseteq \mathcal{M}$ holds for all $0 \leq t' \leq t$ and let $P(\mathbf{a}) \in \mathcal{M}^{t+1}(\Pi)$ such that $P(\mathbf{a}) \notin \mathcal{M}^{t}(\Pi)$. Then by the definition of $\mathcal{M}^{t+1}(\Pi)$, we have that there exists a rule $r \in \Pi$ with $Head(r) = \alpha_1 \times \ldots \times \alpha_i \times \ldots \times \alpha_k$ and assignment η such that:

1. $P(\mathbf{a}) = \alpha_i\eta$;
2. $Pos(r)\eta \subseteq \mathcal{M}^{t}(\Pi)$ and $Neg(r) \cap \mathcal{M} = \emptyset$;
3. i (for $1 \leq i \leq k$) is the largest i such that $\{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} \cap \mathcal{M} = \emptyset$.

Then since $Pos(r)\eta \subseteq \mathcal{M}^{t}(\Pi)$ and $Neg(r) \cap \mathcal{M} = \emptyset$, and where $\mathcal{M}^{t}(\Pi) \subseteq \mathcal{M}$, it follows that $\mathcal{M} \models \widehat{Body(r)}\eta$. Thus, since $\mathcal{M} \models \widehat{\Pi}$ (because $\mathcal{M} \models \mathrm{SM}_{\mathbf{P}}[\Pi]$), then it follows that $\mathcal{M} \models Head(r)^{FO}\eta \equiv (\neg\alpha_1 \wedge \ldots \wedge \neg\alpha_{i-1} \wedge \alpha_i)\eta$, which implies that $\alpha_i\eta = P(\mathbf{a}) \in \mathcal{M}$. Therefore, we had shown that $\mathcal{M}^{\infty}(\Pi) \subseteq \mathcal{M}$. Now let us assume on the contrary that it is also the case that $\mathcal{M}^{\infty}(\Pi) \subset \mathcal{M}$ and we show we derive a contradiction. Then define a $\tau(\Pi) \cup \{U_1, \ldots, U_n\}$-structure $\mathcal{U}$ as follows:

- $c^{\mathcal{U}} = c^{\mathcal{M}}$ for every constant c of $\tau(\Pi)$;
- $Q^{\mathcal{U}} = Q^{\mathcal{M}}$ for every extensional predicate of $\tau(\Pi)$;
- $P_i^{\mathcal{U}} = P_i^{\mathcal{M}}$ for every intensional predicate P_i of $\tau(\Pi)$, such that $1 \leq i \leq n$.
- $U_i^{\mathcal{U}} = P_i^{\mathcal{M}^{\infty}(\Pi)}$ for every predicate U_i in $\{U_1, \ldots, U_n\}$, with $1 \leq i \leq n$.

Now we will show that $\mathcal{U} \models \mathbf{U} < \mathbf{P}$ and $\mathcal{U} \models \widehat{\Pi}(\mathbf{U})^*$. The first part $\mathcal{U} \models \mathbf{U} < \mathbf{P}$ follows from the fact that $\mathcal{M}^{\infty}(\Pi) \subset \mathcal{M}$ and by the way the $\tau(\Pi) \cup \{U_n, \ldots, U_n\}$-structure $\mathcal{U}$ is constructed from $\mathcal{M}$ and $\mathcal{M}^{\infty}(\Pi)$ above. Now we show the second part. So assume for some rule r in Π of the form (3) and some assignment η that $\mathcal{U} \models \widehat{Body(r)}^*\eta$. Then since $\mathcal{U} \models \mathbf{U} < \mathbf{P}$ and by the definition of $\mathcal{U}$, it follows that $Pos(r)\eta \subseteq \mathcal{M}^{\infty}(\Pi)$, which implies that $Pos(r)\eta \subseteq \mathcal{M}^{t}(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi) \subset \mathcal{M}$ for some $t \geq 0$. Then since $Pos(r)\eta \subset \mathcal{M}$, then $\mathcal{M} \models \widehat{Body(r)}\eta$ as well. Then since $\mathcal{M} \models \widehat{\Pi}$, we have that $\mathcal{M} \models Head(r)^{FO}\eta$, which further implies that $\mathcal{M} \models \neg\alpha_1 \wedge \ldots \wedge \neg\alpha_{i-1} \wedge \alpha_i$ for some $1 \leq i \leq k$. Then by Item 2 of (9), this implies that $\alpha_i\eta \in \mathcal{M}^{t+1}(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$ since i will be the largest i such that $\{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} \cap \mathcal{M} = \emptyset$, which further implies that $\mathcal{U} \models Head(r)^*\eta$ by the way $\mathcal{U}$ was constructed from $\mathcal{M}^{\infty}(\Pi)$ and since $\mathcal{M}^{t+1}(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$. Therefore, because we have shown that $\mathcal{U} \models \widehat{\Pi}(\mathbf{U})^*$, then we now have a contradiction since we initially assumed that $\mathcal{M} \models \mathrm{SM}_{\mathbf{P}}(\Pi)$, and where this implies that $\mathcal{M} \models \neg\exists\mathbf{U}(\mathbf{U} < \mathbf{P} \wedge \widehat{\Pi}(\mathbf{U})^*)$ (which is contradicted by the existence of the structure $\mathcal{U}$).

($\Leftarrow$) Now let us assume $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$ but, on the contrary that $\mathcal{M} \not\models \mathrm{SM}_{\mathbf{P}}[\Pi]$. Since by Lemma 1 we have that $\mathcal{M} \models \widehat{\Pi}$, then this implies that $\mathcal{M} \not\models \neg\exists\mathbf{U}(\mathbf{U} < \mathbf{P} \wedge \widehat{\Pi}(\mathbf{U})^*)$. Thus, assume that $\mathcal{U}$ is a $\tau(\Pi) \cup \{U_n, \ldots, U_n\}$-structure such that that $\mathcal{U} \models \mathbf{U} < \mathbf{P} \wedge \widehat{\Pi}(\mathbf{U})^*$. Then we will show by induction on t that $P_i^{\mathcal{M}^{t}(\Pi)} \subseteq U_i^{\mathcal{U}}$, for $1 \leq i \leq n$. Therefore, since $\mathcal{U} \models \mathbf{U} < \mathbf{P}$ (i.e., $\mathbf{U}$ is "strictly smaller" than $\mathbf{P}$), then this contradicts the assumption that $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$ (because $\mathcal{M}^{\infty}(\Pi)$ would be "strictly less" than $\mathcal{M}$ in this case). The base case for $\mathcal{M}^{0}(\Pi)$ clearly holds since $P_i^{\mathcal{M}^{0}(\Pi)} = \emptyset$, for $1 \leq i \leq n$. So let us assume that $P_i^{\mathcal{M}^{t}(\Pi)} \subseteq U_i^{\mathcal{U}}$, for $1 \leq i \leq n$, and consider $\mathcal{M}^{t+1}(\Pi)$. Indeed, let $P(\mathbf{a}) \in \mathcal{M}^{t+1}(\Pi) \setminus \mathcal{M}^{t}(\Pi)$. Then by the definition of $\mathcal{M}^{t+1}(\Pi)$ as we find in (9) of Definition (2), there exists a rule $r \in \Pi$ with $Head(r) = \alpha_1 \times \ldots \times \alpha_i \times \ldots \times \alpha_k$ such that:

1. $Pos(r)\eta \subseteq \mathcal{M}^{t}(\Pi)$ and $Neg(r)\eta \cap \mathcal{M} = \emptyset$;
2. i (for $1 \leq i \leq k$) is the largest i such that $\{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} \cap \mathcal{M} = \emptyset$,

and where $P_i(\mathbf{a}) = \alpha_i\eta$. Then it is not difficult to see that $\mathcal{U} \models \widehat{Body(r)}^*\eta$ as well since $P_i^{\mathcal{M}^{t}(\Pi)} \subseteq U_i^{\mathcal{U}}$, for $1 \leq i \leq n$. Then since $\mathcal{U} \models \widehat{\Pi}(\mathbf{U})^*$, we have that $\mathcal{U} \models Head(r)^*$, and where in particular, $\mathcal{U} \models (\neg\alpha_1 \wedge \ldots \wedge \neg\alpha_{i-1} \wedge \alpha_i^*)\eta$, so that $\alpha_i^*\eta = P(\mathbf{a}) \in \mathcal{U}$. This completes the proof of Theorem 2. □

Example 1 *Let Π be the following LPOD with the two rules:*

$$r_1 : P(x) \times Q(x) \times R(x) \;\leftarrow\; S(x), \tag{10}$$
$$r_2 : T(x) \times U(x) \;\leftarrow\; Q(x), \tag{11}$$

such that S is the only extensional predicate. Now let $\mathcal{M}$ be a $\tau(\Pi)$-structure such that:

$$\mathcal{M} = (\{a,b\}, S^{\mathcal{M}} = \{a,b\}, P^{\mathcal{M}} = \{a\}, Q^{\mathcal{M}} = \{b\}, R^{\mathcal{M}} = \emptyset, T^{\mathcal{M}} = \emptyset, U^{\mathcal{M}} = \{b\}).$$

Now based on (8) of Definition 2, we have that

$$\mathcal{M}^{0}(\Pi) = (\{a,b\}, S^{\mathcal{M}} = \{a,b\}, P^{\mathcal{M}} = \emptyset, Q^{\mathcal{M}} = \emptyset, R^{\mathcal{M}} = \emptyset, T^{\mathcal{M}} = \emptyset, U^{\mathcal{M}} = \emptyset),$$

i.e., all the intensional relations are initially set to empty. Now let us compute $\mathcal{M}^{1}(\Pi)$. From (9) of Definition 2, we have that $P(a) \in \mathcal{M}^{1}(\Pi)$ since $\alpha_{i=1}\eta = P(a)$ (with $\eta : x \longrightarrow a$) is the largest i such that $\{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} = \emptyset \cap \mathcal{M} = \emptyset$. Similarly, we also have that $Q(b) \in \mathcal{M}^{1}(\Pi)$ since in this case, with $\eta : x \longrightarrow b$, we have that $\alpha_{i=2}\eta = Q(b)$ is the largest i such that $\{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} = \{P(b)\} \cap \mathcal{M} = \emptyset$ (i.e., since $P(b) \notin \mathcal{M}$), so that we now have

$$\mathcal{M}^{1}(\Pi) = (\{a,b\}, S^{\mathcal{M}} = \{a,b\}, P^{\mathcal{M}} = \{a\}, Q^{\mathcal{M}} = \{b\}, R^{\mathcal{M}} = \emptyset, T^{\mathcal{M}} = \emptyset, U^{\mathcal{M}} = \emptyset).$$

Similarly, we will have $\mathcal{M}^{2}(\Pi)$ as follows:

$$(\{a,b\}, S^{\mathcal{M}} = \{a,b\}, P^{\mathcal{M}} = \{a\}, Q^{\mathcal{M}} = \{b\}, R^{\mathcal{M}} = \emptyset, T^{\mathcal{M}} = \emptyset, U^{\mathcal{M}} = \{b\}),$$

and further, we can obtain $\mathcal{M}^2(\Pi) = \mathcal{M}^t(\Pi)$ for all $t \geq 3$, that is $\mathcal{M}^\infty(\Pi) = \mathcal{M}^2(\Pi)$. Therefore, since $\mathcal{M}^\infty(\Pi) = \mathcal{M}$, we have by Theorem 2 that $\mathcal{M}$ is a stable model of Π. $\Box$

From LPODs to First-order Formulas

While Definition 2 provides an alternative semantics for FO LPODs, which represents the program reasoning feature through progression process, it, however, still does not reveal much information about how a stable model of a given LPOD may be computed. In this section, we show how a variant of the ordered completion (Asuncion et al. 2012) for normal logic programs can capture the stable models of first-order LPODs, which, like demonstrated in (Asuncion et al. 2012), may be viewed as a computational basis for an FO LPOD solver development.

For this purpose, for a given pair of predicates (P, Q) (where P and Q can be the same), by $\leq_{PQ}$, we denote a new predicate such that its arity is the sum of the arities of P and Q. We refer to such predicates as the *comparison predicates*. The intuitive meaning of atom $\leq_{PQ}(\mathbf{x}, \mathbf{y})$ is that: $Q(\mathbf{y})$ is true only if $P(\mathbf{x})$ is true.

Definition 3 *Let Π be an LPOD with tuple of distinct (intensional) predicates $\mathbf{P}$. Then by $\text{MC}\textsc{omp}_{\mathbf{P}}(\Pi)$, we denote the following first-order sentence:*

$$\bigwedge_{r\in\Pi} \forall \mathbf{x}_r(\, \widehat{Body(r)} \rightarrow Head(r)^{FO}\,) \tag{12}$$

$$\wedge \bigwedge_{P\in\mathbf{P}} \forall \mathbf{x}(\, P(\mathbf{x}) \rightarrow \bigvee_{\substack{r\in\Pi,\\ Head(r)_i = P(\mathbf{y}),\\ 1\leq i\leq k}} \exists \mathbf{x}_r(\mathbf{x} = \mathbf{y} \wedge \widehat{Body(r)} \wedge$$

$$\widehat{Pos(r) < P}(\mathbf{x}) \wedge \bigwedge_{1\leq j\leq i-1} \neg Head(r)_j)\,), \tag{13}$$

where:

- *For a rule $r \in \Pi$, $\mathbf{x}_r$ in (12) and (13) is the tuple of distinct variables of r;*
- *We assume that $\mathbf{x}$ in (13) is a tuple of fresh distinct variables not mentioning those from $\mathbf{x}_r$;*
- *For two tuples $\mathbf{x} = x_1 x_2 \ldots x_s$ and $\mathbf{y} = y_1 y_2 \ldots y_s$, $\mathbf{x} = \mathbf{y}$ in (13) denotes the formula $(x_1 = y_1) \wedge (x_2 = y_2) \wedge \ldots \wedge (x_s = y_s)$;*
- *$\widehat{Pos(r) < P}(\mathbf{x})$ in (13) denotes the formula*
$$\bigwedge_{\substack{Q(\mathbf{z})\in Pos(r),\\ Q\in\mathbf{P}}} (\, \leq_{QP}(\mathbf{z}, \mathbf{x}) \wedge \neg \leq_{PQ}(\mathbf{x}, \mathbf{z})\,),$$
where the "$\leq_{QP}$" and "$\leq_{PQ}$" here are comparison predicates.

Then finally, we define

$$\text{OC}_{\mathbf{P}}(\Pi) = \text{MC}\textsc{omp}_{\mathbf{P}}(\Pi) \wedge \text{T}\textsc{rans}_{\mathbf{P}}(\Pi),$$

where $\text{T}\textsc{rans}_{\mathbf{P}}(\Pi)$ denotes the following formula:

$$\bigwedge_{P,Q,R\in\mathbf{P}} \forall \mathbf{xyz}(\, \leq_{PQ}(\mathbf{x}, \mathbf{y}) \wedge$$

$$\leq_{QR}(\mathbf{y}, \mathbf{z}) \rightarrow \leq_{PR}(\mathbf{x}, \mathbf{z})\,), \tag{14}$$

and each of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are tuples of distinct variables.

Now we present the main result of this section as follows.

Theorem 3 *Let Π be an LPOD with tuple of distinct intensional predicates $\mathbf{P}$ and $\mathcal{M}$ a finite $\tau(\Pi)$-structure. Then $\mathcal{M}$ is a stable model of Π iff it can be expanded to a model of $\text{OC}_{\mathbf{P}}(\Pi)$.*

Proof: ($\Rightarrow$) Then $\mathcal{M} \models \widehat{\Pi}$, so that $\mathcal{M}$ is also a model of (12). Therefore, it is now sufficient to show that $\mathcal{M}$ can be expanded to a model of (13) and (14). Indeed, by Theorem 2, we have that $\mathcal{M}^\infty(\Pi) = \mathcal{M}$ as well since $\mathcal{M}$ is a stable model of Π (by assumption). Then based on the progression stages $\mathcal{M}^1(\Pi)$, $\mathcal{M}^2(\Pi)$, …, $\mathcal{M}^k(\Pi)$, $\mathcal{M}^{k+1}(\Pi)$, …, we now construct an expansion $\mathcal{M}'$ of $\mathcal{M}$ to the signature $\tau(\Pi) \cup \{\leq_{PQ} \mid P, Q \in \mathbf{P}\}$. Thus, for $t \geq 0$, set $\Delta^t(\Pi)$ as the $\tau(\Pi)$-structure defined inductively as follows: $\Delta^0(\Pi) = \mathcal{M}^0(\Pi)$ and where for $t \geq 1$, $\Delta^t(\Pi) = \mathcal{M}^t(\Pi) \setminus \mathcal{M}^{t-1}(\Pi)$. Then intuitively speaking, for $t \geq 1$, $\Delta^t(\Pi)$ represents the difference of the intensional relations between the stages t and $t-1$. Then we can prove the following Claim 1.

Claim 1. *If $P(\mathbf{a}) \in \Delta^t(\Pi)$ then for all $t' \neq t$, we have that $P(\mathbf{a}) \notin \Delta^{t'}(\Pi)$.*

Now the expansion $\mathcal{M}'$ of $\mathcal{M}$ is constructed by setting:

$$\leq_{PQ}^{\mathcal{M}'} = \{\mathbf{ab} \mid P(\mathbf{a}) \in \Delta^{t_1}(\Pi), Q(\mathbf{b}) \in \Delta^{t_2}(\Pi), \text{ and } t_1 < t_2\},$$

for each pair of predicates P and Q (can be the same) of $\mathbf{P}$. Now we show that $\mathcal{M}'$ satisfies both (13) and (14). Indeed, let $\mathbf{ab} \in \leq_{PQ}^{\mathcal{M}'}$ and $\mathbf{bc} \in \leq_{QR}^{\mathcal{M}'}$ for some predicates P, Q, and R of $\mathbf{P}$. Then by the definitions of the interpretations $\leq_{PQ}^{\mathcal{M}'}$ and $\leq_{QR}^{\mathcal{M}'}$, we have that: $P(\mathbf{a}) \in \Delta^{t_1}(\Pi)$ and $Q(\mathbf{b}) \in \Delta^{t_2}(\Pi)$ with $t_1 < t_2$; and $Q(\mathbf{b}) \in \Delta^{t'_1}(\Pi)$ and $R(\mathbf{c}) \in \Delta^{t'_2}(\Pi)$ with $t'_1 < t'_2$. Then since by Claim 1, $Q(\mathbf{b})$ can only be in one particular t such that it is in $\Delta^t(\Pi)$, then we have that $t_2 = t'_1$, so that by transitivity, we have $t_1 < t'_2$. Then by the definition of the interpretation $\leq_{PR}^{\mathcal{M}'}$, we have that $\mathbf{ac} \in \leq_{PR}^{\mathcal{M}'}$, so that $\mathcal{M}'$ satisfies (14). Now to show $\mathcal{M}'$ satisfies (13), let $P(\mathbf{a}) \in \mathcal{M}'$ such that $P \in \mathbf{P}$. Then $P(\mathbf{a}) \in \mathcal{M}^\infty(\Pi)$ since $\mathcal{M}$ is a stable model of Π and $\mathcal{M}^\infty(\Pi) = \mathcal{M}$ by Theorem 2. Then for some t, we have that $P(\mathbf{a}) \in \mathcal{M}^t(\Pi)$. Let us assume without loss of generality that $P(\mathbf{a}) \notin \mathcal{M}^{t'}(\Pi)$ for all $0 \leq t' < t$, i.e., t is the first stage that derives $P(\mathbf{a})$. Then by the definition of $\mathcal{M}^t(\Pi)$, there exists some rule $r \in \Pi$ with $Head(r) = \alpha_1 \times \ldots \times \alpha_i \times \ldots \times \alpha_k$ and assignment η such that:

1. $Pos(r)\eta \subseteq \mathcal{M}^{t-1}(\Pi) \subseteq \mathcal{M}$ (and where $\mathcal{M}^{t-1}(\Pi) \subseteq \mathcal{M}$ since $\mathcal{M}^{t-1}(\Pi) \subseteq \mathcal{M}^\infty(\Pi) = \mathcal{M}$) and $Neg(r)\eta \cap \mathcal{M} = \emptyset$;
2. i (for $1 \leq i \leq k$) is the largest i such that $\{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} \cap \mathcal{M} = \emptyset$.

Then since $\mathcal{M}'$ is simply the expansion of $\mathcal{M}$ to include the interpretation of the comparison symbols in $\{\leq_{PQ} \mid P, Q \in \mathbf{P}\}$, we have that

$$\mathcal{M}' \models (\mathbf{a} = \mathbf{y} \wedge \widehat{Body(r)} \wedge \bigwedge_{1\leq j\leq i-1} \neg\alpha_j)\eta.$$

Therefore, it is only left to show that $\mathcal{M}' \models (\widehat{Pos(r) < P}(\mathbf{a}))\eta$ so that

$$\mathcal{M}' \models (\mathbf{a} = \mathbf{y} \wedge \widehat{Body(r)} \wedge \widehat{Pos(r) < P}(\mathbf{a}) \wedge \bigwedge_{1 \leq j \leq i-1} \neg\alpha_j)\eta \tag{15}$$

as well. Indeed, since $\mathcal{M}^t(\Pi)$ is the first stage that derives $P(\mathbf{a})$, then by the definition of $\Delta^t(\Pi)$, we have that $P(\mathbf{a}) \in \Delta^t(\Pi)$. Now let $Q(\mathbf{b}) \in Pos(r)\eta$ such that $Q \in \mathbf{P}$. Then since $Q(\mathbf{b}) \in \mathcal{M}^{t-1}(\Pi)$, there exists the least stage $t' \leq t-1$ such that $Q(\mathbf{b}) \in \mathcal{M}^{t'}(\Pi)$. Then this implies that $Q(\mathbf{b}) \in \Delta^{t'}(\Pi)$, and since $t' \leq t-1 < t$, we further have by the definition of the interpretation $\leq_{QP}^{\mathcal{M}'}$ that $\mathbf{ba} \in \leq_{QP}^{\mathcal{M}'}$. Now we show that $\mathbf{ab} \notin \leq_{PQ}^{\mathcal{M}'}$. Otherwise, assume that $\mathbf{ab} \in \leq_{PQ}^{\mathcal{M}'}$. Then by the definition of $\leq_{PQ}^{\mathcal{M}'}$, there exists t_1 and t_2, where $t_1 < t_2$, such that $P(\mathbf{a}) \in \Delta^{t_1}(\Pi)$ and $Q(\mathbf{b}) \in \Delta^{t_2}(\Pi)$. Then by Claim 1, we have that $t_2 = t'$ since $Q(\mathbf{b})$ is both in $\Delta^{t_2}(\Pi)$ and $\Delta^{t'}(\Pi)$. Then because we already have that $t' < t$ and where $t_1 < t_2 = t' < t$, then we have that $t_1 < t$. Thus, since $P(\mathbf{a})$ is both in $\Delta^{t_1}(\Pi)$ and $\Delta^t(\Pi)$ where $t_1 \neq t$ (since $t_1 < t$), this is a contradiction by Claim 1. Therefore, it follows that $\mathcal{M}' \models \widehat{Pos(r) < P}(\mathbf{a})$ so that (15) holds as well.

($\Leftarrow$) Assume that $\mathcal{M}'$ is the expansion of $\mathcal{M}$ such that $\mathcal{M}' \models \mathrm{OC}_{\mathbf{P}}(\Pi)$. Then there exists a *strict-partial* order $\mathcal{P} = (Dom(\mathcal{P}), <^{\mathcal{P}})$, where

$$Dom(\mathcal{P}) = \{P(\mathbf{a}) \mid \mathbf{a} \in P^{\mathcal{M}} \text{ and } P \in \mathbf{P}\},$$

such that $\mathcal{P}$ is induced by the interpretations of the comparison predicates $\leq_{PQ}$ (for $P, Q \in \mathbf{P}$). Then by the *Order-extension Theorem* (Kaye and Macpherson 1994), there exists a *strict-total* order $\mathcal{T} = (Dom(\mathcal{T}), <^{\mathcal{T}})$, where $Dom(\mathcal{T}) = Dom(\mathcal{P})$, such that for all distinct elements $P(\mathbf{a})$ and $Q(\mathbf{b})$ in $Dom(\mathcal{T})$, either $P(\mathbf{a}) <^{\mathcal{T}} Q(\mathbf{b})$ or $Q(\mathbf{b}) <^{\mathcal{T}} P(\mathbf{a})$ holds. Now, since $\mathcal{M}$ is finite, then $Dom(\mathcal{T})$ will be finite as well, so that a bottom element, denoted $bot(\mathcal{T})$, exists. Then for a $P(\mathbf{a}) \in Dom(\mathcal{T})$, define $\mathcal{T}^{P(\mathbf{a})}$ inductively by:

1. $\mathcal{T}^{bot(\mathcal{T})} = \{bot(\mathcal{T})\}$;
2. $\mathcal{T}^{succ(P(\mathbf{a}))} = \mathcal{T}^{P(\mathbf{a})} \cup \{succ(P(\mathbf{a}))\}$,

where $succ(P(\mathbf{a}))$ denotes the *successor* of $P(\mathbf{a})$ under the ordering $<^{\mathcal{T}}$. Then $\mathcal{T}^{top(\mathcal{T})}$, where $top(\mathcal{T})$ is the greatest element under $<^{\mathcal{T}}$, is simply the collection of all the elements of $Dom(\mathcal{T})$, since $\mathcal{T}$ is a strict-total ordering of $Dom(\mathcal{T})$. We will now show by induction on $P(\mathbf{a})$ that $\mathcal{T}^{P(\mathbf{a})} \subseteq \mathcal{M}^\infty(\Pi)$ for all $P(\mathbf{a}) \in Dom(\mathcal{T})$. For the base case, assume $bot(\mathcal{T}) = P(\mathbf{a})$. Then since $\mathcal{M}'$ satisfies (13), there exists a rule $r \in \Pi$ with $Head(r) = \alpha_1 \times \ldots \times \alpha_k$ such that for some $1 \leq i \leq k$ and some assignment η, we have that

$$\mathcal{M}' \models (\mathbf{a} = \mathbf{y} \wedge \widehat{Body(r)} \wedge \widehat{Pos(r) < P}(\mathbf{a}) \wedge \bigwedge_{1 \leq j \leq i-1} \neg\alpha_j)\eta,$$

and where $\alpha_i = P(\mathbf{y})$. Moreover, since $P(\mathbf{a})$ is the bottom element under the ordering $<^{\mathcal{T}}$, then $Pos(r)$ does not mention atoms of intensional predicates (since the strict-total order $\mathcal{T}$ is induced by the interpretation of the comparison atoms). Therefore, since $P(\mathbf{a}) = \alpha_i\eta \in \mathcal{M}$ (because $Dom(\mathcal{T}) = \{P(\mathbf{a}) \mid \mathbf{a} \in P^{\mathcal{M}} \text{ and } P \in \mathbf{P}\}$), then it is not difficult to verify by Item 2 in (9) that $P(\mathbf{a}) \in \mathcal{M}^1(\Pi) \subseteq \mathcal{M}^\infty(\Pi)$. Now assume $\mathcal{T}^{Q(\mathbf{b})} \subseteq \mathcal{M}^\infty(\Pi)$ and we will show that $\mathcal{T}^{succ(Q(\mathbf{b}))} \subseteq \mathcal{M}^\infty(\Pi)$ as well. Thus, for convenience, assume that $succ(Q(\mathbf{b})) = P(\mathbf{a})$. Now, since $\mathcal{M}'$ satisfies (13), then there exists some rule $r \in \Pi$ with $Head(r) = \alpha_1 \times \ldots \times \alpha_k$ and assignment η such that

$$\mathcal{M}' \models (\mathbf{a} = \mathbf{y} \wedge \widehat{Body(r)} \wedge \widehat{Pos(r) < P}(\mathbf{a}) \wedge \bigwedge_{1 \leq j \leq i-1} \neg\alpha_j)\eta,$$

where $\alpha_i = P(\mathbf{y})$ (the "ordered support" for $P(\mathbf{a})$). Then since $\mathcal{M}' \models (\widehat{Pos(r) < P}(\mathbf{a}))\eta$ and $\mathcal{T}$ is the total-order extension of the strict-partial order induced by the interpretation of the comparison atoms, we have that $Pos(r)\eta \subseteq \mathcal{T}^{Q(\mathbf{b})}$ since $Q(\mathbf{b}) <^{\mathcal{T}} P(\mathbf{a})$. Moreover, since $\mathcal{M}' \models \widehat{Body(r)}\eta$, then $Neg(r)\eta \cap \mathcal{M} = \emptyset$ as well. Therefore, since $\mathcal{T}^{Q(\mathbf{b})} \subseteq \mathcal{M}^t(\Pi)$ for some t (since $\mathcal{T}^{Q(\mathbf{b})} \subseteq \mathcal{M}^\infty(\Pi)$ by the inductive hypothesis), then we have that both $Pos(r)\eta \subseteq \mathcal{M}^t(\Pi)$ and $Neg(r)\eta \cap \mathcal{M} = \emptyset$ holds. In addition, since we also have from (13) that $\mathcal{M} \models (\neg\alpha_1 \times \ldots \times \neg\alpha_{i-1})\eta$ and where $\alpha_i\eta = P(\mathbf{a}) \in \mathcal{M}$, then it follows from (9) that $P(\mathbf{a}) \in \mathcal{M}^{t+1}(\Pi)$. Hence, since we have shown that $\mathcal{T}^{top(\mathcal{T})} \subseteq \mathcal{M}^\infty(\Pi)$, then since $\mathcal{M}^0(\Pi) \subseteq \mathcal{M}^\infty(\Pi)$, it follows that $(\mathcal{T}^{top(\mathcal{T})} \cup \mathcal{M}^0(\Pi)) = \mathcal{M} \subseteq \mathcal{M}^\infty(\Pi)$. Therefore, to show that $\mathcal{M} = \mathcal{M}^\infty(\Pi)$, it is left for us to show that $\mathcal{M}^\infty(\Pi) \subseteq \mathcal{M}$. We now show this fact by induction on t. For the base case, we clearly have that $\mathcal{M}^0(\Pi) \subseteq \mathcal{M}$ since for $\mathcal{M}^0(\Pi)$, all the interpretation of the comparison predicates are set to empty. Now let us assume that for $0 \leq t' \leq t$, we have that $\mathcal{M}^{t'}(\Pi) \subseteq \mathcal{M}$ holds and we will now show that $\mathcal{M}^{t+1}(\Pi) \subseteq \mathcal{M}$ holds as well. Indeed, let $P(\mathbf{a}) \in \mathcal{M}^{t+1}(\Pi) \setminus \mathcal{M}^t(\Pi)$. Then by the definition of $\mathcal{M}^{t+1}(\Pi) \setminus \mathcal{M}^t(\Pi)$, there exists a rules $r \in \Pi$ with $Head(r) = \alpha_1 \times \ldots \times \alpha_i \times \ldots \times \alpha_k$ and assignment η such that

1. $P(\mathbf{a}) = \alpha_i\eta$;
2. $Pos(r)\eta \subseteq \mathcal{M}^t(\Pi)$ and $Neg(r)\eta \cap \mathcal{M} = \emptyset$;
3. i (for $1 \leq i \leq k$) is the largest i such that $\{\alpha_1\eta, \ldots, \alpha_{i-1}\eta\} \cap \mathcal{M} = \emptyset$.

Then since $\mathcal{M}^t(\Pi) \subseteq \mathcal{M}$ and $\mathcal{M} \models \widehat{\Pi}$ (since $\mathcal{M}$ satisfies (12)), then it follows that $P(\mathbf{a}) \in \mathcal{M}$. Therefore, since we had shown that $\mathcal{M}^\infty(\Pi) = \mathcal{M}$, then we have by Theorem 2 that $\mathcal{M}$ is a stable model of Π. $\square$

Theorem 3 has an important practical value towards an LPOD solver development. Basically, for a given LPOD, to compute its stable models, we can firstly translate this program into its corresponding ordered completion $\mathrm{OC}_{\mathbf{P}}(\Pi)$, then by taking extensional databases as input to ground $\mathrm{OC}_{\mathbf{P}}(\Pi)$ to a propositional formula, and finally compute its classical models by calling an SAT solver (e.g., SMT). Some advantages of this approach over other ASP solvers for normal logic programs have been demonstrated in (Asuncion et al. 2012).

Proposition 2 *Let Π be an LPOD. Then $\mathrm{OC}_{\mathbf{P}}(\Pi)$ can be computed in time $O(|\Pi| \cdot N \cdot H \cdot |\mathbf{P}| + |\mathbf{P}|^3)$, where N and H are the maximum length of the rules and the maximum length of the ordered disjunctive heads in the rules of Π, respectively.*

Complexity and Expressiveness

Now we investigate the complexity and expressiveness issues of LPODs. For the propositional case, we can prove that the stable model existence problem for LPODs is NP-complete[1]. In the following, we focus on the first-order case. Let Π_1 and Π_2 be two programs and $\tau(\Pi_1) \subseteq \tau(\Pi_2)$. We say that Π_1 and Π_2 are *equivalent under* $\tau(\Pi_1)$ if Π_1 and Π_2 have exactly the same stable models by restricting each Π_2's stable model $\mathcal{M}$ to $\mathcal{M}\upharpoonright_{\tau(\Pi_1)}$.

We say that a program is *positive* if negation only occurs on atoms of extensional predicates. A normal logic program is *local variable free* if all the variables in the bodies of rules are also mentioned in their corresponding heads.

Proposition 3 *Under finite structures, every local variable free normal logic program Π can be translated to a positive LPOD with auxiliary predicates such that these two programs are equivalent under $\tau(\Pi)$.*

Proof: Let Π^{NORM} be a local variable free FO normal program with intensional predicate symbols $\mathcal{P}_{int}(\Pi^{\text{NORM}})$. Then for each predicate $P \in \mathcal{P}_{int}(\Pi^{\text{NORM}})$, let us introduce a new predicate $\overline{P}$ of the same arity as P. Roughly speaking, $\overline{P}$ will encode the negative extents of P. Now we are ready to define the positive LPOD Π^{LPOD}.

Denote by $(\Pi^{\text{NORM}})^{pos}$ the following set of rules:

$$\begin{aligned}\{\alpha \ \leftarrow \ \beta_1, \ldots, \beta_l, \overline{\gamma_1}, \ldots, \overline{\gamma_m} \mid \alpha \ \leftarrow \ \beta_1, \ldots, \beta_l,&\\ \mathsf{not}\,\gamma_1, \ldots, \mathsf{not}\,\gamma_m \in \Pi^{\text{NORM}},&\\ \text{for } 1 \leq i \leq m, \text{if } \gamma_i = P(\mathbf{t}) \text{ and } P \in \mathcal{P}_{int}(\Pi^{\text{NORM}})&\\ \text{then } \overline{\gamma_i} = \overline{P}(\mathbf{t}), \text{ otherwise } \overline{\gamma_i} = \gamma_i\, \}.& \quad (16)\end{aligned}$$

Then clearly, $(\Pi^{\text{NORM}})^{pos}$ is a positive LPOD of signature $\tau(\Pi) \cup \{\overline{P} \mid P \in \mathcal{P}_{int}(\Pi^{\text{NORM}})\}$.

Let S be a set of atoms occurring in Π, we denote by $\mathcal{A}_{ext}(S)$ and $\mathcal{A}_{int}(S)$ the sets of extensional and intensional atoms in S, respectively. Now let us define another program Π' as follows:

$$\begin{aligned}\{Bd_{r_1}(\mathbf{x}) \times \ldots \times Bd_{r_k}(\mathbf{x}) \ \leftarrow \ P(\mathbf{x}) \mid P \in \mathcal{P}_{int}(\Pi^{\text{NORM}}),&\\ P \text{ has defining rules } r_1, \ldots, r_k \in \Pi^{\text{NORM}}\}&\\ &(17)\\ \cup \{\beta \ \leftarrow \ Bd_r(\mathbf{x}), \bot \ \leftarrow \ \mathsf{not}\,\beta', Bd_r(\mathbf{x}), \overline{\gamma} \ \leftarrow \ Bd_r(\mathbf{x}),&\\ \bot \ \leftarrow \ \gamma', Bd_r(\mathbf{x}) \mid r \in \Pi^{\text{NORM}}, \beta \in \mathcal{A}_{int}(Pos(r)),&\\ \beta' \in \mathcal{A}_{ext}(Pos(r)), \gamma \in \mathcal{A}_{int}(Neg(r)), \gamma' \in \mathcal{A}_{ext}(Neg(r))\}&\\ &(18)\\ \cup \{<_{QP} (\mathbf{z}, \mathbf{x}) \ \leftarrow \ Bd_r(\mathbf{x}) \mid r \in \Pi^{\text{NORM}}, Head(r) = P(\mathbf{x}),&\\ Q(\mathbf{z}) \in \mathcal{A}_{int}(Pos(r))\,\}&\\ &(19)\\ \cup \{<_{PR} (\mathbf{x}, \mathbf{z}) \ \leftarrow <_{PQ} (\mathbf{x}, \mathbf{y}), <_{QR} (\mathbf{y}, \mathbf{z}) \mid&\\ P, Q, R \in \mathcal{P}_{int}(\Pi^{\text{NORM}})\}&\\ &(20)\\ \cup \{\bot \ \leftarrow <_{PQ} (\mathbf{x}, \mathbf{y}), <_{QP} (\mathbf{y}, \mathbf{x}) \mid P, Q \in \mathcal{P}_{int}(\Pi^{\text{NORM}})\}.&\\ &(21)\end{aligned}$$

Lastly, further define one more program Π'' as follows:

$$\begin{aligned}\{\ \bot \ \leftarrow \ P(\mathbf{x}), \overline{P}(\mathbf{x}),& \quad (22)\\ P(\mathbf{x}) \times \overline{P}(\mathbf{x}) \ \leftarrow \ \mid P \in \mathcal{P}_{int}(\Pi^{\text{NORM}})\}.& \quad (23)\end{aligned}$$

Then we can now set Π^{LPOD} as the union $(\Pi^{\text{NORM}})^{pos} \cup \Pi' \cup \Pi''$. Clearly, Π^{LPOD} is a positive LPOD.

Intuitively, we have that (16)-(21) encode the ordered completion of Π^{NORM} within the LPOD Π^{LPOD} itself. In particular, we note (18), which encodes the "supporting" atoms for the "body atoms" $Bd_r(\mathbf{x})$ (which corresponds to the body of the rule r), but in such a way that we avoid defining the extensional predicates in a head of a rule. On the other hand, (22) enforces that $(P(\mathbf{x}) \rightarrow \neg\overline{P}(\mathbf{x})) \wedge (\overline{P}(\mathbf{x}) \rightarrow \neg P(\mathbf{x}))$ while (23) enforces that $(\neg P(\mathbf{x}) \rightarrow \overline{P}(\mathbf{x})) \wedge (\neg\overline{P}(\mathbf{x}) \rightarrow P(\mathbf{x}))$, i.e., the "necessary" counterpart of (22) with respect to the extents of $\overline{P}$ being the symmetric negation of those in P. $\Box$

Theorem 4 *Let Π be an LPOD and $\mathcal{M}$ a finite $\tau_{ext}(\Pi)$-structure, i.e., an extensional or input database structure. Then the problem of determining if $\mathcal{M}$ can be expanded to a stable model of Π is NP-complete. This result remains true even for positive LPODs.*

Proof: (*Membership*) Given our reduction of Π to the FO formula $\mathrm{OC}_{\mathbf{P}}[\Pi]$ via Theorem 3 (i.e., the reduction of Π to its ordered completion), we have that determining if $\mathcal{M}$ can be expanded to a model of $\mathrm{OC}_{\mathbf{P}}[\Pi]$ is the model expansion problem, which is in NP (model expansion is in fact NP-complete).

(*Hardness*) Consider the 3-color program Π_{3color} as fol-

[1] In our full version, we provide a polynomial translation from normal logic programs to LPODs. This result can be also obtained using a different translation proposed in (Nieuwenborgh and Vermeir 2003).

lows:

$$
\begin{aligned}
C_1(x) &\leftarrow \mathsf{not}\, C_2(x), \mathsf{not}\, C_3(x), && (24)\\
C_2(x) &\leftarrow \mathsf{not}\, C_1(x), \mathsf{not}\, C_3(x), && (25)\\
C_3(x) &\leftarrow \mathsf{not}\, C_1(x), \mathsf{not}\, C_2(x), && (26)\\
&\leftarrow E(x,y), C_1(x), C_1(y), && (27)\\
&\leftarrow E(x,y), C_2(x), C_2(y), && (28)\\
&\leftarrow E(x,y), C_3(x), C_3(y). && (29)\\
& && (30)
\end{aligned}
$$

Given a graph structure $\mathcal{G} = (Dom(\mathcal{G}), V^{\mathcal{G}}, E^{\mathcal{G}})$ such that $V^{\mathcal{G}} = Dom(\mathcal{G})$ (i.e., the vertices of $\mathcal{G}$), $\mathcal{G}$ has a corresponding 3-coloring iff Π_{3color} has a stable model. It is well known that the problem of 3-coloring is NP-complete. On the other hand, we have by Proposition 3 that Π_{3color} can be reduced to a positive LPOD since Π_{3color} is a local variable free normal program. □

A LPOD is called *almost positive* if each negated intensional atoms in the program only occurs in the bodies of some constraints. The following theorem states that almost positive LPODs precisely capture the full class of normal logic programs. That is, ordered disjunction and constraints are sufficient enough to represent negation as failure.

Theorem 5 *Every normal logic program* Π *can be translated to an almost positive LPODs with auxiliary predicates such that these two programs are equivalent under* $\tau(\Pi)$.

Proof: Let Π^{NORM} be an arbitrary normal logic program. We define a reduction from Π^{NORM} to an almost positive LPOD Π^{LPOD} by defining Π^{LPOD} as the following set of rules:

$$
\begin{aligned}
&\{\alpha \leftarrow \beta_1, \ldots, \beta_l, \overline{\gamma_1}, \ldots, \overline{\gamma_m} \mid \alpha \leftarrow \beta_1, \ldots, \beta_l,\\
&\qquad \mathsf{not}\,\gamma_1, \ldots, \mathsf{not}\,\gamma_m \in \Pi^{\text{NORM}},\\
&\qquad \text{for } 1 \le i \le m, \text{if } \gamma_i = P(\mathbf{t}) \text{ and } P \in \mathcal{P}_{int}(\Pi^{\text{NORM}})\\
&\qquad \text{then } \overline{\gamma_i} = \overline{P}(\mathbf{t}),\ \text{otherwise } \overline{\gamma_i} = \gamma_i \} \quad (31)\\
&\cup \{\bot \leftarrow P(\mathbf{x}), \overline{P}(\mathbf{x}), \quad (32)\\
&\bot \leftarrow \mathsf{not}\, P(\mathbf{x}), \mathsf{not}\, \overline{P}(\mathbf{x}), \quad (33)\\
&\bot \leftarrow P'(\mathbf{x}), \overline{P}(\mathbf{x}), \quad (34)\\
&P'(\mathbf{x}) \times \overline{P}(\mathbf{x}) \leftarrow \mid P \in \mathcal{P}_{int}(\Pi^{\text{NORM}})\}, \quad (35)
\end{aligned}
$$

where for each $P \in \mathcal{P}_{int}(\Pi)$, we introduce two new predicate symbols $\overline{P}$ and P'. Then Π^{LPOD} is clearly an almost positive LPOD since negation only occurs in the constraints in (33).

Intuitively, the key here is the combination of (34) and (35), which enforces $(\neg P'(\mathbf{x}) \rightarrow \overline{P}(\mathbf{x})) \wedge (\neg \overline{P}(\mathbf{x}) \rightarrow P'(\mathbf{x}))$. This has the effect of "fixing" $\overline{P}(\mathbf{x})^*$ since the "minimization" of $\overline{P}^*$ implies the "expansion" of $(P')^*$ (and vice versa). This simulates the condition that $\overline{P}(\mathbf{x})$ (which encodes "not $P(\mathbf{x})$") will be fixed in the minimization of $\overline{P}(\mathbf{x})^*$ since we also have $(\neg P'(\mathbf{x})^* \rightarrow \overline{P}(\mathbf{x})^*) \wedge (\neg \overline{P}(\mathbf{x})^* \rightarrow P'(\mathbf{x})^*)$. On the other hand, (32) and (33) again enforces $(P(\mathbf{x}) \rightarrow \neg \overline{P}(\mathbf{x})) \wedge (\overline{P}(\mathbf{x}) \rightarrow \neg P(\mathbf{x}))$ and $(\neg P(\mathbf{x}) \rightarrow \overline{P}(\mathbf{x})) \wedge (\neg \overline{P}(\mathbf{x}) \rightarrow P(\mathbf{x}))$, respectively. □

Preferred Stable Model Semantics

As indicated by Brewka (2002; 2006), the stable models based on split programs are usually not sufficient to capture the preference semantics for a given LPOD, while a certain preference relation based on the degree of satisfaction has to be imposed on the stable models. Now we show how such preferred stable model semantics can be defined for our FO LPODs.

Definition 4 *Let r be a rule in some LPOD Π, $\mathcal{M}$ a $\tau(\Pi)$-structure, and η an assignment from the distinct tuple of variables $\mathbf{x}$ of r to $Dom(\mathcal{M})$. The* satisfaction degree *of r under the structure $\mathcal{M}$ and assignment η, denoted by $Deg_{(\mathcal{M},\eta)}(r)$, is defined as follows:*

$$
Deg_{(\mathcal{M},\eta)}(r) := \begin{cases} 1 & \text{if } \mathcal{M} \not\models \widehat{Body(r)}\eta, \text{ otherwise} \\ i & \text{if } \mathcal{M} \models (\widehat{Body(r)} \wedge \neg\alpha_1 \wedge \neg\alpha_2 \wedge \\ & \quad \ldots \wedge \neg\alpha_{i-1} \wedge \alpha_i)\eta. \end{cases}
$$

Similarly to the propositional case, if $(\mathcal{M}, \eta)$ does not satisfy the rule r's body, then the default satisfaction degree is 1. Otherwise, the satisfaction degree is the "minimal" (i.e., most preferred) of the atoms in $\alpha_1 \times \ldots \times \alpha_k$ that $(\mathcal{M}, \mathbf{a})$ satisfies. Again as in the propositional case, this encodes that we are paying for a penalty for the "least preferred" atoms in $\alpha_1 \times \ldots \times \alpha_k$ being satisfied in the sense that, the smaller the satisfaction degree, then the better it is.

Definition 5 *Let Π be an LPOD and $\mathcal{M}_1$ and $\mathcal{M}_2$ be two stable models of Π such that $\mathcal{M}_1 \upharpoonright_{\tau_{ext}(\Pi)} = \mathcal{M}_2 \upharpoonright_{\tau_{ext}(\Pi)}$. Then we say that $\mathcal{M}_1$ is* Pareto-preferred *to $\mathcal{M}_2$, denoted by $\mathcal{M}_1 > \mathcal{M}_2$, if the following two conditions hold:*

1. *For each rule $r \in \Pi$ and assignment η, we have that $Deg_{(\mathcal{M}_1,\eta)}(r) \le Deg_{(\mathcal{M}_2,\eta)}(r)$;*
2. *There is a rule $r' \in \Pi$ and an assignment η' such that $Deg_{(\mathcal{M}_1,\eta')}(r') < Deg_{(\mathcal{M}_2,\eta')}(r')$.*

Example 2 *Assume Π to be the following LPOD with the single rule r:*

$$
r:\ P(x) \times Q(x) \times R(x) \leftarrow S(x) \qquad (36)
$$

such that S is the only extensional predicate. Now let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two $\tau(\Pi)$-structures such that:

$$
\begin{aligned}
\mathcal{M}_1 = (\{a,b\}, S^{\mathcal{M}_1} = \{a,b\}, P^{\mathcal{M}_1} = \{a\},\\
Q^{\mathcal{M}_1} = \{b\}, R^{\mathcal{M}_1} = \emptyset);\\
\mathcal{M}_2 = (\{a,b\}, S^{\mathcal{M}_2} = \{a,b\}, P^{\mathcal{M}_2} = \emptyset,\\
Q^{\mathcal{M}_2} = \{b\}, R^{\mathcal{M}_2} = \{a\}).
\end{aligned}
$$

Then we have that $\mathcal{M}_1 \upharpoonright_{\tau_{ext}(\Pi)} = \mathcal{M}_2 \upharpoonright_{\tau_{ext}(\Pi)}$ and where the grounding of Π under the domain $\{a,b\}$ will be the propositional program

$$
\begin{aligned}
P(a) \times Q(a) \times R(a) &\leftarrow S(a) && (37)\\
P(b) \times Q(b) \times R(b) &\leftarrow S(b). && (38)
\end{aligned}
$$

Then based on the grounded Π, it can be seen that both $\mathcal{M}_1$ and $\mathcal{M}_2$ are stable models of Π. In addition, since for all the assignments $\eta_1 : x \longrightarrow a$ and $\eta_2 : x \longrightarrow b$ of the variable x to $\{a,b\}$, we have that $Deg_{(\mathcal{M}_1,\eta_1)}(r) < Deg_{(\mathcal{M}_2,\eta_1)}(r)$ and $Deg_{(\mathcal{M}_1,\eta_2)}(r) = Deg_{(\mathcal{M}_2,\eta_2)}(r)$, then $\mathcal{M}_1 > \mathcal{M}_2$, i.e., $\mathcal{M}_1$ is Pareto-preferred to $\mathcal{M}_2$. □

Based on the notion of a Pareto-preferred stable models, we can now define the notion of a preferred stable model of an LPOD.

Definition 6 *Let Π be an LPOD and $\mathcal{M}$ its stable model. $\mathcal{M}$ is called a* preferred stable model *of Π if there does not exists another structure $\mathcal{M}'$ for which $\mathcal{M}' > \mathcal{M}$.*

Logic Formalization of Preferred Stable Models

Now we provide a logical characterization of the preferred stable models. For this purpose, let us assume the tuple of distinct intensional predicates in our signature to be $\mathbf{P} = P_1 \dots P_n$. We use $\mathbf{P'}$ to denote the tuple of distinct predicates $P'_1 \dots P'_s$ such that each P'_i $(1 \le i \le s)$ is a new predicate symbol.

Consider a rule r of an LPOD with the form (3) and whose atoms are predicates from $\mathbf{P}$. We use $Deg(r)^{\mathbf{P'}} \le Deg(r)^{\mathbf{P}}$ to denote the following formula:

$$\bigwedge_{1 \le i \le k} \forall \mathbf{x}_r \big((\widehat{Body(r)} \wedge \bigwedge_{1 \le j \le i} \neg \alpha_j)[\mathbf{P}/\mathbf{P'}] \rightarrow (\widehat{Body(r)} \wedge \bigwedge_{1 \le j \le i} \neg \alpha_j) \big), \qquad (39)$$

where $(\widehat{Body(r)} \wedge \bigwedge_{1 \le j \le i} \neg \alpha_j)[\mathbf{P}/\mathbf{P'}]$ denotes the formula obtained from $(\widehat{Body(r)} \wedge \bigwedge_{1 \le j \le i} \neg \alpha_j)$ by replacing the occurrences of predicates from $\mathbf{P}$ by those corresponding ones in $\mathbf{P'}$.

Roughly speaking, $Deg(r)^{\mathbf{P'}} \le Deg(r)^{\mathbf{P}}$ is a formula that encodes all instances of r under the interpretation of the predicates from $\mathbf{P'}$ is at an equal or lower satisfaction degree than those ones from $\mathbf{P}$. Indeed, (39) encodes that the degree of r under $\mathbf{P'}$ is less than or equal than those under $\mathbf{P}$. As we will see later on, the purpose of the predicates $\mathbf{P'}$ that matches $\mathbf{P}$ is it will be used as second-order variables.

We now extend this notion to a whole LPOD Π so that by $Deg(\Pi)^{\mathbf{P'}} \le Deg(\Pi)^{\mathbf{P}}$, we denote the conjunctions $\bigwedge_{r \in \Pi} (Deg(r)^{\mathbf{P'}} \le Deg(r)^{\mathbf{P}})$, and that we further denote by $Deg(\Pi)^{\mathbf{P'}} < Deg(\Pi)^{\mathbf{P}}$ as the conjunction

$$(Deg(\Pi)^{\mathbf{P'}} \le Deg(\Pi)^{\mathbf{P}}) \wedge \neg (Deg(\Pi)^{\mathbf{P}} \le Deg(\Pi)^{\mathbf{P'}}). \qquad (40)$$

Intuitively, $Deg(\Pi)^{\mathbf{P'}} < Deg(\Pi)^{\mathbf{P}}$ is a formula which encodes that the interpretation $\mathbf{P'}$ satisfies Π in a more optimal manner (i.e., in the sense of Definition 5) than that of Π under $\mathbf{P}$.

The following theorem now provides a classical logic characterization of preferred stable models of LPODs via a second-order sentence.

Theorem 6 *Let Π be an LPOD with tuple of intensional predicates $\mathbf{P} = P_1 \dots P_n$ and $\mathbf{P'} = P'_1 \dots P'_n$ the fresh tuple of predicates matching $\mathbf{P}$, and $\mathcal{M}$ a $\tau(\Pi)$-structure. Then $\mathcal{M}$ is a preferred stable model of Π if and only if it satisfies the following second-order sentence:*

$$\mathrm{SM}_{\mathbf{P}}(\Pi) \wedge \neg \exists \mathbf{P'} \big(\mathrm{SM}_{\mathbf{P}}(\Pi)[\mathbf{P}/\mathbf{P'}] \wedge Deg(\Pi)^{\mathbf{P'}} < Deg(\Pi)^{\mathbf{P}} \big), \qquad (41)$$

where $\mathrm{SM}_{\mathbf{P}}(\Pi)[\mathbf{P}/\mathbf{P'}]$ denotes the formula obtained from $\mathrm{SM}_{\mathbf{P}}(\Pi)$ by simultaneously replacing all predicates from $\mathbf{P}$ by those corresponding ones from $\mathbf{P'}$.

It can be showed that (41) also correctly represents Brewka's preferred stable model semantics when we restrict it to the case of propositional LPODs.

Corollary 1 *On finite structures, $\mathcal{M}$ is a preferred stable model of Π if and only if it satisfies the following second-order sentence:*

$$\mathrm{OC}_{\mathbf{P}}(\Pi) \wedge \neg \exists \mathbf{P'T'} \big(\mathrm{OC}_{\mathbf{P}}(\Pi)[\mathbf{PT}/\mathbf{P'T'}] \wedge Deg(\Pi)^{\mathbf{P'T'}} < Deg(\Pi)^{\mathbf{QT}} \big),$$

where $\mathrm{OC}_{\mathbf{P}}(\Pi)$ is the ordered completion of Π (a first-order sentence) as defined in Definition 3, and we assume that $\mathbf{T}$ denotes the tuple of distinct comparison predicates and such that $\mathbf{T'}$ are the new predicate symbols that matches $\mathbf{T}$.

Theorem 7 *Let Π be an LPOD and $\mathcal{M}$ a $\tau(\Pi)$-structure. Then determining if $\mathcal{M}$ is a preferred stable model of Π is co-NP-complete.*

Proof: (*Sketch*) Due to a space limit, here we only present a sketch proof for the hardness. A full proof is given in our full paper. For a given normal logic program Π^{NORM} and an extensional structure $\mathcal{M}_{ext}$ of it, we reduce the problem of determining if $\mathcal{M}_{ext}$ cannot be expanded to a stable model of Π^{NORM} to the problem of determining if a stable model $\mathcal{M}^*$ of an LPOD Π^* is a preferred stable model. Let Π^{NORM} be an arbitrary normal logic program. We specify a program $\Pi^{\text{NORM}}_{\{P(a)\}}$ as follows:

$$\Pi^{\text{NORM}}_{\{P(a)\}} := \{ \alpha \leftarrow P(a), \beta_1, \dots, \beta_l, \mathsf{not}\, \gamma_1, \dots, \mathsf{not}\, \gamma_m \mid \alpha \leftarrow \beta_1, \dots, \beta_l, \mathsf{not}\, \gamma_1, \dots, \mathsf{not}\, \gamma_m \in \Pi^{\text{NORM}} \},$$

where we assume that $P(a)$ is an atom with P a new predicate symbol and a a new constant symbol. Then $\Pi^{\text{NORM}}_{\{P(a)\}}$ is simply the program obtained from Π^{NORM} by adding the atom $P(a)$ (i.e., propositional atom) into each of the positive bodies of the rules of Π^{NORM}. Now by Π^{LPOD}, define the following set of rules such that:

$$\Pi^{\text{LPOD}} := \{ \begin{array}{rcll} P(a) & \leftarrow & \mathsf{not}\, Q(a), & (42) \\ Q(a) & \leftarrow & \mathsf{not}\, P(a), & (43) \\ R_1(a) \times R_2(a) & \leftarrow & Q(a), & (44) \\ & \leftarrow & R_1(a) \quad \}, & (45) \end{array}$$

where we assume here that Q, R_1, and R_2 are new predicate symbols as well. Intuitively, rules (42) and (43) are the defining (and the only defining) rules for $P(a)$ and $Q(a)$, respectively, such that at least one, and not both, must be in any stable model of Π^{LPOD}. In addition, the rule (44) is an LPOD rule with $R_1(a)$ taking preference over $R_2(a)$. Finally, the constraint (45) simply enforces to choose $R_2(a)$ (which is less preferred over $R_1(a)$) over $R_1(a)$ so that the satisfaction degree of (44) will only be as low as 1 only if $Q(a)$ is false (since $Q(a)$ false makes the body of (44) to be false as well). Thus, if we want to get the most preferred stable model of Π^{LPOD}, then it must imply $Q(a)$ to be false. Finally, we can prove the following claim, which leads to the completion of this proof.

Claim: *There exists a stable model $\mathcal{M}^*$ of program $\Pi^{\text{NORM}}_{\{P(a)\}} \cup \Pi^{\text{LPOD}}$, such that $\mathcal{M}^*$ is a preferred stable model of $\Pi^{\text{NORM}}_{\{P(a)\}} \cup \Pi^{\text{LPOD}}$ iff $\mathcal{M}_{ext}$ cannot be expanded to a stable model of Π^{NORM}.*

□

Proposition 4 *Let Π be an LPOD and $\mathcal{M}$ a corresponding extensional structure. The problem of determining if $\mathcal{M}$ can be expanded to a preferred stable model of Π is NP-complete.*

Conclusions

Preference plays an important role in commonsense reasoning, while developing an effective yet expressive mechanism of handling preference in Answer Set Programming is technically challenging, e.g., (Brewka, Truszczynski, and Woltran 2010; Delgrande, Schaub, and Tompits 2004). In this paper we have developed a formulation of fist-order LPODs which may be viewed as a natural generalization of Brewka's propositional LPODs. The relevant semantics and expressiveness results also provide important insights for us to understand this type of first-order LPODs.

The proposed both the second-order based stable model semantics and the progression semantics capture two important aspects of LPODs: their relationship to classical logic as well as the underlying reasoning feature involving ordered disjunction on the first-order level.

The translation from LPODs to first-order sentences extends the previous work of Asuncion *et al*' ordered completion and we argue that this translation will be useful in developing an effective solver for first-order LPODs, as demonstrated in (Asuncion et al. 2012) for normal logic programs. The complexity and expressiveness results confirm that LPODs remain in NP and the hardness holds even for positive LPODs, while almost positive LPODs capture the full class of normal logic programs. Our logic characterization of preferred stable model semantics reveals that we can eventually use a classical second-order sentence to precisely represent the preference relation among stable models so that such first-order preference semantics may be formalized in a unified way as LPOD first-order stable model semantics.

For future work, we are considering to develop a first-order LPOD solver based on the translation proposed in section 4. Another interesting work is to extend our semantics for first-order LPODs by allowing both ordered and unordered disjunction in the heads of rules, as discussed in (Cabalar 2011), and explore their logical properties and possible applications in practical domains.

Acknowledgement

This publication was made possible by the support of an NPRP grant (NPRP 09-079-1-013) from the Qatar National Research Fund (QNRF). The statements made herein are solely the responsibility of the authors.

References

Asuncion, V.; Lin, F.; Zhang, Y.; and Zhou, Y. 2012. Ordered completion for first-order logic programs on finite structures. *Artificial Intelligence* 177-179:1–24.

Asuncion, V.; Zhang, Y.; and Zhou, Y. 2013. Preferred first-order answer set programs. *ACM Transactions on Computational Logic* to appear.

Babb, J., and Lee, J. 2012. Module theorem for the general theory of stable models. *Theory and Practice of Llogic Programming* 12(4-5):719–735.

Brewka, G.; Benferhat, S.; and Berre, D. L. 2002. Qualitive choice logic. In *Proceedings of KR-2002*, 158–169.

Brewka, G.; Truszczynski, M.; and Woltran, S. 2010. Representing preferences among sets. In *Proceedings of AAAI-2010*, 273–278.

Brewka, G. 2002. Logic programming with ordered disjunction. In *Proceedings of AAAI-2002*, 100–105.

Brewka, G. 2006. Answer sets and qualitative optimization. *Logic Journal of the IGPL* 14(3):413–433.

Cabalar, P. 2011. A logical characterisation of ordered disjunction. *AI Communications* 24(2):165–175.

Confalonieri, R.; Nieves, J.; Osorio, M.; and Vazquez-Salceda, J. 2010. Possibilistic semantics for logic programs with ordered disjunction. In *Proceedings of FoIKS-2010*, 133–152.

Delgrande, J.; Schaub, T.; and Tompits, H. 2004. A classification and survey of preference handling approaches in nonmonotonic reasoning. *Computational Intelligence* 20:308–334.

Ferraris, P.; Lee, J.; and Lifschitz, V. 2011. Stable models and circumscription. *Artificial Intelligence* 175(1):236–263.

Karger, P.; Lopes, N.; Olmedilla, D.; and Polleres, A. 2008. Towards logic programs with ordered and unordered disjunction. In *Proceedings of Workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP 2008), 24th International Conference on Logic Programming (ICLP 2008)*.

Kaye, R., and Macpherson, D. 1994. *Automorphism of First-order Structures*. Clarendon Press.

Lee, J., and Meng, Y. 2011. First-order stable model semantics and first-order loop formulas. *Journal of Artificial Intelligence Research (JAIR)* 42:125–180.

Lin, F., and Zhou, Y. 2011. From answer set logic programming to circumscription via logic of gk. *Artificial Intelligence* 175(1):264–277.

Nieuwenborgh, D. V., and Vermeir, D. 2003. Order and negation as failure. In *Proceedings of ICLP-2003*, 194–208.

Zhang, Y., and Zhou, Y. 2010. On the progression semantics and boundedness of answer set programs. In *Proceedings of KR-2010*, 518–527.