

## Preferred First-Order Answer Set Programs

VERNON ASUNCION, YAN ZHANG, and YI ZHOU, University of Western Sydney, Australia

In this article, we consider the issue of how first-order answer set programs can be extended for handling preference reasoning. To this end, we propose a progression-based preference semantics for first-order answer set programs while explicit preference relations are presented. We study essential properties of the proposed preferred answer set semantics. To understand the expressiveness of preferred first-order answer set programming, we further specify a second-order logic representation which precisely characterizes the progression-based preference semantics.

Categories and Subject Descriptors: F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic; I.2.3 [Artificial Intelligence]: Deduction and Theorem Proving—*logic programming and nonmonotonic reasoning*; I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods—*predicate logic, representation languages*

General Terms: Languages, Theory

Additional Key Words and Phrases: Answer set programming, first-order nonmonotonic reasoning, preference, second-order logic

### ACM Reference Format:

Vernon Asuncion, Yan Zhang, and Yi Zhou. 2014. Preferred first-order answer set programs. *ACM Trans. Comput. Logic* 15, 2, Article 11 (April 2014), 42 pages.  
DOI: <http://dx.doi.org/10.1145/2579817>

## 1. INTRODUCTION

Preferences play an important role in knowledge representation and reasoning. In the past decade, a number of approaches for handling preferences have been developed in various nonmonotonic reasoning formalism (e.g., see a survey by Delgrande et al. [2004]), while adding preferences into answer set programming (ASP) has promising advantages from both implementation and application viewpoints [Brewka 2006; Delgrande et al. 2003].

In recent years, as an important enhancement of traditional ASP approaches, first-order answer set programs have intensively been studied by researchers [Asuncion et al. 2012a; Lee and Palla 2010; Zhang and Zhou 2010]. First-order answer set programming generalizes the traditional propositional ASP paradigm in which the semantics of a program with variables is precisely captured by a second-order sentence, and hence program grounding will be no longer needed to compute answer sets of the underlying program [Ferraris et al. 2011; Lin and Zhou 2011]. This provides a new research direction to develop more efficient ASP solvers, because it has been well understood that program grounding is the most computationally expensive phase in all current ASP solvers, that is, Gebser et al. [2007].

---

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. Authors' address: V. Asuncion, Y. Zhang (corresponding author), and Y. Zhou, School of Computing, Engineering and Mathematics, University of Western Sydney, Locked Bag 1797, Penrith, NSW 2751, Australia; email: {vernon, yan, yzhou}@scem.uws.edu.au.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2014 ACM 1529-3785/2014/04-ART11 \$15.00

DOI: <http://dx.doi.org/10.1145/2579817>

An important research agenda in this direction is to redevelop important functionalities and properties, that have been successful in propositional ASP, under the new framework of first-order ASP [Asuncion et al. 2012a, 2012b; Lee and Meng 2009]. In this article, we propose a semantic framework for preferred first-order answer set programs in the case of normal logic programs, and show how preference reasoning is properly captured under this new framework. In particular, in this article, we make the following original contributions towards the development of preferred first-order answer set programming:

- We propose a progression-based preference semantics for first-order answer set programs. This semantics is a generalization of the progression semantics for first-order normal answer set programs proposed by Zhang and Zhou [2010], which also extends Delgrande et al.’s preference semantics for propositional ASP [Delgrande et al. 2003; Schaub and Wang 2003].
- We investigate essential semantic properties of preference reasoning under the proposed preferred first-order ASP. In order to prove these important properties, we specifically consider the grounding of preferred answer set programs and establish its connections to the first-order case.
- Finally, we address the expressiveness of preferred first-order ASP in relation to classical second-order logic. In particular, we show that the proposed preferred semantics can be precisely represented by a second-order sentence on arbitrary structures. Furthermore, by restricting on finite structures, the preferred semantics can be characterized by an existential second-order sentence. As a consequence of this result, we know that on finite structures, a preferred first-order normal logic program can always be represented by a first-order sentence under an extended vocabulary. Applying such second-order characterization method, we further generalize Schaub and Wang’s and Brewka and Eiter’s approaches of preferred logic programs [Schaub and Wang 2003; Brewka and Eiter 1999] to the corresponding first-order cases respectively.

The rest of the article is organized as follows. Section 2 provides logical preliminaries that we will need throughout this article. Section 3 focuses on the development of a progression-based semantics for preferred first-order answer set programs. Section 4 proposes a grounding semantic characterization for preferred first-order answer set programs, and investigates several important semantic properties based on such grounding characterization. Section 5 then provides a logical formulation on the preferred answer set semantics, which shows that the progression-based preferred answer set semantics can be precisely represented by a second-order sentence on arbitrary structures. Such formulation may be further simplified if only finite structures are considered. Section 6 compares our approach with other existing propositional preferred logic programming frameworks in detail, and reveals some interesting insights in this aspect. Finally, Section 7 concludes the article with some remarks.

## 2. BASIC CONCEPTS

In this section, we first introduce necessary logical concepts and notions that we will need in this article, and then provide a semantic overview of first-order answer set programs.

### 2.1. Logical Preliminaries

We start with necessary logic notions and concepts. We consider a second-order language without function symbols but with equality. A *vocabulary*  $\tau$  is a finite set that consists of *relation symbols* (or *predicates*) including the *equality symbol*  $=$  and *constant*

symbols (or constants). Each predicate is associated with a natural number, called its *arity*. Given a vocabulary, *term*, *atom*, *substitution*, (first-order and second-order) *formula* and (first-order and second-order) *sentence* are defined in a standard way. In particular, an atom is called an *equality atom* if it has the form  $t_1 = t_2$ , where  $t_1$  and  $t_2$  are terms. Otherwise, it is called a *proper atom*.

A *structure*  $\mathcal{A}$  of vocabulary  $\tau$  (or a  $\tau$ -*structure*) is a tuple  $\mathcal{A} = (A, c_1^A, \dots, c_m^A, P_1^A, \dots, P_n^A)$ , where  $A$  is a nonempty set called the *domain* of  $\mathcal{A}$  (sometimes we use  $\text{Dom}(\mathcal{A})$  to denote  $\mathcal{A}$ 's domain),  $c_i^A$  ( $1 \leq i \leq m$ ) is an element in  $A$  for every constant  $c_i$  in  $\tau$ , and  $P_j^A$  ( $1 \leq j \leq n$ ) is a  $k$ -ary relation over  $A$  for every  $k$ -ary predicate  $P_j$  in  $\tau$ .  $P_j^A$  is also called the *interpretation* of  $P_j$  in  $\mathcal{A}$ . A structure is *finite* if its domain is a finite set. In this article, we will consider arbitrary structures without restricting to finite structures unless we specifically mention it.

Given two vocabularies  $\tau_1$  and  $\tau_2$  where  $\tau_2 \subseteq \tau_1$ , and a structure  $\mathcal{A}$  of  $\tau_1$ , we say that the *restriction* of  $\mathcal{A}$  on  $\tau_2$ , denoted by  $\mathcal{A} \upharpoonright_{\tau_2}$ , is a structure of  $\tau_2$  whose domain is the same as  $\mathcal{A}$ 's, and for each constant  $c$  and relation symbol  $R$  in  $\tau_2$ ,  $c^A$  and  $R^A$  are in  $\mathcal{A} \upharpoonright_{\tau_2}$ . On the other hand, if we are given a structure  $\mathcal{A}'$  of  $\tau_2$ , a structure  $\mathcal{A}$  of  $\tau_1$  is an *expansion* of  $\mathcal{A}'$  to  $\tau_1$ , if  $\mathcal{A}$  has the same domain of  $\mathcal{A}'$  and retains all  $c^{A'}$  and  $R^{A'}$  for all constants  $c$  and relation symbols  $R$  in  $\tau_2$ .

Let  $\mathcal{A}$  be a  $\tau$ -structure and  $A = \text{Dom}(\mathcal{A})$ . An *assignment* in  $\mathcal{A}$  is a function  $\eta$  from the set of variables to  $A$ . An assignment can be extended to a corresponding function from the set of terms to  $A$  by mapping  $\eta(c)$  to  $c^A$ , where  $c$  is an arbitrary constant. Let  $P(\vec{x})$  be an atom and  $\eta$  an assignment in structure  $\mathcal{A}$ . We also use  $P(\vec{x})\eta \in \mathcal{A}$  to denote  $\eta(\vec{x}) \in P^A$ . The *satisfaction relation*  $\models$  between a structure  $\mathcal{A}$  and a formula  $\phi$  associated with an assignment  $\eta$ , denoted by  $\mathcal{A} \models \phi[\eta]$ , is defined as usual. Let  $\vec{x}$  be the set of free variables occurring in a formula  $\phi$ . Then, the satisfaction relation only relies on the assignment of  $\vec{x}$ . In this case, we write  $\mathcal{A} \models \phi(\vec{x}/\vec{a})$  to denote the satisfaction relation, where  $\vec{a}$  is a tuple of elements in  $A$ . In particular, if  $\phi$  is a sentence, then the satisfaction relation is independent on the assignment. In this case, we simply write  $\mathcal{A} \models \phi$  for short.

Given a  $\tau$ -structure  $\mathcal{A}$  and an assignment  $\eta$  in  $\mathcal{A}$ , if  $P$  is a predicate in  $\tau$ , then we use  $\mathcal{A} \cup \{P(\vec{x})\eta\}$  to denote a new structure of  $\tau$  which is obtained from  $\mathcal{A}$  by expanding the interpretation of predicate  $P$  in  $\mathcal{A}$  (i.e.,  $P^A$  to  $P^A \cup \{\eta(\vec{x})\}$ ). If  $\mathcal{P}$  is a set of predicates, we simply write  $\mathcal{P}\eta \subseteq \mathcal{A}$  if for each  $P \in \mathcal{P}$ ,  $P(\vec{x})\eta \in P^A$ . Similarly, we write  $\mathcal{P}\eta \cap \mathcal{A} = \emptyset$  if for each  $P \in \mathcal{P}$ ,  $P(\vec{x})\eta \cap P^A = \emptyset$ .

Let  $\mathcal{A}_1$  and  $\mathcal{A}_2$  be two  $\tau$ -structures sharing the same domain, that is,  $\text{Dom}(\mathcal{A}_1) = \text{Dom}(\mathcal{A}_2)$ , and for each constant  $c$  in  $\tau$ ,  $c^{A_1} = c^{A_2}$ . By  $\mathcal{A}_1 \subseteq \mathcal{A}_2$ , we simply mean that for each predicate  $P \in \tau$ ,  $P^{A_1} \subseteq P^{A_2}$ . By  $\mathcal{A}_1 \subset \mathcal{A}_2$ , we mean that  $\mathcal{A}_1 \subseteq \mathcal{A}_2$  but not  $\mathcal{A}_2 \subseteq \mathcal{A}_1$ . We write  $\mathcal{A}_1 \cup \mathcal{A}_2$  to denote the structure of  $\tau$  where the domain of  $\mathcal{A}_1 \cup \mathcal{A}_2$  is the same as  $\mathcal{A}_1$  and  $\mathcal{A}_2$ 's domain, each constant  $c$  is interpreted in the same way as in  $\mathcal{A}_1$  and  $\mathcal{A}_2$ , and for each predicate  $P$  in  $\tau$ ,  $P^{A_1 \cup A_2} = P^{A_1} \cup P^{A_2}$ .

## 2.2. An Overview of First-Order Answer Set Programs

In this article, we will focus on normal answer set programs. Whenever there is no confusion, we may simply say answer set programs or logic programs (programs) in our discussions throughout this article. A *rule* is of the form:

$$\alpha \leftarrow \beta_1, \dots, \beta_k, \text{not } \gamma_1, \dots, \text{not } \gamma_l, \quad (1)$$

where  $\alpha$  is a proper atom or the falsity  $\perp$  (i.e., an empty head), and  $\beta_1, \dots, \beta_k, \gamma_1, \dots, \gamma_l$  ( $k, l \geq 0$ ) are atoms. Here,  $\alpha$  is called the *head*,  $\{\beta_1, \dots, \beta_k\}$  the *positive body* and  $\{\text{not } \gamma_1, \dots, \text{not } \gamma_l\}$ , the *negative body* of the rule, respectively. Sometimes, for convenience

we use  $Head(r)$ ,  $Pos(r)$  and  $Neg(r)$  to denote the sets  $\{\alpha\}$ ,  $\{\beta_1, \dots, \beta_k\}$  and  $\{\gamma_1, \dots, \gamma_l\}$ , respectively.

A (*first-order*) *normal answer set program* (or simply called *program*)  $\Pi$  is a finite set of rules. Every relation symbol occurring in the head of some rule of  $\Pi$  is called an *intentional predicate*, and all other relation symbols in  $\Pi$  are *extensional predicates*. We use the notions  $\tau(\Pi)$  to denote the vocabulary containing all of the relation symbols and constants in  $\Pi$ ,  $\tau_{int}(\Pi)$  to denote the vocabulary containing all intentional predicates in  $\Pi$ , and  $\tau_{ext}(\Pi)$  to denote the vocabulary containing all extensional predicates and constants in  $\Pi$ . We also use notions  $\mathcal{P}(\Pi)$ ,  $\mathcal{P}_{int}(\Pi)$ , and  $\mathcal{P}_{ext}(\Pi)$  to denote the sets of all predicates, intentional, and extensional predicates in  $\Pi$ , respectively. A proper atom  $P(\vec{\tau})$  is *extensional (intentional)* if  $P$  is extensional (intentional). Furthermore, given a set  $X$  of atoms, we denote by  $\text{Pred}(X)$  the set of all the predicate symbols occurring in  $X$ . For instance, given a rule  $r$  of the form (1),  $\text{Pred}(Pos(r))$  denotes the set of all the predicate symbols occurring in the set  $\{\beta_1, \dots, \beta_k\}$ . Obviously, we use  $Head(\Pi)$ ,  $Pos(\Pi)$  and  $Neg(\Pi)$  to denote the unions of all  $Head(r)$ ,  $Pos(r)$  and  $Neg(r)$  for all rules  $r \in \Pi$ , respectively.

For convenience, from here on, we abbreviate by FO and SO the words *first-order* and *second-order* respectively.

### 2.3. Semantics

For an FO answer set program  $\Pi$ , denote by  $\widehat{\Pi}$  the FO sentence:

$$\bigwedge_{\substack{r \in \Pi, \\ r: \alpha \leftarrow \beta_1, \dots, \beta_k, \text{not } \gamma_1, \dots, \text{not } \gamma_l}} \forall \vec{x}_r (\beta_1 \wedge \dots \wedge \beta_k \wedge \neg \gamma_1 \wedge \dots \wedge \neg \gamma_l \rightarrow \alpha), \quad (2)$$

where for a rule  $r \in \Pi$ ,  $\vec{x}_r$  denotes the tuple of distinguishable variables occurring in  $r$ . The formula  $\widehat{\Pi}$  is usually referred to as the *universal closure* of  $\Pi$ . Then, a  $\tau(\Pi)$ -structure  $\mathcal{A}$  is an *answer set* of  $\Pi$  iff  $\mathcal{A}$  is a model of the following SO sentence  $\varphi$ :

$$\widehat{\Pi} \wedge \neg \exists \vec{u} (\vec{u} < \vec{P} \wedge \widehat{\Pi}^*(\vec{u})), \quad (3)$$

where:

- $\vec{P} = P_1 \dots P_k$  denotes the tuple of intentional predicates  $P_1, \dots, P_k$  of  $\Pi$ ;
- $\vec{u} = u_1 \dots u_k$  denotes the tuple of predicate variables  $u_1, \dots, u_k$  such that for each  $i$  ( $1 \leq i \leq k$ ), the arity of  $u_i$  matches that of  $P_i$ ;
- $\vec{u} < \vec{P}$  is an SO formula:  
 $\bigwedge_{P_i \in \vec{P}} \forall \vec{x} (u_i(\vec{x}) \rightarrow P_i(\vec{x})) \wedge \neg \bigwedge_{P_i \in \vec{P}} \forall \vec{x} (P_i(\vec{x}) \rightarrow u_i(\vec{x}));$
- and finally,  $\widehat{\Pi}^*(\vec{u})$  is an SO formula obtained from  $\widehat{\Pi}$  by replacing every  $P_i$  occurring among  $\alpha, \beta_1, \dots, \beta_k$  in Formula (2) with the predicate variable  $u_i$  ( $1 \leq i \leq k$ ) on  $\widehat{\Pi}$ .

These semantics for the FO answer set program generalizes the traditional stable model semantics for propositional answer set programs [Baral 2003; Gelfond and Lifschitz 1988]. Here under the given FO structure, the extensional predicates are viewed as the initial input to the program  $\Pi$ , while minimization applies to the underlying intentional predicates. It has been shown by Chen et al. [2011] that the given FO answer set program semantics definition is equivalent to Ferraris et al.'s original FO stable model semantics [Ferraris et al. 2011] when we restrict to normal answer set programs.

*Example 2.1.* We consider a simple program  $\Pi$  consisting of the following two rules:

$$\begin{aligned} P(x) &\leftarrow \text{not } Q(x), \\ Q(x) &\leftarrow Q(x). \end{aligned}$$

Note that program  $\Pi$  has no extensional predicate. According to the given definition,  $\Pi$ 's answer sets are precisely the models of the formula  $\forall x(P(x) \wedge \neg Q(x))$ .

### 3. A PROGRESSION-BASED SEMANTICS

Let  $\Pi$  be an FO program, and  $<\subseteq \Pi \times \Pi$  an irreflexive and transitive relation. Then  $(\Pi, <)$  is called a *preferred FO program*. Intuitively, if  $r_1, r_2 \in \Pi$  and  $(r_1, r_2) \in <$  (we may write  $r_1 < r_2$ ), we mean that  $r_1$  is *more preferred than*  $r_2$ . That is, when we evaluate  $\Pi$ , we consider that  $r_1$  has a higher priority than  $r_2$  during the evaluation. However, this intuition is quite vague, and we need to make this precise.

To develop a semantics for preferred FO answer set programs, we first introduce some useful notions. Let  $\Pi$  be an FO program and  $r$  a rule in  $\Pi$ . We denote by  $\text{Var}(r)$  and  $\text{Const}(r)$  the set of all variables and constants occurring in  $r$  respectively. We also denote by  $\text{Term}(r)$  the set  $\text{Var}(r) \cup \text{Const}(r)$ . Consider a  $\tau(\Pi)$ -structure  $\mathcal{M}$  and  $r \in \Pi$ . We define the set  $\Sigma(\Pi)_{\mathcal{M}}$  as follows:

$$\Sigma(\Pi)_{\mathcal{M}} = \{(r, \eta) \mid r \in \Pi \text{ and } \eta : \text{Term}(r) \longrightarrow \text{Dom}(\mathcal{M})\}.$$

Basically,  $\Sigma(\Pi)_{\mathcal{M}}$  contains all rules of  $\Pi$  with the reference of all possible associated assignments in the structure  $\mathcal{M}$ . Such assignment references will be essential in the development of the semantics for preferred programs.

Given a program  $\Pi$ , let  $\mathcal{M}$  be a  $\tau(\Pi)$ -structure. We specify  $\mathcal{M}^0(\Pi)$  to be a new  $\tau(\Pi)$ -structure obtained from  $\mathcal{M}$  as follows:

$$\mathcal{M}^0(\Pi) = (\text{Dom}(\mathcal{M}), c_1^{\mathcal{M}^0}, \dots, c_r^{\mathcal{M}^0}, P_1^{\mathcal{M}^0}, \dots, P_s^{\mathcal{M}^0}, Q_1^{\mathcal{M}^0}, \dots, Q_n^{\mathcal{M}^0}),$$

where  $c_i^{\mathcal{M}^0} = c_i^{\mathcal{M}}$  for each constant  $c_i$  of  $\tau(\Pi)$  ( $1 \leq i \leq r$ ),  $P_j^{\mathcal{M}^0} = P_j^{\mathcal{M}}$  for each extensional predicate  $P_j$  in  $\tau_{ext}(\Pi)$  ( $1 \leq j \leq s$ ), and  $Q_k^{\mathcal{M}^0} = \emptyset$  for each intentional predicate  $Q_k$  in  $\tau_{int}(\Pi)$  ( $1 \leq k \leq n$ ). Furthermore, for some  $X \subseteq \Sigma(\Pi)_{\mathcal{M}}$ , we also define  $\lambda_{\mathcal{M}^0}(X)$  to be a  $\tau(\Pi)$ -structure generated from  $\mathcal{M}^0$  and  $X$  in the following way:

$$\lambda_{\mathcal{M}^0}(X) = \mathcal{M}^0(\Pi) \cup \{\text{Head}(r)\eta \mid (r, \eta) \in X\}.$$

We are now ready to present a progression-based semantics for preferred FO programs.

*Definition 3.1 (Preferred Evaluation Stage).* Let  $(\Pi, <)$  be a preferred FO answer set program and  $\mathcal{M}$  a  $\tau(\Pi)$ -structure. We define a sequence as follows:

$$\begin{aligned} \Gamma^0(\Pi)_{\mathcal{M}} &= \{(r, \eta) \mid (1) \text{Pos}(r)\eta \subseteq \mathcal{M}^0(\Pi) \text{ and } \text{Neg}(r)\eta \cap \mathcal{M} = \emptyset; \\ &\quad (2) \text{ there does not exist a rule } r' \in \Pi \text{ and an assignment } \eta' \\ &\quad \text{such that } r' < r, \text{Pos}(r')\eta' \subseteq \mathcal{M} \text{ and} \\ &\quad \text{Neg}(r')\eta' \cap \mathcal{M}^0(\Pi) = \emptyset\}; \\ \Gamma^{t+1}(\Pi)_{\mathcal{M}} &= \Gamma^t(\Pi)_{\mathcal{M}} \cup \{(r, \eta) \mid (1) \text{Pos}(r)\eta \subseteq \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) \text{ and} \\ &\quad \text{Neg}(r)\eta \cap \mathcal{M} = \emptyset; \\ &\quad (2) \text{ there does not exist a rule } r' \in \Pi \\ &\quad \text{and an assignment } \eta' \text{ such that } r' < r, \\ &\quad (r', \eta') \notin \Gamma^t(\Pi)_{\mathcal{M}}, \text{ and } \text{Pos}(r')\eta' \subseteq \mathcal{M} \\ &\quad \text{and } \text{Neg}(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) = \emptyset\}. \end{aligned}$$

Let  $\Gamma^\infty(\Pi)_{\mathcal{M}} = \bigcup_{t=0}^\infty \Gamma^t(\Pi)_{\mathcal{M}}$ , where  $\Gamma^t(\Pi)_{\mathcal{M}}$  is called the *t-th preferred evaluation stage of  $(\Pi, <)$  based on  $\mathcal{M}$* .

Let us take a closer look at Definition 3.1. First,  $\mathcal{M}^0(\Pi)$  contains all extensional relations as input, and any rule  $r$  with the highest priority, where its positive body is satisfied in  $\mathcal{M}^0(\Pi)$  (i.e.,  $\text{Pos}(r)\eta \subseteq \mathcal{M}^0(\Pi)$ ) and negative body is not satisfied in  $\mathcal{M}$  (i.e.,

$Neg(r)\eta \cap \mathcal{M}^0(\Pi) = \emptyset$ ), will be generated along with the underlying assignment  $\eta$  (i.e., as a tuple  $(r, \eta)$ ). This forms the initial stage,  $\Gamma^0(\Pi)_{\mathcal{M}}$ .

Second, we consider the structure  $\lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}})$ , that is obtained from the structure  $\mathcal{M}^0(\Pi)$  by expanding all relations derived from the rules in  $\Gamma^t(\Pi)_{\mathcal{M}}$  (with the associated assignments). Then, the structure  $\lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}})$  will be used as a basis to evaluate program  $(\Pi, <)$  at the  $(t+1)$ -th stage. Condition (1) in  $\Gamma^{t+1}(\Pi)_{\mathcal{M}}$  is quite straightforward: the positive body of the rule to be generated should be satisfied by the structure obtained from the previous evaluation stages, and the negative body should not be satisfied in  $\mathcal{M}$ . Condition (2) in  $\Gamma^{t+1}(\Pi)_{\mathcal{M}}$ , on the other hand, takes the preference into account. In particular, in addition to condition (1), we further require a rule to be generated based on two criteria: (a) no other rules with higher priorities have *not* been generated earlier; and (b) these rules' positive bodies are already satisfied in  $\mathcal{M}$  and their negative bodies are not satisfied (not defeated) by the structure  $\lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}})$  generated from the previous stages. Generally speaking, this strategy ensures that when we consider a rule for application, other *more preferred* rules had already been settled in the sense that the more preferred rules had already been derived or that they cannot possibly ever be derived since either their positive bodies are not satisfied by  $\mathcal{M}$  or that they are already defeated by the structure from the previous stage.

Definition 3.1 may be viewed as the generalization of Zhang and Zhou's progression semantics [Zhang and Zhou 2010] for the FO normal answer set programs by taking preference into account. We should also emphasize its connection to Schaub and Wang's order preservation semantics for propositional preferred logic programs [Schaub and Wang 2003]. Specifically, the condition (2) in the specification of  $\Gamma^{t+1}(\Pi)_{\mathcal{M}}$  is a first-order generalization of Schaub and Wang's immediate consequence operator (i.e., operator  $T_{(\Pi, <), Y} X$  for D-preference in Definition 6 in Schaub and Wang [2003]). Nevertheless, Schaub and Wang's order preservation semantics may not be directly extended into our first-order case, because during each evaluation stage, we must keep track of the assignments that are applied to all rules in the progression evaluation. Without such assignment references, the rules generated from the evaluation stage will lose their preference features because rules with different priorities may be instantiated to the same grounded rule.

*Definition 3.2 (Progression-Based Preferred Semantics).* Let  $(\Pi, <)$  be a preferred FO program and  $\mathcal{M}$  a  $\tau(\Pi)$ -structure.  $\mathcal{M}$  is called a *preferred answer set* of  $(\Pi, <)$  iff  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) = \mathcal{M}$ .

*Example 3.3.* Let us consider the following simple preferred program  $(\Pi_1, <_1)$ , where

$$\begin{aligned} r_1: & \text{Flies}(x) \leftarrow \text{Bird}(x), \text{ not } \text{Cannot\_fly}(x), \\ r_2: & \text{Cannot\_fly}(x) \leftarrow \text{Penguin}(x), \text{ not } \text{Flies}(x), \\ & r_2 <_1 r_1. \end{aligned}$$

We consider a finite structure  $\mathcal{M}$ , where

$$\begin{aligned} \text{Dom}(\mathcal{M}) &= \{\text{cody}, \text{tweety}\}, \\ \text{Bird}^{\mathcal{M}} &= \{\text{cody}, \text{tweety}\}, \\ \text{Penguin}^{\mathcal{M}} &= \{\text{tweety}\}, \\ \text{Flies}^{\mathcal{M}} &= \{\text{cody}\}, \text{ and} \\ \text{Cannot\_fly}^{\mathcal{M}} &= \{\text{tweety}\}. \end{aligned}$$

In  $\Pi_1$ , *Bird* and *Penguin* are extensional predicates, and *Flies* and *Cannot\_fly* are intentional. According to Definition 3.1, we have

$$\begin{aligned}\Gamma^0(\Pi_1)_{\mathcal{M}} &= \{(r_2, \eta) \mid \eta : \{x\} \longrightarrow \{tweety\}\}, \\ \Gamma^2(\Pi_1)_{\mathcal{M}} &= \Gamma^1(\Pi_1)_{\mathcal{M}} = \Gamma^0(\Pi_1)_{\mathcal{M}} \cup \{(r_1, \eta') \mid \eta' : \{x\} \longrightarrow \{cody\}\}.\end{aligned}$$

Then from Definition 3.2, we have

$$\lambda_{\mathcal{M}^0}(\Gamma^2(\Pi_1)_{\mathcal{M}}) = \mathcal{M}^0 \cup \{\text{Cannot\_fly}(tweety), \text{Flies}(cody)\} = \mathcal{M}.$$

So  $\mathcal{M}$  is a preferred answer set of  $(\Pi_1, <_1)$ .

**PROPOSITION 3.4.** *Let  $(\Pi, <)$  be a preferred program. If  $< = \emptyset$ , then a structure  $\mathcal{M}$  of  $\tau(\Pi)$  is a preferred answer set of  $(\Pi, <)$  iff  $\mathcal{M}$  is an answer set of  $\Pi$ .*

**PROOF.** By restricting the preference relation  $<$  to be empty, the definition of  $\Gamma^t(\Pi)_{\mathcal{M}}$  is simplified to the following form:

$$\begin{aligned}\Gamma^0(\Pi)_{\mathcal{M}} &= \{(r, \eta) \mid \text{Pos}(r)\eta \subseteq \mathcal{M}^0(\Pi) \text{ and } \text{Neg}(r)\eta \cap \mathcal{M} = \emptyset\}; \\ \Gamma^{t+1}(\Pi)_{\mathcal{M}} &= \Gamma^t(\Pi)_{\mathcal{M}} \cup \{(r, \eta) \mid \text{Pos}(r) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) \\ &\quad \text{and } \text{Neg}(r)\eta \cap \mathcal{M} = \emptyset\}.\end{aligned}$$

It is easy to observe the correspondence between  $\Gamma^t(\Pi)_{\mathcal{M}}$  and  $\mathcal{M}^t(\Pi)$  in Definition 1 from Zhang and Zhou [2010]. Indeed, we can show that for each  $t \geq 0$ ,  $\lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) = \mathcal{M}^{t+1}(\Pi)$ . Therefore,  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) = \mathcal{M}^\infty(\Pi)$ . Then from Theorem 1 by Zhang and Zhou [2010], we conclude that  $\mathcal{M}$  is a preferred answer set of  $(\Pi, \emptyset)$  iff  $\mathcal{M}$  is an answer set of  $\Pi$ .  $\square$

**PROPOSITION 3.5.** *Let  $(\Pi, <)$  be a preferred program. If a  $\tau(\Pi)$ -structure  $\mathcal{M}$  is a preferred answer set of  $(\Pi, <)$ , then  $\mathcal{M}$  is an answer set of  $\Pi$ .*

**PROOF.** We prove this proposition by using a result from Zhang and Zhou [2010], which provides a progression-based answer set semantics for FO answer set programs. Such progression semantics is a special case of our progression-based preferred semantics specified in Definition 3.1.

We first give Zhang and Zhou's progression semantics as follows. Let  $\Pi$  be an FO answer set program and  $\Omega_\Pi = \{Q_1, \dots, Q_n\}$  the set of all the intentional predicates of  $\Pi$ . Consider a structure  $\mathcal{M}$  of  $\tau(\Pi)$ . The  $t$ -th simultaneous evolution stage of  $\Pi$  based on  $\mathcal{M}$  ( $t \geq 0$ ), denoted as  $\mathcal{M}^t(\Pi)$ , is a structure of  $\tau(\Pi)$  defined inductively as follows:

$$\begin{aligned}\mathcal{M}^0(\Pi) &= (\text{Dom}(\mathcal{M}), c_1^{\mathcal{M}^0}, \dots, c_r^{\mathcal{M}^0}, P_1^{\mathcal{M}^0}, \dots, P_s^{\mathcal{M}^0}, Q_1^{\mathcal{M}^0}, \dots, Q_n^{\mathcal{M}^0}), \\ &\quad \text{where } c_i^{\mathcal{M}^0} = c_i^{\mathcal{M}} \text{ for each constant } c_i \text{ of } \tau \text{ (} 1 \leq i \leq r \text{),} \\ &\quad P_j^{\mathcal{M}^0} = P_j^{\mathcal{M}} \text{ for each extensional predicate } P_j \text{ in } \tau_{\text{ext}}(\Pi) \\ &\quad (1 \leq j \leq s), \text{ and } Q_k^{\mathcal{M}^0} = \emptyset \text{ for each intentional predicate } Q_k \\ &\quad \text{in } \tau_{\text{int}}(\Pi) \text{ (} 1 \leq k \leq n \text{);}\end{aligned}$$

$$\begin{aligned}\mathcal{M}^{t+1}(\Pi) &= \mathcal{M}^t(\Pi) \cup \{Q_i(\vec{x})\eta \mid \text{there exists a rule} \\ &\quad Q_i(\vec{x}) \leftarrow \beta_1, \dots, \beta_m, \text{not } \gamma_1, \dots, \text{not } \gamma_l \in \Pi \text{ and an assignment } \eta \\ &\quad \text{such that for all } j \text{ (} 1 \leq j \leq m \text{), } \beta_j\eta \in \mathcal{M}^t(\Pi), \text{ and for all} \\ &\quad k \text{ (} 1 \leq k \leq l \text{), } \gamma_k\eta \notin \mathcal{M}^t(\Pi)\};\end{aligned}$$

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

Then, by Theorem 1 from Zhang and Zhou [2010], we know that  $\mathcal{M}$  is an answer set of  $\Pi$  iff  $\mathcal{M}^\infty(\Pi) = \mathcal{M}$ .

Now to prove this proposition, we will show that under the condition  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) = \mathcal{M}$ ,  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) \subseteq \mathcal{M}^\infty(\Pi)$  and  $\mathcal{M}^\infty(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . Indeed, it is easy to show the former holds by the definition of  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . Therefore, it is only left to show  $\mathcal{M}^\infty(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . We show this by induction on  $t$  that  $\mathcal{M}^t(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ .

*Basis.* Clearly, since  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) = \mathcal{M}^0(\Pi) \cup \{\text{Head}(r)\eta \mid (r, \eta) \in \Gamma^\infty(\Pi)_{\mathcal{M}}\}$ , then it immediately follows that  $\mathcal{M}^0(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ .

*Step.* Assume for  $0 \leq t' \leq t$  we have  $\mathcal{M}^{t'}(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . We show  $\mathcal{M}^{t+1}(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . On the contrary, suppose  $\mathcal{M}^{t+1}(\Pi) \not\subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . Then there exists a rule  $r \in \Pi$  and an assignment  $\eta$  such that  $\text{Pos}(r)\eta \subseteq \mathcal{M}^t(\Pi)$  and  $\text{Neg}(r)\eta \cap \mathcal{M} = \emptyset$ , and where  $\text{Head}(r)\eta \in \mathcal{M}^{t+1}(\Pi)$  and  $\text{Head}(r)\eta \notin \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . Now, since  $\mathcal{M}^t(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$  (i.e., by the inductive assumption), then for some  $n \geq 0$ , we have  $\mathcal{M}^t(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^n(\Pi)_{\mathcal{M}})$ . Furthermore, since  $\text{Head}(r)\eta \notin \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ , then we also have  $\text{Head}(r)\eta \notin \lambda_{\mathcal{M}^0}(\Gamma^{n+1}(\Pi)_{\mathcal{M}})$ , and hence, that  $(r, \eta) \notin \Gamma^{n+1}(\Pi)_{\mathcal{M}}$ . Then, by the definition of  $\Gamma^{n+1}(\Pi)_{\mathcal{M}}$ , there must exist a rule  $r' < r$  and an assignment  $\eta'$  such that:

- (a)  $\text{Pos}(r')\eta' \subseteq \mathcal{M}$  and  $\text{Neg}(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^n(\Pi)_{\mathcal{M}}) = \emptyset$ ;
- (b)  $(r', \eta') \notin \Gamma^n(\Pi)_{\mathcal{M}}$ ,

that is, a rule blocking  $r$  from being applied at stage  $n + 1$ . Now, for a  $k \geq 0$  and (arbitrary) rule  $r^* \in \Pi$  and corresponding assignment  $\eta^*$ , set  $B^k(r^*, \eta^*)$  to be such that

$$\begin{aligned} B^0(r^*, \eta^*) &= \{(r', \eta') \mid r' < r^*, \text{Pos}(r')\eta' \subseteq \mathcal{M} \text{ and } \text{Neg}(r')\eta' \cap \mathcal{M}^0(\Pi) = \emptyset\} \text{ and} \\ B^k(r^*, \eta^*) &= \{(r', \eta') \mid \text{(a) } r' < r^*; \\ &\quad \text{(b) } \text{Pos}(r')\eta' \subseteq \mathcal{M} \text{ and } \text{Neg}(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^k(\Pi)_{\mathcal{M}}) = \emptyset; \\ &\quad \text{(c) } (r', \eta') \notin \Gamma^k(\Pi)_{\mathcal{M}}\} \end{aligned}$$

for  $k \geq 1$ . Intuitively,  $B^k(r^*, \eta^*)$  comprises the pair  $(r', \eta')$  that blocks  $(r^*, \eta^*)$  from being applied at stage  $k + 1$ . Now, the following Claims 1, 2, and 3 reveal important properties of the set  $B^k(r^*, \eta^*)$ .

**CLAIM 1.** *For all  $k' \geq k \geq 1$ , we have  $B^k(r^*, \eta^*) \supseteq B^{k'}(r^*, \eta^*)$ . In other words, we do not gain anymore pairs  $(r', \eta')$  blocking  $(r^*, \eta^*)$  from being applied as we progress along the stages of  $\Gamma^k(\Pi)_{\mathcal{M}}$ .*

**PROOF OF CLAIM 1.** For simplicity we assume that  $k \geq 1$  (the case where we allow  $k = 0$  immediately follows). Set  $k'$  to be such that  $k' \geq k$  and let  $(r'', \eta'') \in B^{k'}(r^*, \eta^*)$ . Then by the definition of  $B^{k'}(r^*, \eta^*)$ , we have:

- (a)  $r'' < r^*$ ;
- (b)  $\text{Pos}(r'')\eta'' \subseteq \mathcal{M}$  and  $\text{Neg}(r'')\eta'' \cap \lambda_{\mathcal{M}^0}(\Gamma^{k'}(\Pi)_{\mathcal{M}}) = \emptyset$ ;
- (c)  $(r'', \eta'') \notin \Gamma^{k'}(\Pi)_{\mathcal{M}}$ .

Then by the monotonicity of  $\Gamma^k(\Pi)_{\mathcal{M}}$  for  $k' \geq k$  (i.e.,  $\Gamma^{k'}(\Pi)_{\mathcal{M}} \subseteq \Gamma^k(\Pi)_{\mathcal{M}}$ ), we also have:

- (a)  $r'' < r^*$ ;
- (b)  $\text{Pos}(r'')\eta'' \subseteq \mathcal{M}$  and  $\text{Neg}(r'')\eta'' \cap \lambda_{\mathcal{M}^0}(\Gamma^k(\Pi)_{\mathcal{M}}) = \emptyset$ ;
- (c)  $(r'', \eta'') \notin \Gamma^k(\Pi)_{\mathcal{M}}$ ,

and hence, that  $(r'', \eta'') \in B^k(r^*, \eta^*)$ . This ends the proof of Claim 1.



**CLAIM 2.** For each  $(r', \eta') \in B^k(r^*, \eta^*)$ ,  $B^k(r', \eta') \subseteq B^k(r^*, \eta^*)$ . That is, for each pairs  $(r', \eta') \in B^k(r^*, \eta^*)$  blocking  $(r^*, \eta^*)$  from being applied at stage  $k+1$ , the pairs  $(r'', \eta'')$  blocking  $(r', \eta')$  in turn at stage  $k+1$ , are themselves in  $B^k(r^*, \eta^*)$ . Intuitively, this implies  $B^k(r^*, \eta^*)$  satisfies some form of closure.

**PROOF OF CLAIM 2.** Set  $(r', \eta') \in B^k(r^*, \eta^*)$  and let  $(r'', \eta'') \in B^k(r', \eta')$ . Then we have  $r' < r^*$  and  $r'' < r'$  by the definitions of  $B^k(r^*, \eta^*)$  and  $B^k(r', \eta')$ , respectively. Then, by transitivity, we also have  $r'' < r^*$ . Hence, by the definition of  $B^k(r^*, \eta^*)$ , we also have  $(r'', \eta'') \in B^k(r^*, \eta^*)$ . This ends the proof of Claim 2.

**CLAIM 3.**  $B^k(r^*, \eta^*) = \emptyset$  for some ordinal  $k \geq 1$ . That is, eventually, there will be some stage  $k \geq 1$  such that there will be no more pairs  $(r', \eta')$  blocking  $(r^*, \eta^*)$  from being applied. First, we provide the intuition of the proof of Claim 3.

Basically, under the assumption  $\Gamma^\infty(\Pi)_\mathcal{M} = \mathcal{M}$ , we prove by showing that for all the pairs  $(r', \eta') \in B^k(r^*, \eta^*)$ , there will be some  $l > 0$  such that either  $\Gamma^{k+l}(\Pi)_\mathcal{M}$  will “defeat”  $(r', \eta')$  (i.e., as in  $Neg(r')\eta' \cap \Gamma^{k+l}(\Pi)_\mathcal{M} \neq \emptyset$ ) or gets “eaten up” by it (i.e., as in  $(r', \eta') \in \Gamma^{k+l}(\Pi)_\mathcal{M}$ ).

**PROOF OF CLAIM 3.** First, we show for all  $k \geq 1$ ,  $B^k(r^*, \eta^*) \neq \emptyset$  implies there exists some  $l > 0$  such that  $B^k(r^*, \eta^*) \supset B^{k+l}(r^*, \eta^*)$ . Thus, for a  $k \geq 1$ , let  $(r', \eta') \in B^k(r^*, \eta^*)$ . Moreover, without loss of generality, assume that for all  $(r'', \eta'') \in B^k(r^*, \eta^*)$ , we have  $r'' \not< r'$  (i.e., note that due to the finiteness of  $\Pi$ , we will always have these pairs  $(r', \eta') \in B^k(r^*, \eta^*)$ ). Then we have  $B^k(r', \eta') = \emptyset$ . This is because if  $B^k(r', \eta') \neq \emptyset$  and let  $(r'', \eta'') \in B^k(r', \eta')$ , we will have  $r'' < r'$  by the definition of  $B^k(r', \eta')$ , which contradicts the initial assumption about the pair  $(r', \eta')$ . Now, as  $(r', \eta') \in B^k(r^*, \eta^*)$ , then by the definition of  $B^k(r^*, \eta^*)$ , we have  $Pos(r')\eta' \subseteq \mathcal{M}$  and  $Neg(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^k(\Pi)_\mathcal{M}) = \emptyset$ . Also, since  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_\mathcal{M}) = \mathcal{M}$  by assumption, then for some  $l \geq 0$ , we have  $Pos(r')\eta' \subseteq \lambda_{\mathcal{M}^0}(\Gamma^{k+l}(\Pi)_\mathcal{M})$ . Now there can only be two possibilities.

*Case 1.*  $Neg(r')\eta' \cap \mathcal{M} = \emptyset$ . Then since we also have  $B^{k+l}(r', \eta') = \emptyset$  by Claim 1 (i.e., since  $B^k(r', \eta') = \emptyset$ ), then we have  $(r', \eta') \in \mathcal{M}^0(\Gamma^{k+l+1}(\Pi)_\mathcal{M})$  by the definition of  $\Gamma^{k+l+1}(\Pi)_\mathcal{M}$  since  $(r', \eta')$  will now be applicable at this stage (i.e., since there are no pairs  $(r'', \eta'')$  blocking  $(r', \eta')$  from application at stage  $k+l+1$  since  $B^{k+l}(r', \eta') = \emptyset$ ). Then by the definition of  $B^{k+l+1}(r^*, \eta^*)$ , we have  $(r', \eta') \notin B^{k+l+1}(r^*, \eta^*)$ , which implies  $B^k(r^*, \eta^*) \neq B^{k+l+1}(r^*, \eta^*)$ . Then since by Claim 1, we have  $B^k(r^*, \eta^*) \supseteq B^{k+l+1}(r^*, \eta^*)$ , then we must have  $B^k(r^*, \eta^*) \supset B^{k+l+1}(r^*, \eta^*)$ .

*Case 2.*  $Neg(r')\eta' \cap \mathcal{M} \neq \emptyset$ . Then there could be further two possibilities.

—*Subcase 1.*  $Neg(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^{k+l}(\Pi)_\mathcal{M}) \neq \emptyset$ . Then we have  $(r', \eta') \notin B^{k+l}(r^*, \eta^*)$  by the definition of  $B^{k+l}(r^*, \eta^*)$ . Moreover, since  $(r', \eta') \in B^k(r^*, \eta^*)$ , then it must be that  $k+l > k$ , which implies  $l > 0$ . Hence, similarly to before, we have also that  $B^k(r^*, \eta^*) \supset B^{k+l}(r^*, \eta^*)$  with  $l > 0$ .

—*Subcase 2.*  $Neg(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^{k+l}(\Pi)_\mathcal{M}) = \emptyset$ . Then since  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_\mathcal{M}) = \mathcal{M}$  by assumption, there exists some  $m > 0$  such that  $Neg(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^{k+l+m}(\Pi)_\mathcal{M}) \neq \emptyset$ , which implies  $(r', \eta') \notin B^{k+l+m}(r^*, \eta^*)$  by the definition of  $B^{k+l+m}(r^*, \eta^*)$ . Hence, in a similar manner, we have  $B^k(r^*, \eta^*) \supset B^{k+l+m}(r^*, \eta^*)$  with  $m > 0$ .

Therefore, we have that for all  $k \geq 1$ ,  $B^k(r^*, \eta^*) \neq \emptyset$  implies there exists some  $l > 0$  such that  $B^k(r^*, \eta^*) \supset B^{k+l}(r^*, \eta^*)$ . Now we show for all  $k \geq 1$  and  $l > 0$ ,  $B^k(r^*, \eta^*) \supset B^{k+l}(r^*, \eta^*)$  implies  $\Gamma^k(\Pi)_\mathcal{M} \subset \Gamma^{k+l}(\Pi)_\mathcal{M}$ . Hence, assume that  $B^k(r^*, \eta^*) \supset B^{k+l}(r^*, \eta^*)$  and let  $(r', \eta') \in B^k(r^*, \eta^*)$  where  $(r', \eta') \notin B^{k+l}(r^*, \eta^*)$ . Then by the definition of  $B^k(r^*, \eta^*)$  we have:

- (1)  $r' < r^*$ ;
- (2)  $Pos(r')\eta' \subseteq \mathcal{M}$  and  $Neg(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^k(\Pi)_{\mathcal{M}}) = \emptyset$ ;
- (3)  $(r', \eta') \notin \Gamma^k(\Pi)_{\mathcal{M}}$ .

Also, since  $(r', \eta') \notin B^{k+l}(r^*, \eta^*)$ , then by the definition of  $B^{k+l}(r^*, \eta^*)$ , we have either:

- (1)  $Neg(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^{k+l}(\Pi)_{\mathcal{M}}) \neq \emptyset$ , or
- (2)  $(r', \eta') \in \Gamma^{k+l}(\Pi)_{\mathcal{M}}$ .

In either of these two aforementioned possibilities, it can be seen that  $\Gamma^k(\Pi)_{\mathcal{M}} \neq \Gamma^{k+l}(\Pi)_{\mathcal{M}}$ . Therefore, since  $\Gamma^k(\Pi)_{\mathcal{M}} \subseteq \Gamma^{k+l}(\Pi)_{\mathcal{M}}$ , then we must have  $\Gamma^k(\Pi)_{\mathcal{M}} \subset \Gamma^{k+l}(\Pi)_{\mathcal{M}}$ . Hence, from this, we obtain the equivalent true statement: if for all  $k \geq 1$  and  $l > 0$  we have  $\Gamma^k(\Pi)_{\mathcal{M}} = \Gamma^{k+l}(\Pi)_{\mathcal{M}}$ , then  $B^k(r^*, \eta^*) = B^{k+l}(r^*, \eta^*)$ . Now, by the monotonicity of  $\Gamma^k(\Pi)_{\mathcal{M}}$  for all  $k \geq 1$  and its guaranteed convergence, it follows that there exists some  $K \geq 1$  such that for all  $l > 0$ ,  $\Gamma^K(\Pi)_{\mathcal{M}} = \Gamma^{K+l}(\Pi)_{\mathcal{M}}$  (i.e., reached its convergent point). Then we also have  $B^K(r^*, \eta^*) = B^{K+l}(r^*, \eta^*)$ . Then since this holds for all  $l > 0$ , we must have  $B^K(r^*, \eta^*) = \emptyset$  for if  $B^K(r^*, \eta^*) \neq \emptyset$ , then this will contradict the given result that there exists some  $l > 0$  for which  $B^K(r^*, \eta^*) \supset B^{K+l}(r^*, \eta^*)$  (i.e., which also means  $B^K(r^*, \eta^*) \neq B^{K+l}(r^*, \eta^*)$ ). This ends the proof of Claim 3.

Therefore, by Claim 3, there exists some ordinal  $n' > n$  for which  $B^{n'}(r, \eta) = \emptyset$ . Then at this stage (i.e.,  $n'$ ), there would be no other pairs  $(r', \eta')$  blocking  $(r, \eta)$  from being applied. Now, by the monotonicity of  $\Gamma^n(\Pi)_{\mathcal{M}}$  for all  $n \geq 0$ , we have  $Pos(r)\eta \subseteq \lambda_{\mathcal{M}^0}(\Gamma^{n'}(\Pi)_{\mathcal{M}})$  and  $Neg(r)\eta \cap \mathcal{M} = \emptyset$ . Then this implies  $(r, \eta) \in \Gamma^{n'+1}(\Pi)_{\mathcal{M}}$ , which further implies  $Head(r)\eta \in \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . This then contradicts the initial assumption  $Head(r)\eta \notin \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . Therefore, we must have  $\mathcal{M}^{l+1}(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . This completes our proof of  $\mathcal{M}^\infty(\Pi) \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ .  $\square$

#### 4. PROPERTIES OF PREFERRED ANSWER SETS

In this section, we study some essential properties of the preferred answer set semantics proposed in the previous section. First, from Proposition 3.5, we know that for any preferred program  $(\Pi, <)$ , its preferred answer sets must also be answer sets of  $\Pi$ . Now we take a closer look at the relationship between the existence of an answer set of  $\Pi$  and of a preferred one of  $(\Pi, <)$ .

*Example 4.1.* Let  $(\Pi_2, <_2)$  be a preferred program as follows:

$$\begin{aligned} r_1 &: P(x) \leftarrow Q(x), \\ r_2 &: Q(x) \leftarrow, \\ r_1 &<_2 r_2. \end{aligned}$$

Note that  $\Pi_2$  has no extensional predicate, and any structure  $\mathcal{M}$  on  $\tau(\Pi_2)$  where  $P^{\mathcal{M}} = Q^{\mathcal{M}} = \text{Dom}(\mathcal{M})$  is an answer set of  $\Pi_2$ . But  $(\Pi_2, <_2)$  has no preferred answer set. To see this, we consider Definition 3.1. For any  $\mathcal{M}$  which is an answer set of  $\Pi_2$ , we have  $\Gamma^0(\Pi_2)_{\mathcal{M}} = \Gamma^1(\Pi_2)_{\mathcal{M}} = \emptyset$ . It follows that  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi_2)_{\mathcal{M}}) = \mathcal{M}'$  where  $P^{\mathcal{M}'} = Q^{\mathcal{M}'} = \emptyset$ . By Definition 3.2,  $\mathcal{M}$  cannot be a preferred answer set of  $(\Pi_2, <_2)$ .

Now we consider another preferred program  $(\Pi_3, <_3)$  as follows:

$$\begin{aligned} r_1 &: P(y) \leftarrow P(x), Q(x), \\ r_2 &: P(x) \leftarrow Q(x), \\ r_1 &<_3 r_2. \end{aligned}$$

Here,  $Q$  is the only extensional predicate of  $\Pi_3$ . It is not difficult to show that for any answer set  $\mathcal{M}$  of  $\Pi_3$  where  $P^{\mathcal{M}} = Q^{\mathcal{M}} \neq \emptyset$ ,  $\mathcal{M}$  cannot be a preferred answer set of  $(\Pi_3, <_3)$ . However,  $(\Pi_3, <_3)$  has one preferred answer set  $\mathcal{M}'$  in which  $P^{\mathcal{M}'} = Q^{\mathcal{M}'} = \emptyset$ .

Program  $(\Pi_2, <_2)$  from Example 4.1 suggests that the existence of answer set for a program  $\Pi$  does not necessarily imply the existence of a preferred answer set for a corresponding preferred program  $(\Pi, <)$ . On the other hand, program  $(\Pi_3, <_3)$  seems to suggest that if the positive body for each rule of the program contains proper atoms, then a preferred answer set always exists. The following proposition shows that this is the case.

**PROPOSITION 4.2.** *Let  $\Pi$  be an answer set program where for each rule  $r \in \Pi$ ,  $Pos(r)$  contains proper atoms. Then for any preferred program  $(\Pi, <)$  built upon  $\Pi$ ,  $(\Pi, <)$  has a preferred answer set.*

**PROOF.** Since for each rule  $r \in \Pi$ ,  $Pos(r)$  contains proper atoms, we can always construct a structure  $\mathcal{M}$  in which for all proper atoms  $P(\vec{x}) \in \mathcal{P}_{ext}(\Pi) \cup \mathcal{P}_{int}(\Pi)$ , we set  $P^{\mathcal{M}} = \emptyset$ . In this case, we can see that  $\mathcal{M}$  is an answer set of  $\Pi$ . Furthermore, from Definitions 3.1 and 3.2, we know that  $\mathcal{M}$  is also an answer set of  $(\Pi, <)$  because for any preference relation  $< \subseteq \Pi \times \Pi$ ,  $\Gamma^\infty(\Pi)_{\mathcal{M}} = \emptyset$ .  $\square$

It is also worth mentioning that simplifications suitable for nonpreferred programs usually are not applicable to preferred programs. Consider the following program:

$$\begin{aligned} r_1 &: P(x) \leftarrow Q(x), \\ r_2 &: Q(x) \leftarrow R, \\ r_3 &: R \leftarrow, \\ r_1 &< r_3. \end{aligned}$$

This program has no preferred answer set. But if we replace rule  $r_2$  as follows based on the fact that there is no preference relation between  $r_2$  and  $r_3$ :

$$r'_2 : Q(x) \leftarrow,$$

then the new program

$$\begin{aligned} r_1 &: P(x) \leftarrow Q(x), \\ r'_2 &: Q(x) \leftarrow, \\ r_3 &: R \leftarrow, \\ r_1 &< r_3. \end{aligned}$$

has an answer set. In fact, such simplification has already changed the original program's semantics.

#### 4.1. Grounding Preferred Logic Programs

It has been observed that the answer sets of FO programs can always be obtained via its grounded correspondence [Asuncion et al. 2012a]. We would like to explore whether this approach is also suitable for preferred FO answer set programs, because a grounding based approach will provide an effective means to study semantic properties of the corresponding FO preferred logic programs.

Nevertheless, unlike the case of FO answer set programs, a naive grounding method does not work for preferred programs. Let us consider the following program  $(\Pi_4, <_4)$ :

$$\begin{aligned} r_1 &: P(x) \leftarrow Q(y), \\ r_2 &: P(z) \leftarrow Q(z), \\ r_1 &<_4 r_2. \end{aligned}$$

If we simply ground  $\Pi$  under a domain consisting of a singleton  $\{a\}$ , the grounded program would only contain one instance  $\{P(a) \leftarrow Q(a)\}$ , while the original preference relation  $r_1 <_4 r_2$  collapses with this instance. This problem may be avoided by relating each rule of  $\Pi$  with a corresponding tag predicate, as showed in the following.

Let  $(\Pi, <)$  be an FO preferred answer set program on vocabulary  $\tau(\Pi)$ . A preferred program  $(\Pi', <')$  is called the *tagged preferred program* of  $(\Pi, <)$ , if (1) for each rule in  $\Pi$ :

$$r_i : \alpha \leftarrow \beta_1, \dots, \beta_k, \text{not } \gamma_1, \dots, \text{not } \gamma_l,$$

$\Pi'$  contains its *tagged rule* of the form:

$$r'_i : \alpha \leftarrow \beta_1, \dots, \beta_k, \text{not } \gamma_1, \dots, \text{not } \gamma_l, \text{Tag}_i(\vec{x}),$$

where  $\vec{x}$  is the tuple of all variables occurring in  $r_i$  and  $\text{Tag}_i$  is a new extensional predicate not in  $\tau(\Pi)$ ; (2)  $\Pi'$  does not contain any other rules; and (3) for two rules  $r'_i$  and  $r'_j$  in  $\Pi'$ ,  $r'_i <' r'_j$  iff  $r_i < r_j$  specified in  $(\Pi, <)$ .

Let  $(\Pi, <)$  be a preferred answer set program on vocabulary  $\tau(\Pi)$  and  $(\Pi', <')$  the tagged preferred program of  $(\Pi, <)$  on vocabulary  $\tau(\Pi') = \tau(\Pi) \cup \{\text{Tag}_1, \dots, \text{Tag}_k\}$ <sup>1</sup>. Given a  $\tau$ -structure  $\mathcal{M}$ , we construct an expansion  $\mathcal{M}'$  of  $\mathcal{M}$  to  $\tau(\Pi')$  as follows:

- (1)  $\text{Dom}(\mathcal{M}') = \text{Dom}(\mathcal{M})$ ;
- (2) For each predicate  $P$  and constant  $c$  in  $\tau$ ,  $P^{\mathcal{M}'} = P^{\mathcal{M}}$  and  $c^{\mathcal{M}'} = c^{\mathcal{M}}$ ;
- (3) For each  $n$ -ary  $\text{Tag}_i$  in  $\tau'$  ( $1 \leq i \leq k$ ),  $\text{Tag}_i^{\mathcal{M}'} = \text{Dom}(\mathcal{M}')^n$ .

Let  $r' \in \Pi'$  and  $\eta$  an assignment on structure  $\mathcal{M}'$  of  $\tau(\Pi')$ . We use the notation  $r'\eta$  to denote the ground instance of  $r'$  based on  $\eta$ . Now we are ready to define the grounding of a preferred answer set program.

*Definition 4.3.* Let  $(\Pi, <)$  be a preferred answer set program,  $\mathcal{M}$  a structure of  $\tau(\Pi)$ ,  $(\Pi', <')$  the tagged answer set program of  $(\Pi, <)$ , and  $\mathcal{M}'$  the expansion of  $\mathcal{M}$  to  $\tau(\Pi')$  as described earlier. We say that a pair  $(\text{Ground}(\Pi)_{\mathcal{M}}, <^*)$  is the *grounded preferred answer set program* of  $(\Pi, <)$  based on  $\mathcal{M}$ , if

- (1)  $\text{Ground}(\Pi)_{\mathcal{M}} = \{r^* \mid r^* : \text{Head}(r)\eta \leftarrow \text{Body}(r)\eta, \text{Tag}(\vec{x})\eta, \text{ where } r \in \Pi \text{ and } \eta \text{ is an assignment on } \mathcal{M}'\} \cup \{P(\vec{a}) \leftarrow \mid P \in \mathcal{P}_{\text{ext}}(\Pi) \text{ and } P(\vec{a}) \in P^{\mathcal{M}}\} \cup \{\text{Tag}_i(\vec{a}) \leftarrow \mid \text{Tag}_i(\vec{a}) \in \text{Tag}_i^{\mathcal{M}'} (1 \leq i \leq k)\} \cup \{a = a \leftarrow \mid a \text{ is an element of } \text{Dom}(\mathcal{M}')\}$ <sup>2</sup>;
- (2)  $<^* = \{(r_i^*, r_j^*) \mid r_i^*, r_j^* \in \text{Ground}(\Pi)_{\mathcal{M}} \text{ for some } r'_k, r'_l \in \Pi' \text{ such that there exist assignments } \eta \text{ and } \eta' \text{ on } \mathcal{M}' \text{ satisfying } r_i^* = r'_k\eta, r_j^* = r'_l\eta' \text{ and } r'_k <' r'_l\}$ .

*Example 4.4.* We consider the preferred program  $(\Pi_1, <_1)$  from Example 3.3 again. Under the structure  $\mathcal{M}$  given in Example 3.3, the grounded preferred program of  $(\Pi_1, <_1)$  is as follows:

$$\begin{aligned} r_1^* &: \text{Flies}(\text{cody}) \leftarrow \text{Bird}(\text{cody}), \text{not } \text{Cannot\_fly}(\text{cody}), \text{Tag}_1(\text{cody}), \\ r_2^* &: \text{Flies}(\text{tweety}) \leftarrow \text{Bird}(\text{tweety}), \text{not } \text{Cannot\_fly}(\text{tweety}), \text{Tag}_1(\text{tweety}), \\ r_3^* &: \text{Cannot\_fly}(\text{cody}) \leftarrow \text{Penguin}(\text{cody}), \text{not } \text{Flies}(\text{cody}), \text{Tag}_2(\text{cody}), \\ r_4^* &: \text{Cannot\_fly}(\text{tweety}) \leftarrow \text{Penguin}(\text{tweety}), \text{not } \text{Flies}(\text{tweety}), \text{Tag}_2(\text{tweety}), \\ r_5^* &: \text{Bird}(\text{cody}) \leftarrow, \\ r_6^* &: \text{Bird}(\text{tweety}) \leftarrow, \\ r_7^* &: \text{Penguin}(\text{tweety}) \leftarrow, \\ r_8^* &: \text{Tag}_1(\text{cody}) \leftarrow, \\ r_9^* &: \text{Tag}_1(\text{tweety}) \leftarrow, \end{aligned}$$

<sup>1</sup>We assume that  $\Pi$  contains rules  $r_1, \dots, r_k$ .

<sup>2</sup>Here, we view  $a = a$  as a propositional atom.

$$\begin{aligned}
r_{10}^* &: \text{Tag}_2(\text{cody}) \leftarrow, \\
r_{11}^* &: \text{Tag}_2(\text{tweety}) \leftarrow, \\
r_{12}^* &: \text{cody} = \text{cody} \leftarrow, \\
r_{13}^* &: \text{tweety} = \text{tweety} \leftarrow, \\
r_3^* &<^* r_1^*, \\
r_3^* &<^* r_2^*, \\
r_4^* &<^* r_1^*, \\
r_4^* &<^* r_2^*.
\end{aligned}$$

Now let us consider the preferred program  $(\Pi_4, <_4)$  we discussed earlier under the finite structure  $\mathcal{M} = (\{a\}, \{P(a)\}, \{Q(a)\})$ . It is clear that  $\mathcal{M}$  is an answer set of  $\Pi_4$ . By Definition 4.3, it is easy to see that the grounded preferred program of program  $(\Pi_4, <_4)$  consists of the following rules and preferences:

$$\begin{aligned}
r_1^* &: P(a) \leftarrow Q(a), \text{Tag}_1(a), \\
r_2^* &: P(a) \leftarrow Q(a), \text{Tag}_2(a), \\
r_3^* &: Q(a) \leftarrow, \\
r_4^* &: \text{Tag}_1(a) \leftarrow, \\
r_5^* &: \text{Tag}_2(a) \leftarrow, \\
r_6^* &: a = a \leftarrow, \\
r_1^* &<^* r_2^*.
\end{aligned}$$

It is noted that with the tag predicate introduced, the preference in the original preferred program  $(\Pi_4, <_4)$  is preserved in its grounded preferred program.

Basically, a grounded preferred answer set program is presented in a propositional form and may contain an infinite number of rules and preference relations, if an infinite domain is considered. We will define the semantics for such grounded preferred programs, and show that it coincides with the progression-based semantics for FO preferred programs.

*Definition 4.5 (Preferred Answer Sets for Grounded Programs).* Let  $(\Pi, <)$  be a grounded preferred answer set program as obtained from Definition 4.3 and  $S$  a set of propositional atoms. We define a sequence as follows:

$$\begin{aligned}
\Delta^0(\Pi)_S &= \{r \mid (1) \text{Pos}(r) = \emptyset \text{ and } \text{Neg}(r) \cap S = \emptyset; \\
&\quad (2) \text{ there does not exist a rule } r' \in \Pi \text{ such that } r' < r, \\
&\quad \text{Pos}(r') \subseteq S \text{ and } \text{Neg}(r') \cap \text{Head}(r) = \emptyset\}; \\
\Delta^{t+1}(\Pi)_S &= \Delta^t(\Pi)_S \cup \{r \mid (1) \text{Pos}(r) \subseteq \text{Head}(\Delta^t(\Pi)_S) \\
&\quad \text{and } \text{Neg}(r) \cap S = \emptyset; \\
&\quad (2) \text{ there does not exist a rule } r' \in \Pi \text{ such that} \\
&\quad r' < r, r' \notin \Delta^t(\Pi)_S, \text{ and } \text{Pos}(r') \subseteq S \\
&\quad \text{and } \text{Neg}(r') \cap \text{Head}(\Delta^t(\Pi)_S) = \emptyset\}.
\end{aligned}$$

Let  $\Delta^\infty(\Pi)_S = \bigcup_{t=0}^\infty \Delta^t(\Pi)_S$ , where  $\Delta^t(\Pi)_S$  is called the  $t$ -th preferred evaluation stage of  $(\Pi, <)$  based on  $S$ .  $S$  is called a preferred answer set of  $(\Pi, <)$  iff  $\text{Head}(\Delta^\infty(\Pi)_S) = S$ .

*Example 4.6.* Example 4.4 continued. It is easy to see that under Definition 4.5, the grounded preferred program of  $(\Pi_1, <_1)$  has a unique preferred answer set:

$$\begin{aligned}
&\{\text{Bird}(\text{cody}), \text{Bird}(\text{tweety}), \text{Penguin}(\text{tweety}), \text{Flies}(\text{cody}), \text{Cannot\_fly}(\text{tweety}), \\
&\quad \text{Tag}_1(\text{cody}), \text{Tag}_2(\text{cody}), \text{Tag}_1(\text{tweety}), \text{Tag}_2(\text{tweety}), \\
&\quad \text{cody} = \text{cody}, \text{tweety} = \text{tweety}\}.
\end{aligned}$$

On the other hand, the grounded preferred program of  $(\Pi_4, <_4)$  has a unique preferred answer set  $\{P(a), Q(a), Tag_1(a), Tag_2(a), a = a\}$ .

The following result shows that the preferred answer sets of each FO preferred answer set program can be precisely computed through its grounded counterpart.

**THEOREM 4.7.** *Let  $(\Pi, <)$  be an FO preferred answer set program,  $\mathcal{M}$  a structure of  $\tau(\Pi)$ , and  $(Ground(\Pi)_{\mathcal{M}}, <^*)$  the grounded preferred answer set program of  $(\Pi, <)$  based on  $\mathcal{M}$  as defined in Definition 4.3. Then,  $\mathcal{M}$  is a preferred answer set of  $(\Pi, <)$  iff there is a preferred answer set  $S$  of  $(Ground(\Pi)_{\mathcal{M}}, <^*)$  such that  $S \cap \mathcal{M} = \mathcal{M}$ .*

**PROOF.** ( $\Rightarrow$ ) Suppose that  $\mathcal{M}$  is a preferred answer set of  $(\Pi, <)$ . According to Definitions 4.3 and 4.5, we construct a set  $S$  of propositional atoms from the expansion  $\mathcal{M}'$  of  $\mathcal{M}$  as described earlier, and show that  $S$  is a preferred answer set for the grounded preferred program  $(Ground(\Pi)_{\mathcal{M}}, <^*)$  and  $S \cap \mathcal{M} = \mathcal{M}$ . Let  $(\Pi', <')$  be the tagged preferred program obtained from  $(\Pi, <)$ , the construction of  $S$  is as follows:

- (1) For each predicate  $P \in \tau(\Pi')$  other than  $=$ , if  $P(\vec{a}) \in P^{\mathcal{M}'}$ , then  $P(\vec{a}) \in S$  (here,  $P(\vec{a})$  is treated as a propositional atom);
- (2) For each element  $a \in \text{Dom}(\mathcal{M}')$ , propositional atom  $a = a$  is in  $S$  (note that  $\text{Dom}(\mathcal{M}) = \text{Dom}(\mathcal{M}')$ );
- (3)  $S$  does not contain any other atoms.

Clearly, if we view all elements occurring in  $\mathcal{M}'$ 's relations as propositional atoms, we have  $S \cap \mathcal{M} = \mathcal{M}' \cap \mathcal{M} = \mathcal{M}$ .

Now we show that  $S$  is a preferred answer set of  $(Ground(\Pi)_{\mathcal{M}}, <^*)$ . For each rule  $r \in \Pi$ , we denote  $r^T \in \Pi'$  as the tagged rule obtained from  $r$  by adding the corresponding atom  $Tag(\vec{x})$  to  $r$ 's positive body (see earlier definition). Also note that since  $\text{Dom}(\mathcal{M}) = \text{Dom}(\mathcal{M}')$ , each assignment  $\eta$  of  $\mathcal{M}$  is also an assignment of  $\mathcal{M}'$ . Then, from the definition of  $(Ground(\Pi)_{\mathcal{M}}, <^*)$ , we can see that for each  $r \in \Pi$ , assignment  $\eta$  of  $\mathcal{M}$  and  $t$  ( $t \geq 0$ ),  $(r, \eta) \in \Gamma^t(\Pi)_{\mathcal{M}}$  iff  $r^T \eta \in \Delta^t(Ground(\Pi)_{\mathcal{M}})_S$ . On the other hand, since the only extra rules in  $\Delta^\infty(Ground(\Pi)_{\mathcal{M}})_S$  are those of the forms  $Tag_i(\vec{a}) \leftarrow$  and  $a = a \leftarrow$ , while  $S$  contains all atoms of the forms  $Tag_i(\vec{a})$  and  $a = a$ ,  $S$  is exactly the set  $\{Head(r^T \eta) \mid r^T \eta \in \Delta^\infty(Ground(\Pi)_{\mathcal{M}})_S\} \cup \{Tag_i(\vec{a}), a = a \mid \text{for all corresponding } i\text{'s, } \vec{a}\text{'s and } a\text{'s}\}$ . This means that  $S$  is a preferred answer set of  $(Ground(\Pi)_{\mathcal{M}}, <^*)$ .

The other direction can be proved with similar arguments.  $\square$

## 4.2. Semantic Properties

In this subsection, we study several specific properties of preferred answer set semantics. As we will see, the grounded preferred answer set semantics provides a basis for our investigation. We first define the notion of generating rules. Let  $(\Pi, <)$  be a grounded preferred answer set program and  $S$  a set of propositional atoms. We say that a rule  $r \in \Pi$  is a *generating rule* of  $S$  if  $Pos(r) \subseteq S$  and  $Neg(r) \cap S = \emptyset$ . Now consider  $(\Pi, <)$  to be an FO preferred answer set program and  $\mathcal{M}$  a structure of  $\tau(\Pi)$ , then we say that a rule  $r \in \Pi$  is a *generating rule* of  $\mathcal{M}$  under the assignment  $\eta$  if  $\mathcal{M} \models \widehat{Pos(r)}\eta \wedge \neg \widehat{Neg(r)}\eta$ , where  $\widehat{Pos(r)}$  and  $\neg \widehat{Neg(r)}$  denote the formulas  $\beta_1 \wedge \dots \wedge \beta_k$  and  $\neg \gamma_1 \wedge \dots \wedge \neg \gamma_k$ , respectively.

**LEMMA 4.8.** *Let  $(\Pi, <)$  be a grounded preferred answer set program and  $S$  an answer set of  $\Pi$ . Then the following two statements are equivalent:*

- (1)  $S$  is a preferred answer set of  $(\Pi, <)$ .
- (2) For each rule  $r \in \Pi$ ,  $r \in \Delta^\infty(\Pi)_S$  iff  $r$  is a generating rule of  $S$ .

PROOF. ( $\Rightarrow$ ) Suppose  $S$  is a preferred answer set of  $(\Pi, <)$ . If  $r \in \Delta^\infty(\Pi)_S$ , then according to Definition 4.5, we know that  $Pos(r) \subseteq Head(\Delta^\infty(\Pi)_S)$  and  $Neg(r) \cap S = \emptyset$ . Since  $Head(\Delta^\infty(\Pi)_S) = S$ , this implies that  $r$  is a generating rule of  $S$ .

On the other hand, suppose that  $r$  is a generating rule of  $S$ . We will show that  $r \in \Delta^\infty(\Pi)_S$ . We prove this result by induction on the sequence of rules under ordering  $<$ . First, consider any rule  $r \in S$  where there does not exist any other rule  $r'$  in  $\Pi$  such that  $r' < r$ . According to Definition 4.5,  $r \in \Delta^0(\Pi)_S$  if  $Pos(r) = \emptyset$ , otherwise there exists some  $k$  such that  $r \in \Delta^k(\Pi)_S$ . Now we assume that for all generating rules  $r'$  of  $S$  such that  $r' < r$ ,  $r' \in \Delta^\infty(\Pi)_S$ . Suppose that  $r \notin \Delta^\infty(\Pi)_S$ . Then for all  $t$ , we have that  $r \notin \Delta^t(\Pi)_S$ . That is, for all  $t$ , either (1)  $Pos(r) \not\subseteq Head(\Delta^{t-1}(\Pi)_S)$  or  $Neg(r) \cap S \neq \emptyset$ ; or (2) there exists some  $r' \in \Pi$  such that  $r' < r$ ,  $r' \notin \Delta^{t-1}(\Pi)_S$ ,  $Pos(r') \subseteq S$  and  $Neg(r') \cap Head(\Delta^{t-1}(\Pi)_S) = \emptyset$ . Since  $r$  is a generating rule of  $S$  and  $S = Head(\Delta^\infty(\Pi)_S)$ , it is obvious that case (1) cannot occur. So it has to be case (2). In this case, we can select a sufficient large  $t$  such that for any other  $t'$  where  $t' > (t-1)$ , no more rules from  $\Pi$  can be added into  $\Delta^{t'}(\Pi)_S$ , that is,  $\Delta^{t-1}(\Pi)_S = \Delta^{t'}(\Pi)_S$  for all  $t' > (t-1)^3$ . Therefore, for this particular  $t$ , we can find some  $r' \in \Pi$  such that  $r' < r$  and  $r' \notin \Delta^{t-1}(\Pi)_S$ . Since  $Pos(r') \subseteq S$  and  $Neg(r') \cap Head(\Delta^{t-1}(\Pi)_S) = Neg(r') \cap Head(\Delta^\infty(\Pi)_S) = Neg(r') \cap S = \emptyset$ , it follows that  $r'$  is a generating rule of  $S$ . Since according to our inductive hypothesis, we had that  $r' \in \Delta^\infty(\Pi)_S$ , then this is a contradiction.

( $\Leftarrow$ ) For each  $r \in \Delta^\infty(\Pi)_S$ ,  $r$  is a generating rule of  $S$ , so  $Pos(r) \subseteq S$  and  $Neg(r) \cap S = \emptyset$ . Also, since  $S$  is an answer set of  $\Pi$ , it follows  $Head(r) \in S$ . So  $Head(\Delta^\infty(\Pi)_S) \subseteq S$ . Now we show  $S \subseteq Head(\Delta^\infty(\Pi)_S)$ . Consider any  $r \in \Pi$  such that  $Head(r) \in S$ . Since  $S$  is an answer set of  $\Pi$ , there must exist a rule  $r' \in \Pi$  such that  $Head(r) = Head(r')$ ,  $Pos(r') \subseteq S$  and  $Neg(r') \cap S = \emptyset$ . This means that  $r'$  is a generating rule of  $S$ . From the condition, we know that  $r' \in \Delta^\infty(\Pi)_S$ . It follows  $Head(r) = Head(r') \in Head(\Delta^\infty(\Pi)_S)$ . That is  $S \subseteq Head(\Delta^\infty(\Pi)_S)$ . Therefore, we have  $S = Head(\Delta^\infty(\Pi)_S)$ . So  $S$  is an answer set of  $(\Pi, <)$ .  $\square$

Now we have the following semantic characterization theorem for preferred first-order logic programs.

**THEOREM 4.9.** *Let  $(\Pi, <)$  be an FO preferred answer set program and  $\mathcal{M}$  an answer set of  $\Pi$ . Then, the following two statements are equivalent:*

- (1)  $\mathcal{M}$  is a preferred answer set of  $(\Pi, <)$ .
- (2) For each rule  $r \in \Pi$ ,  $(r, \eta) \in \Gamma^\infty(\Pi)_\mathcal{M}$  iff  $r$  is a generating rule of  $\mathcal{M}$  under  $\eta$ .

PROOF. For the given  $(\Pi, <)$  and its answer set  $\mathcal{M}$ ,  $(\Pi, <)$ 's grounded preferred program is denoted as  $(Ground(\Pi)_\mathcal{M}, <^*)$  as defined in Definition 4.3. From the proof of Theorem 4.7, we can then construct a set of propositional atoms  $S$  from  $\mathcal{M}$  such that  $S$  is an answer set of  $(Ground(\Pi)_\mathcal{M}, <^*)$  and  $\mathcal{M} = \mathcal{M} \cap S$ . Then, based on Lemma 4.8, in order to prove this theorem, it is sufficient to prove the following two results:

Result 1:  $(r, \eta) \in \Gamma^\infty(\Pi)_S$  iff  $r^* \in \Delta^\infty(\Pi)_S$ , where

$$r^* : Head(r)\eta \leftarrow Body(r)\eta, Tag(\vec{x})\eta,$$

Result 2:  $r$  is a generating rule of  $\mathcal{M}$  under  $\eta$  iff  $r^*$  is a generating rule of  $S$ .

From the definitions of  $\Gamma^\infty(\Pi)_\mathcal{M}$ ,  $\Delta^\infty(\Pi)_S$  and  $(Ground(\Pi)_\mathcal{M}, <^*)$ , together with Theorem 4.7, Results 1 and 2 are easily to be proved.  $\square$

<sup>3</sup>Note that from the grounded preferred logic program definition (i.e., Definition 4.3) and the fact that  $S$  is a preferred answer set of  $(\Pi, <)$ , this is always possible, because there must exist a finite number  $N$  such that for all  $t > N$ ,  $\Delta^t(\Pi)_S = \Delta^{t+1}(\Pi)_S$ .

**PROPOSITION 4.10.** *For FO preferred answer set programs  $(\Pi, <_1)$  and  $(\Pi, <_2)$  where  $<_1 \subseteq <_2$ , and a structure  $\mathcal{M}$  of  $\tau(\Pi)$ , if  $\mathcal{M}$  is a preferred answer set of  $(\Pi, <_2)$ , then  $\mathcal{M}$  is also a preferred answer set of  $(\Pi, <_1)$ .*

**PROOF.** By Theorem 4.7, it is sufficient to prove the propositional version of the result. So we can assume that  $(\Pi, <_1)$  and  $(\Pi, <_2)$  are two propositional preferred programs and  $<_1 \subseteq <_2$ . Suppose  $S$  is a preferred answer set of  $(\Pi, <_2)$ . First, from Definition 4.5, it is easy to see that for all  $t$ ,  $\Delta^t(\Pi)_S^{<_2} \subseteq \Delta^t(\Pi)_S^{<_1}$ . Further, since  $S$  is a preferred answer set of  $(\Pi, <_2)$ , we have  $S = \text{Head}(\Delta^\infty(\Pi)_S^{<_2})$ . Now we show that  $S$  must also be a preferred answer set of  $(\Pi, <_1)$ .

To show this, we simply consider a special preferred answer set program  $(\Pi, <_0)$ , where  $<_0 = \emptyset$ . So we have  $\Delta^t(\Pi)_S^{<_1} \subseteq \Delta^t(\Pi)_S^{<_0}$  for all  $t$ . So we have  $\text{Head}(\Delta^\infty(\Pi)_S^{<_2}) \subseteq \text{Head}(\Delta^\infty(\Pi)_S^{<_1}) \subseteq \text{Head}(\Delta^\infty(\Pi)_S^{<_0})$ . From the condition and Proposition 3.5, we know that  $\text{Head}(\Delta^\infty(\Pi)_S^{<_2}) = \text{Head}(\Delta^\infty(\Pi)_S^{<_0}) = S$ . So it follows that  $\text{Head}(\Delta^\infty(\Pi)_S^{<_1}) = S$ , that is,  $S$  is also a preferred answer set of  $(\Pi, <_1)$ .  $\square$

Now we consider the existence of preferred answer sets for a preferred program. From Theorem 4.9, in order to see whether a structure  $\mathcal{M}$  is a preferred answer set for a given program, we need to compute  $\Gamma^\infty(\Pi)_\mathcal{M}$  and then to check all generating rules against  $\mathcal{M}$ . It is always desirable to discover some stronger sufficient conditions for the existence of preferred answer sets, by which there is no need to undertake the computation of  $\Gamma^\infty(\Pi)_\mathcal{M}$ . The following lemma and theorem are towards this purpose.

**LEMMA 4.11.** *A grounded preferred answer set program  $(\Pi, <)$  has a preferred answer set if there exists an answer set  $S$  of  $\Pi$  such that for each  $(r_1, r_2) \in <$  and  $\text{Head}(r_2) \cap (\text{Pos}(r_1) \cup \text{Neg}(r_1)) \neq \emptyset$ ,  $r_2$  is not a generating rule of  $S$ .*

**PROOF.** We show that  $S$  is a preferred answer set of  $(\Pi, <)$ . We first prove that if  $r \in \Delta^\infty(\Pi)_S$ , then  $r$  is a generating rule of  $S$ . We prove this by induction on  $t$ . Consider  $\Delta^0(\Pi)_S$ . Since for all  $r \in \Delta^0(\Pi)_S$ ,  $\text{Pos}(r) = \emptyset$  and  $\text{Neg}(r) \cap S = \emptyset$ . This means that  $r$  is a generating rule of  $S$ . Since  $S$  is an answer set of  $\Pi$ , this implies  $\text{Head}(r) \in S$ . Suppose for all  $t$  that  $r \in \Delta^t(\Pi)_S$  implies that  $r$  is a generating rule of  $S$ , which implies  $\text{Head}(\Delta^t(\Pi)_S) \subseteq S$ . Now we consider  $\Delta^{t+1}(\Pi)_S$ . According to the definition, if  $r \notin \Delta^t(\Pi)_S$  but  $r \in \Delta^{t+1}(\Pi)_S$ , then  $\text{Pos}(r) \subseteq \text{Head}(\Delta^t(\Pi)_S)$  and  $\text{Neg}(r) \cap S = \emptyset$ . From the induction hypothesis, we have  $\text{Pos}(r) \subseteq \text{Head}(\Delta^t(\Pi)_S) \subseteq S$ . So,  $r$  is also a generating rule of  $S$ , and hence  $\text{Head}(r) \in S$ .

Now we show that under the given conditions of this lemma, if  $r$  is a generating rule of  $S$ , then  $r \in \Delta^\infty(\Pi)_S$ . We prove this by induction on  $<$ . First, suppose that there does not exist a rule  $r^* \in \Pi$  such that  $r^* < r$  and  $r^*$  is a generating rule of  $S$ . Since  $r$  is a generating rule and  $S$  is an answer set of  $\Pi$ , it must be the case that  $\text{Pos}(r) = \emptyset$  and hence, that  $r \in \Delta^0(\Pi)_S$ . Now we assume that for all generating rules  $r^*$  such that  $r^* < r$  and those  $r^*$  satisfying the condition of this lemma, that  $r^* \in \Delta^\infty(\Pi)_S$ . Now we consider  $r$ . Since for any  $r'$  where  $r < r'$  and  $\text{Head}(r') \cap (\text{Pos}(r') \cup \text{Neg}(r')) \neq \emptyset$ , we have that  $r'$  is not a generating rule of  $S$ , then there must exist generating rules  $r_1, \dots, r_k$  of  $S$  such that  $r_i < r$  ( $i = 1, \dots, k$ ),  $\text{Pos}(r) \subseteq \bigcup_{i=1}^k \text{Head}(r_i)$ , and  $\text{Neg}(r) \cap \bigcup_{i=1}^k \text{Head}(r_i) = \emptyset$ . Now, according to the induction hypothesis, these  $r_1, \dots, r_k$  are in  $\Delta^\infty(\Pi)_S$ . So there exists some certain  $t$  for which we have  $\text{Pos}(r) \subseteq \text{Head}(\Delta^t(\Pi)_S)$  and  $\text{Neg}(r) \cap S = \emptyset$  (this is due to the fact that  $r$  is a generating rule of  $S$ ). Therefore, from the definition of  $\Delta^\infty(\Pi)_S$  (see Definition 4.5), we know that if  $r \notin \Delta^\infty(\Pi)_S$ , then for all  $t$ , there must exist a rule  $r'$  such that  $r' < r$ ,  $r' \notin \Delta^{t-1}(\Pi)_S$ , and  $\text{Pos}(r') \subseteq S$  and  $\text{Neg}(r') \cap \text{Head}(\Delta^{t-1}(\Pi)_S)$ . By selecting a sufficient large  $t$ , we would have  $\Delta^t(\Pi)_S = \Delta^\infty(\Pi)_S$ . This implies that there exists some  $r'$  such that  $r' < r$ ,  $r' \notin \Delta^\infty(\Pi)_S$ , and  $r'$  is a generating rule of  $S$ . This is in contradiction with our inductive hypothesis. So  $r$  must be in  $\Delta^t(\Pi)_S$  for some  $t$ . That is,  $r \in \Delta^\infty(\Pi)_S$ .



Finally, from Lemma 4.8, we know that  $S$  is also a preferred answer set of  $(\Pi, <)$ . This completes our proof.  $\square$

**THEOREM 4.12.** *An FO preferred answer set program  $(\Pi, <)$  has a preferred answer set if there exists an answer set  $\mathcal{M}$  of  $\Pi$  such that for each  $(r_1, r_2) \in <$  and for all assignments  $\eta, \eta'$  of  $\mathcal{M}$  satisfying  $\text{Head}(r_2)\eta' \cap (\text{Pos}(r_1) \cup \text{Neg}(r_1))\eta \neq \emptyset$ ,  $r_2$  is not a generating rule of  $\mathcal{M}$  under  $\eta'$ .*

**PROOF.** We prove the result by showing that the given  $\mathcal{M}$  is a preferred answer set of  $(\Pi, <)$ . For the given preferred answer set program  $(\Pi, <)$  and an answer set  $\mathcal{M}$  of  $\Pi$ , we consider the grounded preferred answer set program  $(\text{Ground}(\Pi)_{\mathcal{M}}, <^*)$ . From Theorem 4.7, we know that there is a preferred answer set  $S$  of  $(\text{Ground}(\Pi)_{\mathcal{M}}, <^*)$  where  $S \cap \mathcal{M} = \mathcal{M}$ . We can also show that for each pair of rules  $r_1, r_2 \in \Pi$  where  $r_1 < r_2$  and assignments  $\eta, \eta'$  of  $\mathcal{M}$  such that  $\text{Head}(r_2)\eta' \cap (\text{Pos}(r_1) \cup \text{Neg}(r_1))\eta \neq \emptyset$ , there exists a corresponding pair of rules  $r_1^*, r_2^* \in \text{Ground}(\Pi)$  such that  $r_1^* <^* r_2^*$  and  $\text{Head}(r_2^*) \cap (\text{Pos}(r_1^*) \cup \text{Neg}(r_1^*)) \neq \emptyset$ , and *vice versa*. We can further prove that a rule  $r \in \Pi$  is not a generating rule of  $\mathcal{M}$  under  $\eta$  iff there is a corresponding rule  $r^* \in \text{Ground}(\Pi)$  which is not a generating rule of  $S$ . Then, from Lemma 4.11, it follows that  $S$  is a preferred answer set of  $(\text{Ground}(\Pi), <^*)$ . Finally, from Theorem 4.7,  $\mathcal{M}$  is also a preferred answer set of  $(\Pi, <)$ .  $\square$

It is clear that the sufficient condition represented in Theorem 4.12 is semantics based in the sense that an answer set (answer sets) of  $\Pi$  has (have) to be computed and verified to decide the existence of a preferred answer set. Nevertheless, from Theorem 4.12, we actually can derive a sufficient condition which is more syntactically oriented and is quite effective in deciding the existence of preferred answer sets. To make our claim precise, we first introduce a useful notion.

A *substitution*  $\theta$  is of the form  $(x_1/t_1, \dots, x_k/t_k)$ , where  $x_i$  is a variable and  $t_i$  is a term for all  $i = 1, \dots, k$ . By applying a substitution  $\theta$  on a set  $X$  of first-order formulas, we obtain another set of formulas, denoted as  $X\theta$ , by replacing each variable  $x$  occurring in formulas of  $X$  with term  $t$  whenever  $x/t \in \theta$ .

**Definition 4.13.** Let  $P(\vec{t})$  be an atom,  $X$  a finite set of atoms,  $T_1$  and  $T_2$  two finite sets of formulas of the forms  $t_1 = t_2$  and  $t_1 \neq t_2$  where  $\vec{t}$  is a tuple of terms, and  $t_1$  and  $t_2$  are terms. We call  $T_1$  and  $T_2$  the sets of *term binding constraints* imposing on  $P(\vec{t})$  and  $X$ , respectively. The pair  $(P(\vec{t}), X)$  is called *unifiable* under  $T_1$  and  $T_2$  if there exists a substitution  $\theta$  such that  $P(\vec{t}) \in X\theta$  and  $(X \cup T_2)\theta \cup T_1$  is consistent.

**Example 4.14.** Let us consider an atom  $P(x, y)$ , set  $X = \{P(x', y'), Q(y', z')\}$ , and their term constraint sets  $T_1 = \{x = y\}$  and  $T_2 = \{x' \neq z', y' = c\}$  respectively. It is easy to see that  $P(x, y)$  and  $X$  are unifiable under  $T_1$  and  $T_2$  because we can find a substitution  $\theta = (x'/x, y'/y)$ , such that  $(X \cup T_2)\theta = \{P(x, y), Q(y, z'), x \neq z', y = c\}$ , and  $\{P(x, y), Q(y, z'), x \neq z', y = c\} \cup \{x = y\}$  is consistent.

For a given FO program  $\Pi$  and a rule  $r \in \Pi$ , we specify  $r$ 's set of term binding constraints as follows:

$$T(r) = \{t_1 = t_2 \mid t_1 = t_2 \in \text{Pos}(r)\} \cup \{t_1 \neq t_2 \mid t_1 = t_2 \in \text{Neg}(r)\}.$$

We also use  $\text{Pos}(r)^{-e}$  and  $\text{Neg}(r)^{-e}$  to denote the sets of all atoms of  $\text{Pos}(r)$  and  $\text{Neg}(r)$  except the equality atoms respectively.

**THEOREM 4.15.** *Let  $(\Pi, <)$  be an FO preferred answer set program. Then,  $(\Pi, <)$  has a preferred answer set if  $\Pi$  has an answer set and for each  $r_1 < r_2$ , neither  $(\text{Head}(r_2), \text{Pos}(r_1)^{-e})$  nor  $(\text{Head}(r_2), \text{Neg}(r_1)^{-e})$  is unifiable under  $T(r_2)$  and  $T(r_1)$ .*

PROOF. For any answer set  $\mathcal{M}$  of  $\Pi$  and any two assignments  $\eta$  and  $\eta'$  of  $\mathcal{M}$ , we have either (1)  $Head(r_2)\eta' \cap (Pos(r_1) \cup Neg(r_1))\eta = \emptyset$ , or (2)  $Head(r_2)\eta' \cap (Pos(r_1) \cup Neg(r_1))\eta \neq \emptyset$ . Case (1) implies that the condition of Theorem 4.12 holds. So under such a situation, each answer set  $\mathcal{M}$  of  $\Pi$  is actually a preferred answer set of  $(\Pi, <)$ . For case (2), since neither  $(Head(r_2), Pos(r_1)^{-e})$  nor  $(Head(r_2), Neg(r_1)^{-e})$  is unifiable under  $T(r_2)$  and  $T(r_1)$ , respectively, it is clear that  $r_1$  and  $r_2$  cannot be generating rules of  $\mathcal{M}$  under  $\eta$  and  $\eta'$ , respectively, at the same time. We consider the corresponding grounded preferred program  $(Ground(\Pi)_{\mathcal{M}}, <^*)$ , where we assume that  $S$  is an answer set of  $Ground(\Pi)_{\mathcal{M}}$  (see Theorem 4.7). Obviously we have  $r_1\eta <^* r_2\eta'$ . Consequently, we know that  $r_1\eta$  and  $r_2\eta'$  cannot be generating rules of  $S$  at the same time. If  $r_1\eta$  is not a generating rule of  $S$ , then relation  $r_1\eta <^* r_2\eta'$  will not play any role in program  $(Ground(\Pi)_{\mathcal{M}}, <^*)$ . On the other hand, if  $r_2\eta'$  is not a generating rule, by Lemma 4.11,  $S$  is a preferred answer set of  $(Ground(\Pi)_{\mathcal{M}}, <^*)$ . By Theorem 4.7,  $\mathcal{M}$  is also an answer set of  $(\Pi, <)$ .  $\square$

## 5. LOGICAL CHARACTERIZATIONS

Since the answer set semantics for FO programs is defined via an SO sentence, it is a natural question to ask whether it is also possible to characterize the progression-based semantics for preferred FO programs through an SO sentence. In this section, we will study this issue in detail. Our basic idea is that: we first propose an alternative SO sentence that precisely captures the answer set semantics of FO programs. Our formalization will be different from that of the original one as presented in Ferraris et al. [2011] for the case when restricted to normal logic programs. Instead, our SO formalism will be developed based on the intuition of the progression-based semantics for first-order normal answer set programs [Zhang and Zhou 2010]. Then we extend such an SO formalism by taking the preference into account.

To begin with, we first introduce some useful notions. We suppose that in the rest of this article, each program  $\Pi$  is presented in a *normalized form*. That is, we assume that each proper atom occurring in a rule only contains a tuple of distinguishable variables. Note that every program can be rewritten in such a normalized form. For instance, if  $P(x_1, \dots, x_{i-1}, c, x_{i+1}, \dots, x_n)$  occurs in some rule  $r$  where  $c$  is a constant, then we simply replace  $P(x_1, \dots, x_{i-1}, c, x_{i+1}, \dots, x_n)$  by  $P(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)$  and then add  $x_i = c$  into  $r$ 's positive body. In this way, we can simply write  $\vec{x}_P$  and  $\vec{x}_r$  as the tuples of all distinguishable variables occurring in predicate  $P$  and rule  $r$  respectively. Now consider a program consisting of the single rule

$$r : S(x, y) \leftarrow E(a, y), x = a.$$

Then this program can be equivalently rewritten to a normalized form

$$r : S(x, y) \leftarrow E(z, y), x = a, z = a,$$

where  $\vec{x}_r = xyz$ . Now let  $\vec{x} = x_1 \dots x_n$  and  $\vec{y} = y_1 \dots y_n$ , then we also write  $\vec{x} = \vec{y}$  to denote the formula  $\bigwedge_{1 \leq i \leq n} x_i = y_i$ .

For easier presentation and readability of formulas, we further assume that for any two rules  $r_1$  and  $r_2$  where  $r_1 \neq r_2$  (that is, distinct rules),  $\vec{x}_{r_1}$  and  $\vec{x}_{r_2}$  are disjoint from each other. That is, the names of the variables in  $r_1$  do not occur in  $r_2$  (and vice versa). Note that in cases where we have to refer to the tuples  $\vec{x}_{r_1}$  and  $\vec{x}_{r_2}$  in a formula for which they have to be disjoint, we can always relabel the variables in  $\vec{x}_{r_2}$  without explicitly stating it, when clear from the context.

### 5.1. Formulas Representing Generating Rules and Program Completion

Similar to what we introduced in Section 4.2, here we present an FO formula to represent the notion of *generating rules* in terms of a structure. In particular, given a

program  $\Pi$  and a rule  $r \in \Pi$  of the form (1):  $\alpha \leftarrow \beta_1, \dots, \beta_k, \text{not } \gamma_1, \dots, \text{not } \gamma_l$ , we specify an FO formula

$$\varphi_r^{\text{GEN}}(\vec{x}_r) = \widehat{Pos(r)} \wedge \neg \widehat{Neg(r)}. \quad (4)$$

Clearly, for a given  $\tau(\Pi)$ -structure  $\mathcal{M}$ ,  $\mathcal{M} \models \varphi_r^{\text{GEN}}(\vec{x}_r)$  iff  $Pos(r)\eta \subseteq \mathcal{M}$  and  $Neg(r)\eta \cap \mathcal{M} = \emptyset$ .

*Example 5.1.* Consider the program  $\Pi_5$  consisting of the following rules:

$$\begin{aligned} r_1 : S(x_1, y_1) &\leftarrow E(z_1, y_1), x_1 = a, z_1 = a, \\ r_2 : S(x_2, z_2) &\leftarrow S(x_2, y_2), E(y_2, z'_2), z_2 = a, z'_2 = b. \end{aligned}$$

Then we have

$$\begin{aligned} \varphi_{r_1}^{\text{GEN}}(x_1, y_1, z_1) &= E(z_1, y_1) \wedge x_1 = a \wedge z_1 = a, \\ \varphi_{r_2}^{\text{GEN}}(x_2, y_2, z_2, z'_2) &= S(x_2, y_2) \wedge E(y_2, z'_2) \wedge z_2 = a \wedge z'_2 = b. \end{aligned}$$

Let  $\Pi$  be an FO program. We define the FO sentence  $\varphi_{\Pi}^{\text{COMP}}$  to be the completion of  $\Pi$ ,

$$\varphi_{\Pi}^{\text{COMP}} = \bigwedge_{P \in \mathcal{P}_{\text{int}}(\Pi)} \forall \vec{x}_P (P(\vec{x}_P) \leftrightarrow \bigvee_{\substack{r \in \Pi, \\ \text{Head}(r) = P(\vec{y}_P)}} \exists \vec{x}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \wedge \vec{x}_P = \vec{y}_P)), \quad (5)$$

where we assume that  $\vec{x}_P$  is a tuple of distinguishable variables disjoint from  $\vec{y}_P$ .

## 5.2. Well-Orderings on Generating Rules in Terms of Structures

Given an FO program  $\Pi$  and a  $\tau(\Pi)$ -structure  $\mathcal{M}$ , let  $\Gamma(\Pi)_{\mathcal{M}}^4$  denote the set:

$$\{(r, \eta) \mid r \in \Pi, Pos(r)\eta \subseteq \mathcal{M} \text{ and } Neg(r)\eta \cap \mathcal{M} = \emptyset\},$$

that is, those generating rules under the structure  $\mathcal{M}$ . Then a *well-order* on  $\Gamma(\Pi)_{\mathcal{M}}$  is a structure  $\mathcal{W} = (\Gamma(\Pi)_{\mathcal{M}}, <^{\mathcal{W}})$  with domain  $\Gamma(\Pi)_{\mathcal{M}}$  and binary relation  $<^{\mathcal{W}}$  on  $\Gamma(\Pi)_{\mathcal{M}}$  that satisfies the following properties:

- (1)  $x, y \in \Gamma(\Pi)_{\mathcal{M}}$  and  $x \neq y$  implies  $x <^{\mathcal{W}} y$  or  $y <^{\mathcal{W}} x$  (*totality*);
- (2)  $x <^{\mathcal{W}} y$  and  $y <^{\mathcal{W}} z$  implies  $x <^{\mathcal{W}} z$  (*transitivity*);
- (3)  $x <^{\mathcal{W}} y$  implies  $y \not<^{\mathcal{W}} x$  (*asymmetry*);
- (4) and last, the SO axiom:

$$\forall S (S \neq \emptyset \rightarrow (\exists x \in S)(\forall y \in S)(x \neq y \rightarrow x <^{\mathcal{W}} y)),$$

which expresses that every nonempty subset  $S$  of  $\Gamma(\Pi)_{\mathcal{M}}$  has a least element.

Note that when  $\Gamma(\Pi)_{\mathcal{M}}$  is finite, any *strict total-order* of  $\Gamma(\Pi)_{\mathcal{M}}$  is trivially a well-order, while this is not the case when  $\Gamma(\Pi)_{\mathcal{M}}$  is infinite. In fact, since *totality*, *transitivity*, and *asymmetry* is what corresponds to a strict total-order, when only considering finite structures, we can drop the SO axiom corresponding to the least element property of each nonempty subset.

## 5.3. Formalizing Progression

Now we propose an SO sentence that will simulate the progression semantics we defined earlier but without taking preference into account.

<sup>4</sup>Note that  $\Gamma(\Pi)_{\mathcal{M}}$  differs from  $\Gamma^l(\Pi)_{\mathcal{M}}$  in that  $\Gamma(\Pi)_{\mathcal{M}}$  does not represent a progression stage.

*Definition 5.2.* Given an FO answer set program  $\Pi$ , we define an SO formula  $\varphi_{\Pi}^{\text{PRO}}(\vec{\succ}, \vec{S})$  as follows (here “PRO” stands for *progression*)<sup>5</sup>:

$$\bigwedge_{r_1, r_2, r_3 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} \vec{x}_{r_3} (\widetilde{\prec}_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \wedge \widetilde{\prec}_{r_2 r_3}(\vec{x}_{r_2}, \vec{x}_{r_3}) \rightarrow \widetilde{\prec}_{r_1 r_3}(\vec{x}_{r_1}, \vec{x}_{r_3})) \quad (6)$$

$$\wedge \bigwedge_{r_1, r_2 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} (\widetilde{\prec}_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \rightarrow \neg \widetilde{\prec}_{r_2 r_1}(\vec{x}_{r_2}, \vec{x}_{r_1})) \quad (7)$$

$$\wedge \bigwedge_{r_1, r_2 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} (\widetilde{\prec}_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \rightarrow \varphi_{r_1}^{\text{GEN}}(\vec{x}_{r_1}) \wedge \varphi_{r_2}^{\text{GEN}}(\vec{x}_{r_2})) \quad (8)$$

$$\wedge \bigwedge_{r \in \Pi} \forall \vec{x}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \varphi_r^{\text{SUP}}(\vec{\succ}, \vec{x}_r)) \quad (9)$$

$$\wedge \varphi_{\Pi}^{\text{WELLOR}}(\vec{\succ}, \vec{S}), \quad (10)$$

where:

- for each  $r_1, r_2 \in \Pi$ , the symbol  $\widetilde{\prec}_{r_1 r_2}$  is a predicate variable of arity  $|\vec{x}_{r_1}| + |\vec{x}_{r_2}|$ <sup>6</sup>;
- $\vec{\succ}$  denotes the distinguishable tuple of predicate variables of the set  $\{\widetilde{\prec}_{r_1 r_2} \mid r_1, r_2 \in \Pi\}$ ;
- $\varphi_r^{\text{SUP}}(\vec{\succ}, \vec{x}_r)$  denotes the following formula (here “SUP” stands for *support*):

$$\bigwedge_{\substack{P(\vec{x}_P) \in \text{Pos}(r), \\ P \in \text{Pos}(\Pi)}} \bigvee_{\substack{r' \in \Pi, \\ \text{Head}(r') = P(\vec{y}_P)}} \exists \vec{x}_{r'} (\widetilde{\prec}_{r' r}(\vec{x}_{r'}, \vec{x}_r) \wedge \vec{x}_P = \vec{y}_P) \quad (11)$$

(where in the case that  $r' = r$ , we simply assume a relabeling of  $\vec{x}_{r'}$  such that  $\vec{x}_{r'}$  will be disjoint from  $\vec{x}_r$ );

- $\varphi_{\Pi}^{\text{WELLOR}}(\vec{\succ}, \vec{S})$  denotes the following formula (here “WELLOR” stands for *well-ordered*):

$$\begin{aligned} & \bigwedge_{r \in \Pi} \forall \vec{x}_r (\widetilde{S}_r(\vec{x}_r) \rightarrow \varphi_r^{\text{GEN}}(\vec{x}_r)) \wedge \left( \bigvee_{r \in \Pi} \exists \vec{x}_r \widetilde{S}_r(\vec{x}_r) \right) \rightarrow \\ & \bigvee_{r' \in \Pi} \exists \vec{x}_{r'} (\widetilde{S}_{r'}(\vec{x}_{r'}) \wedge \forall \vec{y}_{r'} (\widetilde{S}_{r'}(\vec{y}_{r'}) \wedge \vec{y}_{r'} \neq \vec{x}_{r'} \rightarrow \widetilde{\prec}_{r' r'}(\vec{x}_{r'}, \vec{y}_{r'})) \\ & \wedge \bigwedge_{\substack{r'' \in \Pi, \\ r'' \neq r'}} \forall \vec{x}_{r''} (\widetilde{S}_{r''}(\vec{x}_{r''}) \rightarrow \widetilde{\prec}_{r' r''}(\vec{x}_{r'}, \vec{x}_{r''})), \end{aligned} \quad (12)$$

where  $\vec{S}$  denotes the distinguishable tuple of predicate variables of the set  $\{\widetilde{S}_r \mid r \in \Pi\}$  such that for each  $r \in \Pi$ , the arity of  $\widetilde{S}_r$  is  $|\vec{x}_r|$ , and  $\vec{y}_{r'}$  (in the consequent above) is a relabeling of the distinct variables of  $\vec{x}_{r'}$  such that  $\vec{y}_{r'}$  is now disjoint from  $\vec{x}_{r'}$ .

Let us take a closer look at Definition 5.2. Basically, formula  $\varphi_{\Pi}^{\text{PRO}}(\vec{\succ}, \vec{S})$  imposes a progression-like order on the set of generating rules with respect to a structure  $\mathcal{M}$ :

$$\Gamma(\Pi)_{\mathcal{M}} = \{(r, \eta) \mid r \in \Pi, \text{Pos}(r)\eta \subseteq \mathcal{M} \text{ and } \text{Neg}(r)\eta \cap \mathcal{M} = \emptyset \text{ for an assignment } \eta \text{ of } \mathcal{M}\},$$

<sup>5</sup>In Formulas (6), (7), and (8), in the case that  $r_1 = r_2$ ,  $r_1 = r_3$ , or  $r_2 = r_3$ , we assume a relabeling of  $\vec{x}_{r_1}$ ,  $\vec{x}_{r_2}$ , or  $\vec{x}_{r_3}$  (however appropriate) such that they will now be disjoint from each other.

<sup>6</sup>For clarity, we denote a predicate variable  $P$  with an accent  $\tilde{P}$  to remind the fact that we will be quantifying over these predicates.

which eventually establishes a correspondence to the sequence of progression sets  $\Gamma^0(\Pi)_{\mathcal{M}}, \Gamma^1(\Pi)_{\mathcal{M}}, \dots$ , as we defined in Definition 3.1. We should emphasize that at this stage, no preference is considered.

In particular, Formulas (6) and (7) express the transitive and asymmetric properties respectively, while Formula (8) expresses the condition that the well-order only involves those of generating rules. Moreover, Formula (9) expresses that if some rule is a generating rule (with respect to certain structure and associated assignment), then it must be supported by rules generated in some earlier stages (i.e.,  $\varphi_r^{\text{SUP}}(\vec{z}, \vec{x}_r)$ ). Finally Formula (10) enforces a well-order (i.e.,  $\varphi_{\Pi}^{\text{WELLOR}}(\vec{z}, \vec{S})$ ).

Specifically, (10) is fulfilled by Formula (12), which encodes that each nonempty subset of generating rules has a least element. Indeed, formula  $\bigwedge_{r \in \Pi} \forall \vec{x}_r (\tilde{S}_r(\vec{x}_r) \rightarrow \varphi_r^{\text{GEN}}(\vec{x}_r))$  encodes that the extents of each  $\tilde{S}_r$  for  $r \in \Pi$  are only those of generating rules, and  $\bigvee_{r \in \Pi} \exists \vec{x}_r \tilde{S}_r(\vec{x}_r)$  encodes that at least one of these subsets is nonempty, while the consequence

$$\begin{aligned} & \bigvee_{r' \in \Pi} \exists \vec{x}_{r'} (\tilde{S}_{r'}(\vec{x}_{r'}) \wedge \forall \vec{y}_{r''} (\tilde{S}_{r''}(\vec{y}_{r''}) \wedge \vec{y}_{r''} \neq \vec{x}_{r'} \rightarrow \widetilde{<}_{r'r''}(\vec{x}_{r'}, \vec{y}_{r'})) \\ & \wedge \bigwedge_{\substack{r'' \in \Pi, \\ r'' \neq r'}} \forall \vec{x}_{r''} (\tilde{S}_{r''}(\vec{x}_{r''}) \rightarrow \widetilde{<}_{r'r''}(\vec{x}_{r'}, \vec{x}_{r''})), \end{aligned}$$

encodes the existence of the least element.

**PROPOSITION 5.3.** *The SO formula  $\varphi_{\Pi}^{\text{PRO}}(\vec{z}, \vec{S})$  is of length  $O(n^3 + mn^2)$  where  $m = \|\text{ATOMS}(\Pi)\|$  (i.e., all the atoms occurring in  $\Pi$ ) and  $n = \|\Pi\|$ .*

**PROOF.** Formula (6) is of length  $O(n^3)$ , (8) is of length  $O(n^2)$ , (9) is of length  $O(mn^2)$ , and (10) is of length  $O(n(m + mn))$ .  $\square$

*Example 5.4.* Consider program  $\Pi_6$  that computes the transitive closure of a binary relation  $E$ :

$$\begin{aligned} r_1 &: T(x_1, y_1) \leftarrow E(x_1, y_1), \\ r_2 &: T(x_2, z_2) \leftarrow T(x_2, y_2), E(y_2, z_2). \end{aligned}$$

Then:

$$\begin{aligned} \varphi_{\Pi_6}^{\text{COMP}} &= \forall x y (T(x, y) \leftrightarrow \exists x_1 y_1 (\varphi_{r_1}^{\text{GEN}}(x_1, y_1) \wedge x = x_1 \wedge y = y_1) \\ & \vee \exists x_2 y_2 z_2 (\varphi_{r_2}^{\text{GEN}}(x_2, y_2, z_2) \wedge x = x_2 \wedge y = z_2)), \end{aligned}$$

where:

$$\begin{aligned} \varphi_{r_1}^{\text{GEN}}(x_1, y_1) &= E(x_1, y_1), \\ \varphi_{r_2}^{\text{GEN}}(x_2, y_2, z_2) &= T(x_2, y_2) \wedge E(y_2, z_2). \end{aligned}$$

In particular, we also have that:

$$\begin{aligned} \varphi_{r_1}^{\text{SUP}}(\vec{z}, x_1, y_1) &= \top, \\ \varphi_{r_2}^{\text{SUP}}(\vec{z}, x_2, y_2, z_2) &= \exists x_3 y_3 z_3 (\widetilde{<}_{r_2 r_2}(x_3, y_3, z_3, x_2, y_2, z_2) \wedge x_2 = x_3 \wedge y_2 = z_3) \\ & \vee \exists x_1 y_1 (\widetilde{<}_{r_1 r_2}(x_1, y_1, x_2, y_2, z_2) \wedge x_2 = x_1 \wedge y_2 = y_1), \end{aligned}$$

where the tuple  $x_3 y_3 z_3$  is a (disjoint) relabeling of  $x_2 y_2 z_2$  and  $\vec{z}$  is the tuple  $\widetilde{<}_{r_1 r_1} \widetilde{<}_{r_1 r_2}$  of predicate variables.

About the formula  $\varphi_{\Pi_6}^{\text{WELLOR}}(\vec{\succ}, \vec{S})$ , we further have:

$$\begin{aligned} & \forall x_1 y_1 (\widetilde{S}_{r_1}(x_1, y_1) \rightarrow \varphi_{r_1}^{\text{GEN}}(x_1, y_1)) \wedge \forall x_2 y_2 z_2 (\widetilde{S}_{r_2}(x_2, y_2, z_2) \rightarrow \varphi_{r_2}^{\text{GEN}}(x_2, y_2, z_2)) \\ & \wedge (\exists x_1 y_1 \widetilde{S}_{r_1}(x_1, y_1) \vee \exists x_2 y_2 z_2 \widetilde{S}_{r_2}(x_2, y_2, z_2)) \rightarrow \\ & (\exists x_1 y_1 (\widetilde{S}_{r_1}(x_1, y_1) \wedge \forall x y (\widetilde{S}_{r_1}(x, y) \wedge (x_1 \neq x \vee y_1 \neq y) \rightarrow \widetilde{<_{r_1 r_1}}(x_1, y_1, x, y)) \\ & \wedge \forall x_2 y_2 z_2 (\widetilde{S}_{r_2}(x_2, y_2, z_2) \rightarrow \widetilde{<_{r_1 r_2}}(x_1, y_1, x_2, y_2, z_2))) \\ & \vee \exists x_2 y_2 z_2 (\widetilde{S}_{r_2}(x_2, y_2, z_2) \wedge \forall x y z (\widetilde{S}_{r_2}(x, y, z) \wedge (x_2 \neq x \vee y_2 \neq y \vee z_2 \neq z) \rightarrow \\ & \widetilde{<_{r_2 r_2}}(x_2, y_2, z_2, x, y, z)) \\ & \wedge \forall x_1 y_1 (\widetilde{S}_{r_1}(x_1, y_1) \rightarrow \widetilde{<_{r_2 r_1}}(x_2, y_2, z_2, x_1, y_1))), \end{aligned}$$

where  $\vec{S}$  is the tuple  $\widetilde{S}_{r_1} \widetilde{S}_{r_2}$  of predicate variables. Then  $\forall \vec{S} \varphi_{\Pi_6}^{\text{WELLOR}}(\vec{\succ}, \vec{S})$  expresses that each nonempty subset of  $\Gamma(\Pi)_{\mathcal{M}}$  possesses a least element as induced by the relations of the predicates  $\widetilde{<_{r_1 r_1}}$ ,  $\widetilde{<_{r_1 r_2}}$ ,  $\widetilde{<_{r_2 r_1}}$ , and  $\widetilde{<_{r_2 r_2}}$ . Note that the nonempty subsets of  $\Gamma(\Pi)_{\mathcal{M}}$  are implicitly encoded via the universally quantified predicate variables  $\widetilde{S}_{r_1}$  and  $\widetilde{S}_{r_2}$ .

**THEOREM 5.5.** *Given an FO answer set program  $\Pi$  and a  $\tau(\Pi)$ -structure  $\mathcal{M}$ ,  $\mathcal{M}$  is an answer set of  $\Pi$  iff  $\mathcal{M} \models \exists \vec{\succ} \forall \vec{S} \varphi_{\Pi}(\vec{\succ}, \vec{S})$ , where  $\varphi_{\Pi}(\vec{\succ}, \vec{S}) = \varphi_{\Pi}^{\text{PRO}}(\vec{\succ}, \vec{S}) \wedge \varphi_{\Pi}^{\text{COMP}}$ .*

**PROOF.** To prove this theorem, we will use a result from [Zhang and Zhou 2010] that  $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$  iff  $\mathcal{M}$  is an answer set of  $\Pi$ . Thus, we prove the equivalent statement:  $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$  iff  $\mathcal{M} \models \exists \vec{\succ} \forall \vec{S} \varphi_{\Pi}(\vec{\succ}, \vec{S})$ .

( $\Rightarrow$ ) Assume  $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$ . For  $t \geq 0$ , define the operator  $(\mathcal{M}^*)^t(\Pi) : 2^{\Sigma(\Pi)_{\mathcal{M}}} \rightarrow 2^{\Sigma(\Pi)_{\mathcal{M}}}$  as

$$(\mathcal{M}^*)^t(\Pi) = \{(r, \eta) \mid \text{Pos}(r)\eta \subseteq \mathcal{M}^{t+1}(\Pi) \text{ and } \text{Neg}(r)\eta \cap \mathcal{M} = \emptyset\}. \quad (13)$$

Then since  $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$  by assumption, we also have that  $(\mathcal{M}^*)^{\infty}(\Pi) = \Gamma(\Pi)_{\mathcal{M}}$ . Now, for  $t \geq 0$ , set  $\Lambda^t(\Pi)_{\mathcal{M}} : 2^{\Sigma(\Pi)_{\mathcal{M}}} \rightarrow 2^{\Sigma(\Pi)_{\mathcal{M}}}$  to be an operator defined inductively as:

$$\begin{aligned} \Lambda^0(\Pi)_{\mathcal{M}} &= (\mathcal{M}^*)^0(\Pi); \\ \Lambda^{t+1}(\Pi)_{\mathcal{M}} &= (\mathcal{M}^*)^{t+1}(\Pi) \setminus (\mathcal{M}^*)^t(\Pi). \end{aligned}$$

Then clearly, we also have  $\bigcup_{t \geq 0} \Lambda^t(\Pi)_{\mathcal{M}} = (\mathcal{M}^*)^{\infty}(\Pi)$ . Moreover, by the definition of  $\Lambda_{\mathcal{M}}^t(\Pi)$ , it is not difficult to see that the sets  $\Lambda^0(\Pi)_{\mathcal{M}}, \Lambda^1(\Pi)_{\mathcal{M}}, \dots, \Lambda^{\infty}(\Pi)_{\mathcal{M}}$ <sup>7</sup> partitions  $(\mathcal{M}^*)^{\infty}(\Pi)$ . Now we construct a *well-ordered* relation  $\mathcal{W} = (\Gamma(\Pi)_{\mathcal{M}}, <^{\mathcal{W}})$  on the set  $\Gamma(\Pi)_{\mathcal{M}}$  as follows:

- (1) For each  $\Lambda_{\mathcal{M}}^t(\Pi)$ , by the *well-ordering theorem* (every set can be well-ordered), there exists a well-ordering on the elements of  $\Lambda_{\mathcal{M}}^t(\Pi)$ . Set such a well-ordering as  $\mathcal{W}_t = (\Lambda_{\mathcal{M}}^t(\Pi), <^{\mathcal{W}_t})$ .
- (2) Then we define the well-order  $\mathcal{W} = (\Gamma(\Pi)_{\mathcal{M}}, <^{\mathcal{W}})$  on  $\Gamma(\Pi)_{\mathcal{M}}$  by setting  $<^{\mathcal{W}} = (\bigcup_{t \geq 0} <^{\mathcal{W}_t}) \cup \{(r_1, \eta_1), (r_2, \eta_2) \mid (r_1, \eta_1) \in \Lambda_{\mathcal{M}}^{t_1}(\Pi), (r_2, \eta_2) \in \Lambda_{\mathcal{M}}^{t_2}(\Pi), \text{ with } t_1 < t_2\}$ .

This simply follows from the fact that the *sum* and *products* of well-ordered types are themselves well-ordered [Enderton 1977].

<sup>7</sup>Here  $\infty$  denotes an arbitrary *order-type*.

Now we denote by  $\forall \vec{S} \varphi_{\Pi}(\vec{S})$  the sentence obtained from  $\forall \vec{S} \varphi_{\Pi}(\vec{\zeta}, \vec{S})$  by treating the existentially quantified predicate variables in  $\vec{\zeta}$  as predicate constants. Then to show  $\mathcal{M} \models \exists \vec{\zeta} \forall \vec{S} \varphi_{\Pi}(\vec{\zeta}, \vec{S})$ , we show that there exists an expansion  $\mathcal{M}'$  of  $\mathcal{M}$  such that  $\mathcal{M}' \models \forall \vec{S} \varphi_{\Pi}(\vec{S})$ . In the following, we denote the treatment of each predicate variable,  $\langle_{r_1 r_2}$ , as a predicate constant by simply denoting it as  $\langle_{r_1 r_2}$  (just removing the “ $\sim$ ” accent). Now set  $\mathcal{M}'$  to be an expansion of  $\mathcal{M}$  such that  $\mathcal{M}'$  is of the extended signature  $\tau(\Pi) \cup \{\langle_{r_1 r_2} \mid r_1, r_2 \in \Pi\}$  of  $\mathcal{M}$  and where each predicate symbol  $\langle_{r_1 r_2}$  (for  $r_1, r_2 \in \Pi$ ) is interpreted by:

$$\langle_{r_1 r_2}^{\mathcal{M}'} = \{ \langle \eta_1(u_1), \dots, \eta_1(u_k), \eta_2(v_1), \dots, \eta_2(v_l) \rangle \mid \langle u_1, \dots, u_k \rangle = \vec{x}_{r_1}, \langle v_1, \dots, v_l \rangle = \vec{x}_{r_2}, \\ \text{and } (r_1, \eta_1) \prec^{\mathcal{W}} (r_2, \eta_2) \}$$

with  $\mathcal{W}$  the well-order on  $\Gamma(\Pi)_{\mathcal{M}}$  as defined earlier. Now, through the following Claims 1 and 2, we show  $\mathcal{M}' \models \forall \vec{S} \varphi_{\Pi}^{\text{PRO}}(\vec{S})$  and  $\mathcal{M}' \models \varphi_{\Pi}^{\text{COMP}}$  respectively, so that  $\mathcal{M}' \models \forall \vec{S} \varphi_{\Pi}(\vec{S})$ .

**CLAIM 1.**  $\mathcal{M}' \models \forall \vec{S} \varphi_{\Pi}^{\text{PRO}}(\vec{S})$ .

**PROOF OF CLAIM 1.** To prove this claim, it is sufficient to show the following:

- (1)  $\mathcal{M}' \models \bigwedge_{r_1, r_2, r_3 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} \vec{x}_{r_3} (\langle_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \wedge \langle_{r_2 r_3}(\vec{x}_{r_2}, \vec{x}_{r_3}) \rightarrow \langle_{r_1 r_3}(\vec{x}_{r_1}, \vec{x}_{r_3}))$ ;
- (2)  $\mathcal{M}' \models \bigwedge_{r_1, r_2 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} (\langle_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \rightarrow \neg \langle_{r_2 r_1}(\vec{x}_{r_2}, \vec{x}_{r_1}))$ ;
- (3)  $\mathcal{M}' \models \bigwedge_{r_1, r_2 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} (\langle_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \rightarrow \varphi_{r_1}^{\text{GEN}}(\vec{x}_{r_1}) \wedge \varphi_{r_2}^{\text{GEN}}(\vec{x}_{r_2}))$ ;
- (4)  $\mathcal{M}' \models \bigwedge_{r \in \Pi} \forall \vec{x}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \varphi_r^{\text{SUP}}(\vec{x}_r))$ ;
- (5)  $\mathcal{M}' \models \forall \vec{S} \varphi_{\Pi}^{\text{WELLOR}}(\vec{S})$ .

Now we show each of the given statements.

- (1)  $\mathcal{M}' \models \bigwedge_{r_1, r_2, r_3 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} \vec{x}_{r_3} (\langle_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \wedge \langle_{r_2 r_3}(\vec{x}_{r_2}, \vec{x}_{r_3}) \rightarrow \langle_{r_1 r_3}(\vec{x}_{r_1}, \vec{x}_{r_3}))$ . Suppose for some assignment  $\alpha$ , we have  $\mathcal{M}' \models \langle_{r_1 r_2}(\vec{a}_{r_1}, \vec{a}_{r_2}) \wedge \langle_{r_2 r_3}(\vec{a}_{r_2}, \vec{a}_{r_3})$  such that  $\vec{a}_{r_1}, \vec{a}_{r_2}$ , and  $\vec{a}_{r_3}$  denotes the tuples obtained from  $\vec{x}_{r_1}, \vec{x}_{r_2}$ , and  $\vec{x}_{r_3}$  respectively by replacing each of the variable  $x$  in  $\vec{x}_r$  with  $\alpha(x)$ . We now show  $\mathcal{M}' \models \langle_{r_1 r_3}(\vec{a}_{r_1}, \vec{a}_{r_3})$ . From the definition of  $\langle_{r_1 r_2}^{\mathcal{M}'}$ , we have that there exists  $(r_1, \eta_1) \in \Gamma(\Pi)_{\mathcal{M}}$  and  $(r_2, \eta_2) \in \Gamma(\Pi)_{\mathcal{M}}$  with  $(r_1, \eta_1) \prec^{\mathcal{W}} (r_2, \eta_2)$  such that if  $\langle u_1, \dots, u_k \rangle = \vec{x}_{r_1}$  and  $\langle v_1, \dots, v_l \rangle = \vec{x}_{r_2}$ , then  $\langle \eta_1(u_1), \dots, \eta_1(u_k) \rangle = \vec{a}_{r_1}$  and  $\langle \eta_2(v_1), \dots, \eta_2(v_l) \rangle = \vec{a}_{r_2}$ . Similarly, by the definition of  $\langle_{r_2 r_3}^{\mathcal{M}'}$ , there exists  $(r_2, \eta'_2) \in \Gamma(\Pi)_{\mathcal{M}}$  and  $(r_3, \eta_3) \in \Gamma(\Pi)_{\mathcal{M}}$  with  $(r_2, \eta'_2) \prec^{\mathcal{W}} (r_3, \eta_3)$  such that if  $\langle w_1, \dots, w_m \rangle = \vec{x}_{r_3}$  then  $\langle \eta'_2(v_1), \dots, \eta'_2(v_l) \rangle = \vec{a}_{r_2}$  and  $\langle \eta_3(w_1), \dots, \eta_3(w_m) \rangle = \vec{a}_{r_3}$ . Then, since  $\langle \eta'_2(v_1), \dots, \eta'_2(v_l) \rangle = \langle \eta_2(v_1), \dots, \eta_2(v_l) \rangle$ , we must have  $\eta'_2 = \eta_2$ . Then we have  $(r_1, \eta_1) \prec^{\mathcal{W}} (r_3, \eta_3)$  by transitivity since  $(r_2, \eta_2) = (r_2, \eta'_2)$ . Then from the definition of  $\langle_{r_1 r_3}^{\mathcal{M}'}$ , it follows that  $\langle \vec{a}_{r_1}, \vec{a}_{r_3} \rangle \in \langle_{r_1 r_3}^{\mathcal{M}'}$ , and hence, that  $\mathcal{M}' \models \langle_{r_1 r_3}(\vec{a}_{r_1}, \vec{a}_{r_3})$ .
- (2)  $\mathcal{M}' \models \bigwedge_{r_1, r_2 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} (\langle_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \rightarrow \neg \langle_{r_2 r_1}(\vec{x}_{r_2}, \vec{x}_{r_1}))$  Towards a contradiction, assume for some assignment  $\alpha$ , such that  $\mathcal{M}' \models \langle_{r_1 r_2}(\vec{a}_{r_1}, \vec{a}_{r_2}) \wedge \langle_{r_2 r_1}(\vec{a}_{r_2}, \vec{a}_{r_1})$ . Then we also have  $\langle_{r_1 r_1}(\vec{a}_{r_1}, \vec{a}_{r_1})$  by the transitivity axiom (which was already shown to be satisfied by  $\mathcal{M}'$ ). Then by the definition of  $\langle_{r_1 r_1}^{\mathcal{M}'}$ , there exists  $(r_1, \eta_1) \in \Lambda^t(\Pi)_{\mathcal{M}}$  and  $(r_1, \eta_2) \in \Lambda^{t_2}(\Pi)_{\mathcal{M}}$  with  $t_1 < t_2$  such that if  $\langle u_1, \dots, u_k \rangle = \vec{x}_{r_1}$ , we have  $\langle \eta_1(u_1), \dots, \eta_1(u_k) \rangle = \vec{a}_{r_1}$  and  $\langle \eta_2(u_1), \dots, \eta_2(u_k) \rangle = \vec{a}_{r_1}$ . Then this implies  $\eta_1 = \eta_2$ , and

since there is a unique  $t$  for which  $(r_1, \eta) \in \Lambda^t(\Pi)_{\mathcal{M}}$  for each  $(r_1, \eta)$ , then it must also be that  $t_1 = t_2$ . Then this is a contradiction since  $t_1 < t_2$ .

- (3)  $\mathcal{M}' \models \bigwedge_{r_1, r_2 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} (<_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \rightarrow \varphi_{r_1}^{\text{GEN}}(\vec{x}_{r_1}) \wedge \varphi_{r_2}^{\text{GEN}}(\vec{x}_{r_2}))$ . This follows from the interpretations,  $<_{r_1 r_2}^{\mathcal{M}'}$ , of the predicates  $<_{r_1 r_2}$  (for  $r_1, r_2 \in \Pi$ ) and where it is a “representation” of the well-order  $\mathcal{W} = (\Gamma(\Pi)_{\mathcal{M}}, <^{\mathcal{W}})$  on  $\Gamma(\Pi)_{\mathcal{M}}$ .
- (4)  $\mathcal{M}' \models \bigwedge_{r \in \Pi} \forall \vec{x}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \varphi_r^{\text{SUP}}(\vec{x}_r))$ . Suppose for some assignment  $\alpha$ , we have  $\mathcal{M}' \models \varphi_r^{\text{GEN}}(\vec{a}_r)$  such that  $\vec{a}_r$  is the tuple obtained from  $\vec{x}_r$  via  $\alpha$  as before. Then we show

$$\mathcal{M}' \models \bigwedge_{\substack{P(\vec{a}_{\vec{p}}) \in \text{Pos}(r), \\ P \in \mathcal{P}_{\text{int}}(\Pi)}} \bigvee_{\substack{r' \in \Pi, \\ \text{Head}(r') = P(\vec{y}_{\vec{p}})}} \exists \vec{x}_{r'} (<_{r' r}(\vec{x}_{r'}, \vec{a}_r) \wedge \vec{a}_{\vec{p}} = \vec{y}_{\vec{p}}).$$

Now, since  $\mathcal{M}' \models \varphi_r^{\text{GEN}}(\vec{a}_r)$ , then by the definition of  $\mathcal{M}'$ , it also follows that  $\mathcal{M}' \upharpoonright_{\tau(\Pi)} = \mathcal{M} \models \varphi_r^{\text{GEN}}(\vec{a}_r)$  (since  $\varphi_r^{\text{GEN}}(\vec{x}_r)$  only involves those symbols occurring in  $\tau(\Pi)$ ). Then there exists an assignment  $\eta$  such that with  $\langle u_1, \dots, u_k \rangle = \vec{x}_r$ , we have  $\langle \eta(u_1), \dots, \eta(u_k) \rangle = \vec{a}_r$  and where  $\text{Pos}(r)\eta \subseteq \mathcal{M}$  and  $\text{Neg}(r) \cap \mathcal{M} = \emptyset$ . Moreover, there must be the least stage  $t$  for which  $\text{Pos}(r)\eta \subseteq \mathcal{M}^t(\Pi)$ . Now let  $P(\vec{a}_{\vec{p}}) \in \text{Pos}(r)\eta$  where  $P \in \mathcal{P}_{\text{int}}(\Pi)$ . Then there must also be some least stage  $t'$  such that for some rule  $r'$  and corresponding assignment  $\eta'$ , we have  $\text{Head}(r')\eta' = P(\vec{a}_{\vec{p}})$ ,  $\text{Pos}(r')\eta' \subseteq \mathcal{M}^{t'}(\Pi)$ , and  $\text{Neg}(r')\eta' \cap \mathcal{M} = \emptyset$  (i.e., the least stage that derives  $P(\vec{a}_{\vec{p}})$ ). Moreover, since  $P(\vec{a}_{\vec{p}}) \in \mathcal{M}^t(\Pi)$  (i.e., since  $P(\vec{a}_{\vec{p}}) \in \text{Pos}(r)\eta \subseteq \mathcal{M}^t(\Pi)$ ), then  $t' < t$  for if  $t = t'$ , this will contradict the assumption that  $P(\vec{a}_{\vec{p}}) \in \mathcal{M}^t(\Pi)$  since this implies  $P(\vec{a}_{\vec{p}}) \notin \mathcal{M}^t(\Pi)$  (i.e., as  $t'$  is the least stage that derives  $P(\vec{a}_{\vec{p}})$ ). Then by the definitions of  $\Lambda_{\mathcal{M}}^{t'+1}(\Pi)$  and  $\Lambda_{\mathcal{M}}^{t+1}(\Pi)$ , we have  $(r', \eta') \in \Lambda_{\mathcal{M}}^{t'+1}(\Pi)$  and  $(r, \eta) \in \Lambda_{\mathcal{M}}^{t+1}(\Pi)$ . Then as  $t' + 1 < t + 1$ , by the definition of  $<^{\mathcal{W}}$ , we also have  $(r', \eta') <^{\mathcal{W}} (r, \eta)$ . Then by the interpretation of  $<_{r' r}^{\mathcal{M}'}$ , if we let  $\vec{a}_{r'} = \langle \eta'(v_1), \dots, \eta'(v_l) \rangle$  such that  $\langle v_1, \dots, v_l \rangle = \vec{x}_{r'}$ , then  $\langle \vec{a}_{r'}, \vec{a}_r \rangle \in <_{r' r}^{\mathcal{M}'}$ . Then this implies that  $\mathcal{M}' \models <_{r' r}(\vec{a}_{r'}, \vec{a}_r) \wedge \vec{a}_{\vec{p}} = \vec{a}_{\vec{p}}$  (where  $\vec{a}_{\vec{p}}$  is the projection under  $\vec{y}_{\vec{p}}$  from the tuple  $\vec{a}_{r'}$ ).

- (5)  $\mathcal{M}' \models \forall \vec{S} \varphi_{\Pi}^{\text{WELLOR}}(\vec{S})$ . That is, we show that  $\mathcal{M}'$  satisfies

$$\forall \vec{S} \left( \left( \bigwedge_{r \in \Pi} \forall \vec{x}_r (\tilde{S}_r(\vec{x}_r) \rightarrow \varphi_r^{\text{GEN}}(\vec{x}_r)) \right) \wedge \left( \bigvee_{r \in \Pi} \exists \vec{x}_r \tilde{S}_r(\vec{x}_r) \right) \rightarrow \right. \\ \left. \bigvee_{r' \in \Pi} \exists \vec{x}_{r'} (\tilde{S}_{r'}(\vec{x}_{r'}) \wedge \forall \vec{y}_{r'} (\tilde{S}_{r'}(\vec{y}_{r'}) \wedge \vec{y}_{r'} \neq \vec{x}_{r'} \rightarrow <_{r' r'}(\vec{x}_{r'}, \vec{y}_{r'})) \right) \\ \wedge \bigwedge_{\substack{r'' \in \Pi, \\ r'' \neq r'}} \forall \vec{x}_{r''} (\tilde{S}_{r''}(\vec{x}_{r''}) \rightarrow <_{r' r''}(\vec{x}_{r'}, \vec{x}_{r''})) \Big):$$

This simply follows from the assumption that  $\mathcal{W}$  is a well-ordering on  $\Gamma(\Pi)_{\mathcal{M}}$  since a well-order implies that each nonempty subset contains a least element. Indeed,  $\bigwedge_{r \in \Pi} \forall \vec{x}_r (\tilde{S}_r(\vec{x}_r) \rightarrow \varphi_r^{\text{GEN}}(\vec{x}_r))$  encodes the condition that we only consider tuples corresponding to the generating rules in the well-order,  $\bigvee_{r \in \Pi} \exists \vec{x}_r \tilde{S}_r(\vec{x}_r)$  encodes the condition that the subset of  $\Gamma(\Pi)_{\mathcal{M}}$  that we are considering is nonempty, and last,

$$\bigvee_{r' \in \Pi} \exists \vec{x}_{r'} (\tilde{S}_{r'}(\vec{x}_{r'}) \wedge \forall \vec{y}_{r'} (\tilde{S}_{r'}(\vec{y}_{r'}) \wedge \vec{y}_{r'} \neq \vec{x}_{r'} \rightarrow <_{r' r'}(\vec{x}_{r'}, \vec{y}_{r'})) \\ \wedge \bigwedge_{\substack{r'' \in \Pi, \\ r'' \neq r'}} \forall \vec{x}_{r''} (\tilde{S}_{r''}(\vec{x}_{r''}) \rightarrow <_{r' r''}(\vec{x}_{r'}, \vec{x}_{r''}))$$

encodes the condition that for such a (nonempty) subset, a least element exists.



This ends the proof of Claim 1.

CLAIM 2.  $\mathcal{M}' \models \varphi_{\Pi}^{\text{COMP}}$ .

PROOF OF CLAIM 2. From Zhang and Zhou [2010],  $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$  iff  $\mathcal{M} \models \varphi$ , where  $\varphi$  is the SO sentence as defined in Section 2.3. Moreover, if  $\mathcal{M} \models \varphi$ , then  $\mathcal{M} \models \varphi_{\Pi}^{\text{COMP}}$ . This ends the proof of Claim 2. So we have shown that  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} \varphi_{\Pi}(\vec{z}, \vec{S})$ .

( $\Leftarrow$ ) Now let us assume  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} \varphi_{\Pi}(\vec{z}, \vec{S})$ . Then for some expansion  $\mathcal{M}'$  of  $\mathcal{M}$  of the extended signature  $\tau(\Pi) \cup \{<_{r_1 r_2} \mid r_1, r_2 \in \Pi\}$ , we have  $\mathcal{M}' \models \forall \vec{S} \varphi_{\Pi}(\vec{S})$  where  $\forall \vec{S} \varphi_{\Pi}(\vec{S})$  is the sentence obtained from  $\exists \vec{z} \forall \vec{S} \varphi_{\Pi}(\vec{z}, \vec{S})$  by simply treating the predicate variables in  $\vec{z}$  as predicate constants. To show  $\mathcal{M}^{\infty}(\Pi) = \mathcal{M}$ , we show  $\mathcal{M}^{\infty}(\Pi) \subseteq \mathcal{M}$  and  $\mathcal{M} \subseteq \mathcal{M}^{\infty}(\Pi)$ . First we show  $\mathcal{M}^{\infty}(\Pi) \subseteq \mathcal{M}$  by induction.

*Basis.* Clearly,  $\mathcal{M}^0(\Pi) \subseteq \mathcal{M}$  by the definition of  $\mathcal{M}^0(\Pi)$  (i.e., only considering the interpretations of the external predicates of  $\Pi$ ).

*Step.* Assume that for  $t' \leq t$ , we have  $\mathcal{M}^{t'}(\Pi) \subseteq \mathcal{M}$ . Then let  $P(\vec{a}_{\bar{p}}) \in \mathcal{M}^{t+1}(\Pi)$  such that  $P(\vec{a}_{\bar{p}}) \notin \mathcal{M}^t(\Pi)$  (i.e., for if  $P(\vec{a}_{\bar{p}}) \in \mathcal{M}^t(\Pi)$  then the result is clear by the inductive hypothesis). We will now show  $P(\vec{a}_{\bar{p}}) \in \mathcal{M}$ . Indeed, since  $P(\vec{a}_{\bar{p}}) \in \mathcal{M}^{t+1}(\Pi)$  with  $P(\vec{a}_{\bar{p}}) \notin \mathcal{M}^t(\Pi)$ , then there exists a rule  $r \in \Pi$  and an assignment  $\eta$  such that  $\text{Head}(r)\eta = P(\vec{a}_{\bar{p}})$ ,  $\text{Pos}(r)\eta \subseteq \mathcal{M}^t(\Pi)$ , and  $\text{Neg}(r)\eta \cap \mathcal{M} = \emptyset$ . Then since  $\mathcal{M}^t(\Pi) \subseteq \mathcal{M}$  by assumption,  $\mathcal{M}' \upharpoonright_{\tau(\Pi)} = \mathcal{M}$  and  $\mathcal{M} \models \varphi_{\Pi}^{\text{COMP}}$  (i.e., since  $\varphi_{\Pi}^{\text{COMP}}$  only involves those symbols in  $\tau(\Pi)$ ), it follows that  $P(\vec{a}_{\bar{p}}) \in \mathcal{M}$ .

Thus we have shown  $\mathcal{M}^{\infty}(\Pi) \subseteq \mathcal{M}$ . Next we show  $\mathcal{M} \subseteq \mathcal{M}^{\infty}(\Pi)$ . Indeed, since  $\mathcal{M}' \models \forall \vec{S} \varphi_{\Pi}^{\text{PRO}}(\vec{S})$ , set the well-order  $\mathcal{W} = (\text{Dom}(\mathcal{W}), <^{\mathcal{W}})$  such that:

$$\text{Dom}(\mathcal{W}) = \Gamma(\Pi)_{\mathcal{M}};$$

$$<^{\mathcal{W}} = \{ \langle (r_1, \eta_1), (r_2, \eta_2) \rangle \mid r_1, r_2 \in \Pi, \eta_1 \text{ and } \eta_2 \text{ are assignments such that if } \vec{x}_{r_1} = \langle u_1, \dots, u_k \rangle \text{ and } \vec{x}_{r_2} = \langle v_1, \dots, v_l \rangle \text{ then } \langle \eta_1(u_1), \dots, \eta_1(u_k), \eta_2(v_1), \dots, \eta_2(v_l) \rangle \in <_{r_1 r_2}^{\mathcal{M}'} \}.$$

Now, for an element  $\alpha \in \text{Dom}(\mathcal{W})$ , we define the operator  $\mathcal{W}^{\alpha}(\Pi)$  inductively as follows:

$$\begin{aligned} \mathcal{W}^{\text{BOT}(\mathcal{W})}(\Pi) &= \{ \text{Head}(r)\eta \mid (r, \eta) = \text{BOT}(\mathcal{W}) \}; \\ \mathcal{W}^{\text{SUCC}(\alpha)}(\Pi) &= \mathcal{W}^{\alpha}(\Pi) \cup \{ \text{Head}(r)\eta \mid (r, \eta) = \text{SUCC}(\alpha) \}, \end{aligned}$$

where  $\text{BOT}(\mathcal{W})$  denotes the least element of  $\text{Dom}(\mathcal{W})$  under  $\mathcal{W}$ ;  $\text{SUCC}(\alpha)$  the successor element of  $\alpha$  under  $\mathcal{W}$ ; and  $\text{ORD}(\mathcal{W})$ , the order type of  $\mathcal{W}$  (that is,  $\text{ORD}(\mathcal{W})$  is equal to the size of  $\text{Dom}(\mathcal{W})$ ). As  $\mathcal{W}$  is a well-order on  $\text{Dom}(\mathcal{W})$ , we use *transfinite induction* [Enderton 1977] on the set  $\text{Dom}(\mathcal{W})$  to show that  $\mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$ .

*Basis.* Without loss of generality, assume  $\text{BOT}(\mathcal{W}) = (r, \eta)$ . Then since  $\mathcal{M}' \models \forall \vec{S} \varphi_{\Pi}^{\text{PRO}}(\vec{S})$ , there can only be two possibilities:

*Case 1.*  $\text{Pred}(\text{Pos}(r)) \cap \mathcal{P}_{\text{int}}(\Pi) = \emptyset$  (i.e., no *intentional* predicates). In this case, since  $(r, \eta) \in \text{Dom}(\mathcal{W}) = \Gamma(\Pi)_{\mathcal{M}}$  (a generating rule) and  $\mathcal{M} \models \varphi_{\Pi}^{\text{COMP}}$  (which implies that  $\mathcal{M}' \upharpoonright_{\tau(\Pi)} = \mathcal{M}$  is logically closed under  $\Pi$ ), it follows from the definition of  $\mathcal{M}^1(\Pi)$  that  $\text{Head}(r)\eta \in \mathcal{M}^1(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$ . Hence, by the definition of  $\mathcal{W}^{\text{BOT}(\mathcal{W})}(\Pi)$ , we have  $\mathcal{W}^{\text{BOT}(\mathcal{W})}(\Pi) = \{ \text{Head}(r)\eta \} \subseteq \mathcal{M}^{\infty}(\Pi)$ .

*Case 2.*  $\text{Pred}(\text{Pos}(r)) \cap \mathcal{P}_{\text{int}}(\Pi) \neq \emptyset$ . Then since  $(r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}$  and

$$\mathcal{M}' \models \bigwedge_{r \in \Pi} \forall \vec{x}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \varphi_r^{\text{SUP}}(\vec{x}_r))$$

(which obeys the notion of a support for each intentional predicate instance in the positive body from a preceding element), this contradicts the assumption  $(r, \eta)$  is the bottom element of  $\text{Dom}(\mathcal{W})$  under  $<^{\mathcal{W}}$ . Therefore, we cannot have  $\text{Pred}(\text{Pos}(r)) \cap \mathcal{P}_{\text{int}}(\Pi) \neq \emptyset$ .

*Step.* Assume for  $\text{BOT}(\mathcal{W}) \leq^{\mathcal{W}} \beta \leq^{\mathcal{W}} \alpha$ , we have  $\mathcal{W}^{\beta}(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$ .

We show  $\mathcal{W}^{\text{SUCC}(\alpha)}(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$ . Hence, assume that  $\text{SUCC}(\alpha) = (r, \eta)$ . In a similar manner to the base case, there can only be two possibilities:

*Case 1.*  $\text{Pred}(\text{Pos}(r)) \cap \mathcal{P}_{\text{int}}(\Pi) = \emptyset$ .

Then in a similar manner to Case 1 of the basis, it follows that  $\text{Head}(r)\eta \in \mathcal{M}^1(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$ , which implies  $\mathcal{W}^{\text{SUCC}(\alpha)}(\Pi) = \mathcal{W}^{\alpha}(\Pi) \cup \{\text{Head}(r)\eta\} \subseteq \mathcal{M}^{\infty}(\Pi)$ .

*Case 2.*  $\text{Pred}(\text{Pos}(r)) \cap \mathcal{P}_{\text{int}}(\Pi) \neq \emptyset$ .

Then as

$$\mathcal{M}' \models \bigwedge_{r \in \Pi} \forall \vec{x}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \varphi_r^{\text{SUP}}(\vec{x}_r))$$

(each intentional predicate instances in the positive body is supported by a preceding element), we have  $\text{Pos}(r)\eta \subseteq \mathcal{W}^{\alpha}(\Pi)$  by the definition of  $\mathcal{W}^{\alpha}(\Pi)$ . Hence, as  $\mathcal{W}^{\alpha}(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$  by assumption, then  $\text{Pos}(r)\eta \subseteq \mathcal{M}^{\infty}(\Pi)$ . Thus, there must be the least stage  $t$  for which  $\text{Pos}(r)\eta \subseteq \mathcal{M}^t(\Pi)$ . Then by the definition of  $\mathcal{M}^{t+1}(\Pi)$  and as  $(r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}$  (i.e., which implies  $\text{Neg}(r)\eta \cap \mathcal{M} = \emptyset$  as  $(r, \eta)$  is a generating rule under  $\mathcal{M}$ ), it follows that  $\text{Head}(r)\eta \in \mathcal{M}^{t+1}(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$  and hence, that  $\mathcal{W}^{\text{SUCC}(\alpha)}(\Pi) = \mathcal{W}^{\alpha}(\Pi) \cup \{\text{Head}(r)\eta\} \subseteq \mathcal{M}^{\infty}(\Pi)$ .

Thus, to show  $\mathcal{M} \subseteq \mathcal{M}^{\infty}(\Pi)$ , it will now be sufficient to only show that  $\mathcal{M} \subseteq \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \cup \mathcal{M}^1(\Pi)$  holds since  $\mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \cup \mathcal{M}^1(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$  (note that we have already verified that  $\mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$  and where  $\mathcal{M}^1(\Pi) \subseteq \mathcal{M}^{\infty}(\Pi)$ ). Thus, let  $P(\vec{a}_{\vec{P}}) \in \mathcal{M}$  where  $P \in \mathcal{P}_{\text{int}}(\Pi)$  (i.e., for if  $P \notin \mathcal{P}_{\text{int}}(\Pi)$ , then it immediately follows that  $P(\vec{a}_{\vec{P}}) \in \mathcal{M}^0(\Pi) \subseteq \mathcal{M}^1(\Pi)$ ). As  $\mathcal{M}' \models \varphi_{\Pi}^{\text{COMP}}$  implies  $\mathcal{M} \models \varphi_{\Pi}^{\text{COMP}}$  (since  $\varphi_{\Pi}^{\text{COMP}}$  only involves the symbols occurring in  $\tau(\Pi)$ ), then for some rule  $r \in \Pi$  and assignment  $\eta$ , we have  $\text{Head}(r)\eta = P(\vec{a}_{\vec{P}})$ ,  $\text{Pos}(r)\eta \subseteq \mathcal{M}$  and  $\text{Neg}(r)\eta \cap \mathcal{M} = \emptyset$ . Now, about the rule  $r$ , there can only be two possibilities:

*Case 1.*  $\text{Pred}(\text{Pos}(r)) \cap \mathcal{P}_{\text{int}}(\Pi) = \emptyset$ . Then by the definition of  $\mathcal{M}^1(\Pi)$ , we have  $\text{Head}(r)\eta = P(\vec{a}_{\vec{P}}) \in \mathcal{M}^1(\Pi) \subseteq \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \cup \mathcal{M}^1(\Pi)$ .

*Case 2.*  $\text{Pred}(\text{Pos}(r)) \cap \mathcal{P}_{\text{int}}(\Pi) \neq \emptyset$ .

Then as  $(r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}$  and since  $\text{Dom}(\mathcal{W}) = \Gamma(\Pi)_{\mathcal{M}}$ , it is clear that  $(r, \eta) \in \text{Dom}(\mathcal{W})$ . Hence, by the definition of  $\mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi)$ , we have  $P(\vec{a}_{\vec{P}}) \in \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \subseteq \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \cup \mathcal{M}^1(\Pi)$ .

Therefore, we have  $\mathcal{M} \subseteq \mathcal{M}^{\infty}(\Pi)$ . This finally completes the proof of this theorem.  $\square$

#### 5.4. Incorporating Preference

We have shown that the SO sentence  $\exists \vec{z} \forall \vec{S} \varphi_{\Pi}(\vec{z}, \vec{S})$  proposed in Theorem 5.5 precisely captures the answer set semantics for FO answer set programs. In this section, we further extend this formula by embedding preferences. We first define the following formula.

**Definition 5.6.** Given an FO preferred answer set  $(\Pi, <)$ , we define the formula  $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$  as follows (here “PREF” stands for preference):

$$\bigwedge_{r \in \Pi} \forall \vec{x}_r \left( \varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \bigwedge_{r' < r} \forall \vec{x}_{r'} \left( (\varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow \widetilde{<}_{r'r}(\vec{x}_{r'}, \vec{x}_r)) \right) \right. \\ \left. \wedge (\neg \varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow (\varphi_{r'}^{-\text{POS}}(\vec{x}_{r'}) \vee \varphi_{r'r'}^{\text{DEF}}(\vec{z}, \vec{x}_r, \vec{x}_{r'}))) \right), \quad (14)$$

where:

—  $\varphi_{r'}^{-\text{POS}}(\vec{x}_{r'})$  denotes the following formula (“ $\neg$ POS” stands for *positive body not satisfied*):

$$\bigvee_{P(\vec{x}_P) \in \text{Pos}(r')} \neg P(\vec{x}_P); \quad (15)$$

—  $\varphi_{r'r'}^{\text{DEF}}(\vec{z}, \vec{x}_r, \vec{x}_{r'})$  (“DEF” stands for *defeated*) denotes the formula

$$\left( \bigvee_{\substack{P(\vec{x}_P) \in \text{Neg}(r'), \\ P \notin \text{Pint}(\Pi)}} P(\vec{x}_P) \right) \vee \left( \bigvee_{\substack{P(\vec{x}_P) \in \text{Neg}(r'), \\ P \in \text{Pint}(\Pi)}} \bigvee_{\substack{r'' \in \Pi, \\ \text{Head}(r'') = P(\vec{y}_P)}} \exists \vec{x}_{r''} (\widetilde{<}_{r''r}(\vec{x}_{r''), \vec{x}_r) \wedge \vec{x}_P = \vec{y}_P) \right). \quad (16)$$

Formula (14) encodes that if  $r$  is a generating rule (under a structure and associated assignment), then for each rule  $r'$  that is more preferred than  $r$ , where  $r'$  is also a generating rule (under the structure and associated assignment), we require that  $r'$  should have already been derived earlier than  $r$  in the progression stages. On the other hand, in the case that  $r'$  is not a generating rule, then either the positive body of  $r'$  is not satisfied, that is, by (15), or it is defeated by some other rule  $r''$  such that  $r''$  is already derived earlier than  $r$  in the progression stages, that is, by (16), which indicates that the head of  $r''$  occurs in the negative body of  $r'$ .

**THEOREM 5.7.** Let  $(\Pi, <)$  be an FO preferred answer set program, and  $\mathcal{M}$  a  $\tau(\Pi)$ -structure.  $\mathcal{M}$  is a preferred answer set of  $(\Pi, <)$  iff  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$ , where  $\varphi_{(\Pi, <)}(\vec{z}, \vec{S}) = \varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z}) \wedge \varphi_{\Pi}^{\text{PRO}}(\vec{z}, \vec{S}) \wedge \varphi_{\Pi}^{\text{COMP}}$ .

**PROOF.** ( $\Rightarrow$ ) Assume  $\lambda_{\mathcal{M}^0}(\Gamma^{\infty}(\Pi)_{\mathcal{M}}) = \mathcal{M}$  (i.e.,  $\mathcal{M}$  is a preferred answer set of  $(\Pi, <)$ ). We show  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  in two steps:

- (1) Given  $\lambda_{\mathcal{M}^0}(\Gamma^{\infty}(\Pi)_{\mathcal{M}}) = \mathcal{M}$ , we show there exists a well-order  $\mathcal{W} = (\Gamma(\Pi)_{\mathcal{M}}, <^{\mathcal{W}})$  on the set  $\Gamma(\Pi)_{\mathcal{M}}$  called the *preference preserving well-order*, satisfying the following conditions. For each  $(r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}$ :
  - (a)  $\text{Pos}(r)\eta \subseteq \mathcal{M}^0(\Pi) \cup \{\text{Head}(r')\eta' \mid (r', \eta') <^{\mathcal{W}} (r, \eta)\}$ ;
  - (b) for each rule  $r' < r$  and assignment  $\eta'$ :
    - i.  $(r', \eta') \in \Gamma(\Pi)_{\mathcal{M}}$  implies  $(r', \eta') <^{\mathcal{W}} (r, \eta)$ ;
    - ii.  $(r', \eta') \notin \Gamma(\Pi)_{\mathcal{M}}$  implies that either:
      - A.  $\text{Pos}(r')\eta' \not\subseteq \mathcal{M}$ , or
      - B.  $\text{Neg}(r')\eta' \cap (\mathcal{M}^0(\Pi) \cup \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^{\mathcal{W}} (r, \eta)\}) \neq \emptyset$ .
- (2) Based on the well-order  $\mathcal{W}$  on  $\Gamma(\Pi)_{\mathcal{M}}$ , we construct an expansion  $\mathcal{M}'$  of  $\mathcal{M}$  and show that  $\mathcal{M}' \models \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  where  $\forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  is the sentence obtained from  $\mathcal{M} \models \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  by treating the predicate variables in  $\vec{z}$  as constants.

Now for the first part, we start with the following claim.

**CLAIM 1.**  $\Gamma^\infty(\Pi)_\mathcal{M} = \Gamma(\Pi)_\mathcal{M}$ . *That is,  $\Gamma^\infty(\Pi)_\mathcal{M}$  contains exactly the generating rules.*

**PROOF OF CLAIM 1.** Since  $\mathcal{M}$  is a preferred answer set of  $(\Pi, <)$ , then by Theorem 4.9,  $(r, \eta) \in \Gamma^\infty(\Pi)_\mathcal{M}$  iff  $r$  is a generating rule of  $\mathcal{M}$  under  $\eta$ . This completes the proof of Claim 1.

Now, based on  $\Gamma^\infty(\Pi)_\mathcal{M}$ , we show by induction on  $t$  for  $t \geq 0$  that there exists a well-order  $\mathcal{W} = (\Gamma^t(\Pi)_\mathcal{M}, <^\mathcal{W})$  on  $\Gamma^t(\Pi)_\mathcal{M}$  with the following properties. For each  $(r, \eta) \in \Gamma^t(\Pi)_\mathcal{M}$ :

- (1)  $Pos(r)\eta \subseteq \mathcal{M}^0(\Pi) \cup \{Head(r')\eta' \mid (r', \eta') <^\mathcal{W} (r, \eta)\}$ ;
- (2) for each rule  $r' < r$  and assignment  $\eta'$ :
  - (a)  $(r', \eta') \in \Gamma^t(\Pi)_\mathcal{M}$  implies  $(r', \eta') <^\mathcal{W} (r, \eta)$ ;
  - (b)  $(r', \eta') \notin \Gamma(\Pi)_\mathcal{M}$  implies that either:
    - i.  $Pos(r')\eta' \not\subseteq \mathcal{M}$  or
    - ii.  $Neg(r')\eta' \cap (\mathcal{M}^0(\Pi) \cup \{Head(r'')\eta'' \mid (r'', \eta'') <^\mathcal{W} (r, \eta)\}) \neq \emptyset$ .

Thus, since  $\Gamma^\infty(\Pi)_\mathcal{M} = \Gamma(\Pi)_\mathcal{M}$  by Claim 1, we would have showed the first part.

*Basis.* By the *well-ordering theorem* [Enderton 1977] (which states that every set can be well-ordered), there exists a well-order  $\mathcal{W} = (\Gamma^0(\Pi)_\mathcal{M}, <^\mathcal{W})$  on  $\Gamma^0(\Pi)_\mathcal{M}$ . Moreover, due to the finiteness of  $\Pi$ , it should not be too difficult to see that we can make the well-order  $\mathcal{W}$  in such a way that for each  $(r, \eta), (r', \eta') \in \Gamma^0(\Pi)_\mathcal{M}$ ,  $r < r'$  implies  $(r, \eta) <^\mathcal{W} (r', \eta')$ . To see this, note that we can partition  $\Gamma^0(\Pi)_\mathcal{M}$  into a sequence of sets  $S_{r_1}, \dots, S_{r_n}$  where  $r_i < r_j$  implies  $i < j$  and such that  $(r, \eta) \in S_{r_i}$  implies  $r = r_i$ . Then, by the well-ordering theorem, there exists a well-order  $\mathcal{W}_i = (S_{r_i}, <^{\mathcal{W}_i})$  of each of the sets  $S_{r_i}$ . Thus, we can just set  $\mathcal{W} = (\Gamma^0(\Pi)_\mathcal{M}, <^\mathcal{W})$  by setting  $<^\mathcal{W} = (\bigcup_{1 \leq i \leq n} <^{\mathcal{W}_i}) \cup \{(r_i, \eta_i), (r_j, \eta_j) \mid (r_i, \eta_i) \in S_{r_i}, (r_j, \eta_j) \in S_{r_j}, i < j\}$ . This simply follows from the fact that the *sum* and *products* of well-ordered types are also well-ordered [Enderton 1977]. Hence, assume  $\mathcal{W}$  to be such a well-order. We now make the following claims:

**CLAIM 2.**  $(r, \eta) \in \Gamma^0(\Pi)_\mathcal{M}$  implies  $Pos(r)\eta \subseteq \mathcal{M}^0(\Pi) \cup \{Head(r')\eta' \mid (r', \eta') <^\mathcal{W} (r, \eta)\}$ . *Clearly by the definition of  $\Gamma^0(\Pi)_\mathcal{M}$ ,  $(r, \eta) \in \Gamma^0(\Pi)_\mathcal{M}$  implies  $Pos(r)\eta \subseteq \mathcal{M}^0(\Pi)$ , which indicates that Claim 2 is true.*

**CLAIM 3.** *If  $(r, \eta) \in \Gamma^0(\Pi)_\mathcal{M}$ , then  $(r', \eta') \in \Gamma^0(\Pi)_\mathcal{M}$ , where  $r' < r$  implies  $(r', \eta') <^\mathcal{W} (r, \eta)$ . This claim directly follows from the description of  $\mathcal{W}$ .*

**CLAIM 4.** *If  $(r, \eta) \in \Gamma^0(\Pi)_\mathcal{M}$ , then  $(r', \eta') \notin \Gamma(\Pi)_\mathcal{M}$ , where  $r' < r$  implies that either:*

- (1)  $Pos(r')\eta' \not\subseteq \mathcal{M}$  or
- (2)  $Neg(r')\eta' \cap (\mathcal{M}^0(\Pi) \cup \{Head(r'')\eta'' \mid (r'', \eta'') <^\mathcal{W} (r, \eta)\}) \neq \emptyset$ .

**PROOF OF CLAIM 4.** On the contrary, assume that  $Pos(r')\eta' \subseteq \mathcal{M}$  and  $Neg(r')\eta' \cap (\mathcal{M}^0(\Pi) \cup \{Head(r'')\eta'' \mid (r'', \eta'') <^\mathcal{W} (r, \eta)\}) = \emptyset$ . Then, by the latter, we also have  $Neg(r')\eta' \cap \mathcal{M}^0(\Pi) = \emptyset$ . But this contradicts the assumption  $(r, \eta) \in \Gamma^0(\Pi)_\mathcal{M}$  since by the definition of  $\Gamma^0(\Pi)_\mathcal{M}$ , we have  $(r', \eta')$  will be blocking  $(r, \eta)$  from being applied.

*Step.* Assume for  $1 \leq t' \leq t$  that the hypothesis holds. We will now show it also holds for  $t + 1$ . Indeed, by the inductive hypothesis, there exists a well-order  $\mathcal{W} = (\Gamma^t(\Pi)_\mathcal{M}, <^\mathcal{W})$  on  $\Gamma^t(\Pi)_\mathcal{M}$  satisfying the conditions of the preference preserving well-order. Moreover, by the well-ordering theorem, there also exists a well-order  $\mathcal{W}' = (\Lambda^{t+1}(\Pi)_\mathcal{M}, <^{\mathcal{W}'})$  on  $\Lambda^{t+1}(\Pi)_\mathcal{M}$  where  $\Lambda^{t+1}(\Pi)_\mathcal{M} = \Gamma^{t+1}(\Pi)_\mathcal{M} \setminus \Gamma^t(\Pi)_\mathcal{M}$ . Furthermore, due to the finiteness of  $\Pi$ , it is not difficult to see that we can further define  $\mathcal{W}'$  in such a way that for each  $(r, \eta), (r', \eta') \in \Lambda^{t+1}(\Pi)_\mathcal{M}$ ,  $r < r'$  implies  $(r, \eta) <^{\mathcal{W}'} (r', \eta')$  (i.e., using the

same argument as before). Now we specify  $\mathcal{W}'' = (\Gamma^{t+1}(\Pi)_{\mathcal{M}}, <^{\mathcal{W}''})$  to be a well-order on  $\Gamma^{t+1}(\Pi)_{\mathcal{M}}$  by setting  $<^{\mathcal{W}''} = \mathcal{W} \cup \mathcal{W}' \cup (\Gamma^t(\Pi)_{\mathcal{M}} \times \Lambda^{t+1}(\Pi)_{\mathcal{M}})$ .

CLAIM 5.  $(r, \eta) \in \Gamma^{t+1}(\Pi)_{\mathcal{M}}$  implies  $Pos(r)\eta \subseteq \mathcal{M}^0(\Pi) \cup \{Head(r')\eta' \mid (r', \eta') <^{\mathcal{W}''} (r, \eta)\}$ .

PROOF OF CLAIM 5. We consider all possible cases:

Case 1.  $(r, \eta) \in \Gamma^t(\Pi)_{\mathcal{M}}$ .

Then by the inductive hypothesis, we have  $Pos(r)\eta \subseteq \mathcal{M}^0(\Pi) \cup \{Head(r')\eta' \mid (r', \eta') <^{\mathcal{W}} (r, \eta)\}$ . Then since  $<^{\mathcal{W}} \subseteq <^{\mathcal{W}''}$ , it also follows that  $Pos(r)\eta \subseteq \mathcal{M}^0(\Pi) \cup \{Head(r')\eta' \mid (r', \eta') <^{\mathcal{W}''} (r, \eta)\}$ .

Case 2.  $(r, \eta) \in \Lambda^{t+1}(\Pi)_{\mathcal{M}}$ .

Then by the definition of  $\Lambda^{t+1}(\Pi)_{\mathcal{M}}$ , we have  $(r, \eta) \in \Gamma^{t+1}(\Pi)_{\mathcal{M}}$  and  $(r, \eta) \notin \Gamma^t(\Pi)_{\mathcal{M}}$ , which implies  $(r, \eta)$  is derived at stage  $t + 1$ . Then by the definition of  $\Gamma^{t+1}(\Pi)_{\mathcal{M}}$ , it follows that  $Pos(r)\eta \subseteq \mathcal{M}^0(\Pi) \cup \{Head(r')\eta' \mid (r', \eta') \in \Gamma^t(\Pi)_{\mathcal{M}}\}$ . Then as  $<^{\mathcal{W}} \subseteq <^{\mathcal{W}''}$  (i.e., which implies  $(r', \eta') <^{\mathcal{W}''} (r, \eta)$  for all  $(r', \eta') \in \Gamma^t(\Pi)_{\mathcal{M}}$ ), it also follows that  $Pos(r)\eta \subseteq \mathcal{M}^0(\Pi) \cup \{Head(r')\eta' \mid (r', \eta') <^{\mathcal{W}''} (r, \eta)\}$ . This completes the proof of Claim 5.

CLAIM 6. If  $(r, \eta) \in \Gamma^{t+1}(\Pi)_{\mathcal{M}}$ , then  $(r', \eta') \in \Gamma^{t+1}(\Pi)_{\mathcal{M}}$  where  $r' < r$  implies  $(r', \eta') <^{\mathcal{W}''} (r, \eta)$ .

PROOF OF CLAIM 6. Towards a contradiction, assume  $(r, \eta), (r', \eta') \in \Gamma^{t+1}(\Pi)_{\mathcal{M}}$ ,  $r' < r$ , where  $(r, \eta) <^{\mathcal{W}''} (r', \eta')$ . We also consider all possibilities:

Case 1.  $(r, \eta), (r', \eta') \in \Gamma^t(\Pi)_{\mathcal{M}}$ .

Then this is a contradiction, since by the inductive hypothesis, we must have that  $(r', \eta') <^{\mathcal{W}''} (r, \eta)$ .

Case 2.  $(r, \eta) \in \Gamma^t(\Pi)_{\mathcal{M}}$  and  $(r', \eta') \in \Lambda^{t+1}(\Pi)_{\mathcal{M}}$ .

Then as  $(r', \eta') \in \Lambda^{t+1}(\Pi)_{\mathcal{M}}$ , we have:

- (1)  $Pos(r')\eta' \subseteq \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}})$ ;
- (2)  $Neg(r')\eta' \cap \mathcal{M} = \emptyset$ .

Then, since  $\lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) \subseteq \mathcal{M}$  (i.e., since  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) = \mathcal{M}$  by assumption), it follows that  $(r', \eta')$  is also a pair such that:

- (1)  $r' < r$  (i.e., by the assumption);
- (2)  $Pos(r')\eta' \subseteq \mathcal{M}$ ;
- (3)  $Neg(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) = \emptyset$ ;
- (4)  $(r', \eta') \notin \Gamma^t(\Pi)_{\mathcal{M}}$  (i.e., by the definition of  $\Lambda^{t+1}(\Pi)_{\mathcal{M}}$  and as  $(r', \eta') \in \Lambda^{t+1}(\Pi)_{\mathcal{M}}$  for all  $1 \leq t' \leq t$ ).

Then by the definition of  $\Gamma^t(\Pi)_{\mathcal{M}}$ , this contradicts the assumption that  $(r, \eta) \in \Gamma^t(\Pi)_{\mathcal{M}}$  because the pair  $(r', \eta')$  will always be blocking  $(r, \eta)$  from being applied at all stages  $0 \leq t' \leq t$ .

Case 3.  $(r', \eta') \in \Gamma^t(\Pi)_{\mathcal{M}}$  and  $(r, \eta) \in \Lambda^{t+1}(\Pi)_{\mathcal{M}}$ .

Then by the construction of  $\mathcal{W}''$ , we have that  $(r', \eta') <^{\mathcal{W}''} (r, \eta)$ , which contradicts the assumption  $(r, \eta) <^{\mathcal{W}''} (r', \eta')$ . Therefore, we cannot have this possibility.

Case 4.  $(r, \eta), (r', \eta') \in \Lambda^{t+1}(\Pi)_{\mathcal{M}}$ .

Then since  $<^{\mathcal{W}'} \subseteq <^{\mathcal{W}''}$  and by the construction of the well-order  $\mathcal{W}'$  on  $\Lambda^{t+1}(\Pi)_{\mathcal{M}}$ , then we must have  $(r', \eta') <^{\mathcal{W}''} (r, \eta)$  since  $r' < r$  by assumption. This is contrary to the initial assumption  $(r, \eta) <^{\mathcal{W}''} (r', \eta')$ . This ends the proof of Claim 6.

CLAIM 7. *If  $(r, \eta) \in \Gamma^{t+1}(\Pi)_{\mathcal{M}}$ , then  $(r', \eta') \notin \Gamma(\Pi)_{\mathcal{M}}$  where  $r' < r$  implies that either:*

- (1)  $Pos(r')\eta' \not\subseteq \mathcal{M}$  or
- (2)  $Neg(r')\eta' \cap (\mathcal{M}^0(\Pi) \cup \{Head(r'')\eta'' \mid (r'', \eta'') <^{\mathcal{W}''} (r, \eta)\}) \neq \emptyset$ .

PROOF OF CLAIM 7. Again we again consider the following possibilities:

Case 1.  $(r, \eta) \in \Gamma^t(\Pi)_{\mathcal{M}}$ .

Then it immediately follows by the inductive hypothesis that Claim 7 holds.

Case 2.  $(r, \eta) \in \Lambda^{t+1}(\Pi)_{\mathcal{M}}$ .

For the sake of contradiction, assume  $Pos(r')\eta' \subseteq \mathcal{M}$  and  $Neg(r')\eta' \cap (\mathcal{M}^0(\Pi) \cup \{Head(r'')\eta'' \mid (r'', \eta'') <^{\mathcal{W}''} (r, \eta)\}) = \emptyset$ . Then by the construction of  $\mathcal{W}''$ , it also follows that  $Neg(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) = \emptyset$ . Then this implies  $(r', \eta')$  is a pair such that:

- (1)  $r' < r$  (i.e., by assumption);
- (2)  $Pos(r')\eta' \subseteq \mathcal{M}$  and  $Neg(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) = \emptyset$ ;
- (3)  $(r', \eta') \notin \Gamma^t(\Pi)_{\mathcal{M}}$  (i.e., since  $(r', \eta') \notin \Gamma(\Pi)_{\mathcal{M}}$  by assumption and where  $\Gamma^\infty(\Pi)_{\mathcal{M}} = \Gamma(\Pi)_{\mathcal{M}}$  by Claim 1).

Then this contradicts the assumption  $(r, \eta) \in \Lambda^{t+1}(\Pi)_{\mathcal{M}} = \Gamma^{t+1}(\Pi)_{\mathcal{M}} \setminus \Gamma^t(\Pi)_{\mathcal{M}}$  (i.e.,  $(r, \eta)$  is applied in  $\Gamma^{t+1}(\Pi)_{\mathcal{M}}$ ) since by the definition of  $\Gamma^{t+1}(\Pi)_{\mathcal{M}}$ , we have  $(r', \eta')$  is blocking  $(r, \eta)$  from being applied at stage  $t + 1$ . This ends the proof of Claim 7.

Thus, by Claims 5, 6, and 7, we have that the hypothesis also holds for  $t + 1$ .

Now we prove the second part by showing  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$ . As mentioned in the beginning, based on the result of the first part (i.e., that there exists a preference preserving well-order  $\mathcal{W} = (\Gamma(\Pi)_{\mathcal{M}}, <^{\mathcal{W}})$  on the set  $\Gamma(\Pi)_{\mathcal{M}}$ ), we now construct an expansion  $\mathcal{M}'$  of  $\mathcal{M}$  based on the well-order  $\mathcal{W}$ . Thus, set the structure  $\mathcal{M}'$  to be an expansion of  $\mathcal{M}$  on the signature  $\tau(\Pi) \cup \{<_{r_1 r_2} \mid r_1, r_2 \in \Pi\}$  (i.e., where  $\mathcal{M}$  is of the signature  $\tau(\Pi)$ ) such that for each (new) predicate symbol  $<_{r_1 r_2}$  (i.e., where  $r_1, r_2 \in \Pi$  and  $r_1$  and  $r_2$  could be the same), we have

$$\begin{aligned} <_{r_1 r_2}^{\mathcal{M}'} = \{ \langle \eta_1(u_1), \dots, \eta_1(u_k), \eta_2(v_1), \dots, \eta_2(v_l) \rangle \mid \langle u_1, \dots, u_k \rangle = \vec{x}_{r_1}, \langle v_1, \dots, v_l \rangle = \vec{x}_{r_2}, \\ & \quad (r_1, \eta_1) <^{\mathcal{W}} (r_2, \eta_2) \}. \end{aligned}$$

Then in a similar manner to the proof of Theorem 5.5, it can be shown (although tedious) that  $\mathcal{M}' \models \forall \vec{S} \varphi_{(\Pi, <)}(\vec{S})$ .

( $\Leftarrow$ ) Assume  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$ . We show  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) = \mathcal{M}$  by showing  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) \subseteq \mathcal{M}$  and  $\mathcal{M} \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . Indeed, since  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  implies  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} (\varphi_{\Pi}^{\text{PRO}}(\vec{z}, \vec{S}) \wedge \varphi_{\Pi}^{\text{COMP}})$ , then  $\mathcal{M}$  is an answer set of  $\Pi$  by Theorem 5.5. Thus, from [Zhang and Zhou 2010], we have  $\mathcal{M}^\infty(\Pi) = \mathcal{M}$ . Then since  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) \subseteq \mathcal{M}^\infty(\Pi)$ , it follows that  $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) \subseteq \mathcal{M}$ . Therefore, it is only left for us to show  $\mathcal{M} \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . Now, since  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$ , there exists an expansion  $\mathcal{M}'$  of  $\mathcal{M}$ , on the signature  $\tau(\Pi) \cup \{<_{r_1 r_2} \mid r_1, r_2 \in \Pi\}$ , such that  $\mathcal{M}' \models \forall \vec{S} \varphi_{(\Pi, <)}(\vec{S})$  where  $\forall \vec{S} \varphi_{(\Pi, <)}(\vec{S})$  is the sentence obtained from  $\forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  by treating the predicate variables in  $\vec{z}$  as constants. Then since  $\mathcal{M}' \models \forall \vec{S} \varphi_{(\Pi, <)}(\vec{S})$  (i.e., which implies a well-order on  $\Gamma(\Pi)_{\mathcal{M}}$ ), we construct a well-order  $\mathcal{W} = (\Gamma(\Pi)_{\mathcal{M}}, <^{\mathcal{W}})$  of  $\Gamma(\Pi)_{\mathcal{M}}$  by setting

$$\begin{aligned} <^{\mathcal{W}} = \{ \langle (r_1, \eta_1), (r_2, \eta_2) \rangle \mid r_1, r_2 \in \Pi, \vec{x}_{r_1} = \langle u_1, \dots, u_k \rangle, \vec{x}_{r_2} = \langle v_1, \dots, v_l \rangle, \\ & \quad \langle \eta_1(u_1), \dots, \eta_1(u_k), \eta_2(v_1), \dots, \eta_2(v_l) \rangle \in <_{r_1 r_2}^{\mathcal{M}'} \}. \end{aligned}$$

Hence, for an  $\alpha \in \Gamma(\Pi)_{\mathcal{M}}$ , define  $\mathcal{W}^\alpha(\Pi)$  inductively as follows:

$$\begin{aligned}\mathcal{W}^{\text{BOT}(\mathcal{W})}(\Pi) &= \{\text{BOT}(\mathcal{W})\}; \\ \mathcal{W}^{\text{SUCC}(\alpha)}(\Pi) &= \mathcal{W}^\alpha(\Pi) \cup \{\text{SUCC}(\alpha)\},\end{aligned}$$

where  $\text{BOT}(\mathcal{W})$ ,  $\text{SUCC}(\alpha)$ , and  $\text{ORD}(\mathcal{W})$  denotes the bottom element, the successor element of  $\alpha$  and the order type of  $\Gamma(\Pi)_{\mathcal{M}}$  under  $\mathcal{W}$ , respectively. Intuitively,  $\mathcal{W}^\alpha(\Pi)$  represents the “gradual” collection of the pairs  $(r, \eta)$  of  $\Gamma(\Pi)_{\mathcal{M}}$  under the well-order  $\mathcal{W}$  up to and including  $\alpha$ .

**CLAIM 1.**  $\mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \subseteq \Gamma^\infty(\Pi)_{\mathcal{M}}$ .

**PROOF OF CLAIM 1.** We prove by induction on  $\alpha$  for  $\alpha \geq \text{BOT}(\mathcal{W})$ .

*Basis.* Since  $\mathcal{W}$  is a preference preserving well-order on  $\Gamma(\Pi)_{\mathcal{M}}$ , then we have that  $\text{BOT}(\mathcal{W}) = (r, \eta)$  is such that:

- (1)  $\text{Pred}(\text{Pos}(r)) \cap \mathcal{P}_{\text{int}}(\Pi) = \emptyset$  (i.e., since  $(r, \eta)$  is the least element under  $\mathcal{W}$  of  $\Gamma(\Pi)_{\mathcal{M}}$  and that the well-order  $\mathcal{W}$  satisfies the notion of *support* since  $\mathcal{M}' \models \bigwedge_{r \in \Pi} \forall \vec{x}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \varphi_r^{\text{SUP}}(\vec{x}_r))$ );
- (2) Any rule  $r' < r$  and assignment  $\eta'$  implies  $(r', \eta') \notin \Gamma(\Pi)_{\mathcal{M}}$ . For suppose  $(r', \eta') \in \Gamma(\Pi)_{\mathcal{M}}$ , then we must have  $(r', \eta') <^{\mathcal{W}} (r, \eta)$  since  $r' < r$ ,  $(r, \eta)$  and  $(r', \eta')$  are both in  $\Gamma(\Pi)_{\mathcal{M}}$ , and

$$\mathcal{M}' \models \forall \vec{x}_r \left( \varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \bigwedge_{r' < r} \forall \vec{x}_{r'} (\varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow \langle_{r'r}(\vec{x}_{r'}, \vec{x}_r) \right)$$

(i.e., since  $\mathcal{M}' \models \varphi_{(\Pi, <)}^{\text{PREF}}$ ), which is absurd since  $(r, \eta)$  is the least most under  $\mathcal{W}$ ;

- (3) For each rule  $r' < r$  and assignment  $\eta'$  with  $(r', \eta') \notin \Gamma(\Pi)_{\mathcal{M}}$ , we have either:
  - (a)  $\text{Pos}(r')\eta' \not\subseteq \mathcal{M}$  or
  - (b)  $\text{Neg}(r')\eta' \cap (\mathcal{M}^0(\Pi) \cup \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^{\mathcal{W}} (r, \eta)\}) \neq \emptyset$ ,
since

$$\mathcal{M}' \models \forall \vec{x}_r \left( \varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \bigwedge_{r' < r} \forall \vec{x}_{r'} (\neg \varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow (\Phi_{r'}^{-\text{POS}}(\vec{x}_{r'}) \vee \Phi_{r'}^{\text{DEF}}(\vec{x}_r, \vec{x}_{r'})) \right).$$

Then since  $(r, \eta) = \text{BOT}(\mathcal{W})$  (i.e., the least element under  $\mathcal{W}$ , which means that  $\{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^{\mathcal{W}} (r, \eta)\} = \emptyset$ ), it must be that  $\text{Pos}(r')\eta' \not\subseteq \mathcal{M}$  or  $\text{Neg}(r')\eta' \cap \mathcal{M}^0(\Pi) \neq \emptyset$ .

Then by (2) and (3), we know that there does not exist a rule  $r' < r$  and assignment  $\eta'$  with  $\text{Pos}(r')\eta' \subseteq \mathcal{M}$  and  $\text{Neg}(r')\eta' \cap \mathcal{M}^0(\Pi) = \emptyset$  since we will always have  $\text{Pos}(r')\eta' \not\subseteq \mathcal{M}$  or  $\text{Neg}(r')\eta' \cap \mathcal{M}^0(\Pi) \neq \emptyset$ . Hence, as  $(r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}$  and where  $\text{Pos}(r)\eta \subseteq \mathcal{M}^0(\Pi)$  by 1 this implies  $(r, \eta) \in \Gamma^0(\Pi)_{\mathcal{M}}$  by the definition of  $\Gamma^0(\Pi)_{\mathcal{M}}$ . Thus, we have  $\mathcal{W}^{\text{BOT}(\mathcal{W})}(\Pi) \subseteq \Gamma^0(\Pi)_{\mathcal{M}} \subseteq \Gamma^\infty(\Pi)_{\mathcal{M}}$ .

*Step.* Assume for  $\text{BOT}(\mathcal{W}) \leq^{\mathcal{W}} \beta \leq^{\mathcal{W}} \alpha$ , we have  $\mathcal{W}^\beta(\Pi) \subseteq \Gamma^\infty(\Pi)_{\mathcal{M}}$ .

We will show  $\mathcal{W}^{\text{SUCC}(\alpha)}(\Pi) \subseteq \Gamma^\infty(\Pi)_{\mathcal{M}}$ . Thus, assume  $\text{SUCC}(\alpha) = (r, \eta)$ . Then by the inductive hypothesis, there exists some  $t$  for which  $\mathcal{W}^\alpha(\Pi) \subseteq \Gamma^t(\Pi)_{\mathcal{M}}$  (i.e., since  $\mathcal{W}^\alpha(\Pi) \subseteq \Gamma^\infty(\Pi)_{\mathcal{M}}$ ). Moreover, since  $\mathcal{M}' \models \bigwedge_{r \in \Pi} \forall \vec{x}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \varphi_r^{\text{SUP}}(\vec{x}_r))$  (i.e., obeys the notion of “support”), then we also have  $\text{Pos}(r)\eta \subseteq \lambda_{\mathcal{M}^0}(\mathcal{W}^\alpha(\Pi))$ , and hence, that  $\text{Pos}(r)\eta \subseteq \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}})$ , since  $\mathcal{W}^\alpha(\Pi) \subseteq \Gamma^t(\Pi)_{\mathcal{M}}$ .

**SUBCLAIM 1.** Here does not exist a rule  $r' < r$  and assignment  $\eta'$  such that

- (1)  $\text{Pos}(r')\eta' \subseteq \mathcal{M}$  and  $\text{Neg}(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) = \emptyset$ ;
- (2)  $(r', \eta') \notin \Gamma^t(\Pi)_{\mathcal{M}}$ .

PROOF OF SUBCLAIM 1. On the contrary, assume that there exists such a rule  $r'$  and assignment  $\eta'$ . Then there can only be two possibilities:

*Case 1.*  $(r', \eta') \in \Gamma(\Pi)_{\mathcal{M}}$ . As

$$\mathcal{M}' \models \forall \vec{x}_r \left( \varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \bigwedge_{r' < r} \forall \vec{x}_{r'} (\varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow \langle_{r'r}(\vec{x}_r, \vec{x}_{r'}) \rangle) \right)$$

and since  $r' < r$ , then we must have  $(r', \eta') <^{\mathcal{W}} (r, \eta) = \text{succ}(\alpha)$ , or in other words, that  $(r', \eta') \in \mathcal{W}^\alpha(\Pi)$  (i.e., by the definition of  $\mathcal{W}^\alpha(\Pi)$ ). Then since  $\mathcal{W}^\alpha(\Pi) \subseteq \Gamma^t(\Pi)_{\mathcal{M}}$  by the inductive hypothesis, this contradicts the assumption  $(r', \eta') \notin \Gamma^t(\Pi)_{\mathcal{M}}$ .

*Case 2.*  $(r', \eta') \notin \Gamma(\Pi)_{\mathcal{M}}$ . Then since

$$\mathcal{M}' \models \forall \vec{x}_r \left( \varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \bigwedge_{r' < r} \forall \vec{x}_{r'} (\neg \varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow (\Phi_{r'}^{-\text{POS}}(\vec{x}_{r'}) \vee \Phi_{r'r'}^{\text{DEF}}(\vec{x}_r, \vec{x}_{r'}))) \right),$$

there can be only two further possibilities:

—*Subcase 1.*  $\text{Pos}(r')\eta' \not\subseteq \mathcal{M}$ . Then this contradicts the assumption  $\text{Pos}(r')\eta' \subseteq \mathcal{M}$ .

—*Subcase 2.*  $\text{Neg}(r')\eta' \cap \lambda_{\mathcal{M}^0}(\mathcal{W}^\alpha(\Pi)) \neq \emptyset$ . But this contradicts the assumption  $\text{Neg}(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^t(\Pi)_{\mathcal{M}}) = \emptyset$ , since  $\mathcal{W}^\alpha(\Pi) \subseteq \Gamma^t(\Pi)_{\mathcal{M}}$ , by the inductive hypothesis. This completes the proof of Subclaim 1.

Therefore, using the fact that  $\mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \subseteq \Gamma^\infty(\Pi)_{\mathcal{M}}$  by Claim 1, it is now sufficient to only show  $\mathcal{M} \subseteq \lambda_{\mathcal{M}^0}(\mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi))$  to show  $\mathcal{M} \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ . Hence, let  $P(\vec{a}_{\hat{P}}) \in \mathcal{M}$  and  $P(\vec{a}_{\hat{P}}) \notin \mathcal{M}^0(\Pi)$  (for if  $P(\vec{a}_{\hat{P}}) \in \mathcal{M}^0(\Pi)$ , the result is clear). We will show  $P(\vec{a}_{\hat{P}}) \in \{\text{Head}(r)\eta \mid (r, \eta) \in \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi)\}$ . Indeed, since  $\mathcal{M}^\infty(\Pi) = \mathcal{M}$  (i.e., since  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} (\varphi_{\Pi}^{\text{PRO}}(\vec{z}, \vec{S}) \wedge \varphi_{\Pi}^{\text{COMP}})$  and by Theorem 5.5 and [Zhang and Zhou 2010]), we have for some  $t > 1$ , rule  $r$ , and corresponding assignment  $\eta$ :

- (1)  $\text{Head}(r)\eta = P(\vec{a}_{\hat{P}})$ ;
- (2)  $\text{Pos}(r)\eta \subseteq \mathcal{M}^t(\Pi) \subseteq \mathcal{M}^\infty(\Pi) = \mathcal{M}$  and  $\text{Neg}(r)\eta \cap \mathcal{M} = \emptyset$ .

Then we have  $(r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}$ , which further implies  $(r, \eta) \in \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi)$  (i.e., since  $\text{Dom}(\mathcal{W}) = \Gamma(\Pi)_{\mathcal{M}}$ ). Therefore, we have  $P(\vec{a}_{\hat{P}}) \in \{\text{Head}(r)\eta \mid (r, \eta) \in \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi)\} \subseteq \lambda_{\mathcal{M}^0}(\mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi))$ . Hence, we have shown that  $\mathcal{M} \subseteq \lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}})$ .

This completes the proof of the theorem.  $\square$

PROPOSITION 5.8. *On finite structures, every FO preferred answer set program is precisely captured by an existential SO formula (i.e.,  $\exists$ SO).*

PROOF. We obtain an  $\exists$ SO formula  $\exists \vec{z} \psi_{(\Pi, <)}(\vec{z})$  from  $\exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  by substituting the SO formula  $\varphi_{\Pi}^{\text{TOTALOR}}(\vec{z})$ <sup>8</sup> for  $\varphi_{\Pi}^{\text{WELLOR}}(\vec{z}, \vec{S})$  in  $\varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  where  $\varphi_{\Pi}^{\text{TOTALOR}}(\vec{z})$  is given by:

$$\begin{aligned} & \bigwedge_{r \in \Pi} \forall \vec{x}_r \vec{y}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \wedge \varphi_r^{\text{GEN}}(\vec{y}_r) \wedge \vec{x}_r \neq \vec{y}_r \rightarrow (\widetilde{\langle}_{rr}(\vec{x}_r, \vec{y}_r) \vee \widetilde{\langle}_{rr}(\vec{y}_r, \vec{x}_r))) \\ & \wedge \bigwedge_{\substack{r_1, r_2 \in \Pi, \\ r_1 \neq r_2}} \forall \vec{x}_{r_1} \vec{x}_{r_2} (\varphi_{r_1}^{\text{GEN}}(\vec{x}_{r_1}) \wedge \varphi_{r_2}^{\text{GEN}}(\vec{x}_{r_2}) \rightarrow (\widetilde{\langle}_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \vee \widetilde{\langle}_{r_2 r_1}(\vec{x}_{r_2}, \vec{x}_{r_1}))), \end{aligned}$$

<sup>8</sup>Where “TOTALOR” stands for *total-order*.



which, in conjunction with Formulas (6), (7), and (8), expresses a *strict total-order* on  $\Gamma(\Pi)_{\mathcal{M}}$ . Then, since well-orders are strict total-orders on finite structures, the result follows.  $\square$

Proposition 5.8 actually indicates that under finite structures, FO preferred answer set programs can be captured by classical first-order logic, though the underlying vocabulary may be larger than that of the original preferred program in general. This is stated as the following corollary.

**COROLLARY 5.9.** *Let  $(\Pi, <)$  be an FO preferred answer set program, and  $\mathcal{A}$  is a finite structure on  $\tau(\Pi)$ . Then there exists a first-order sentence  $\Phi(\Pi, <)$  where  $\tau(\Pi) \subseteq \tau(\Phi(\Pi, <))$ , such that  $\mathcal{A}$  is a preferred answer set of  $(\Pi, <)$  iff there is a model  $\mathcal{M}$  of  $\Phi(\Pi, <)$  and  $\mathcal{A}$  is a restriction of  $\mathcal{M}$  on  $\tau(\Pi)$ .*

**PROOF.** For each predicate variable in the  $\exists$ SO formula mentioned in the proof of Proposition 5.8, we simply introduce a new predicate constant to replace that predicate variable, then such  $\exists$ SO formula reduces to a first-order sentence whose vocabulary is  $\tau(\Pi)$  plus those new predicate constants.  $\square$

Similarly to the ordered completion for FO normal logic programs [Asuncion et al. 2012a], by Corollary 5.9, we can develop a SAT based FO preferred normal logic programming solver as follows: for a given FO preferred normal logic program, we first translate it to a first-order sentence under a larger vocabulary, perform grounding for this first-order sentence by taking extensional database as input, and finally call an SAT solver to compute the models.

*Example 5.10.* On finite structures, let us consider once again the preferred program  $(\Pi_2, <_2)$  of Example 4.1<sup>9</sup>:

$$\begin{aligned} r_1 &: P(x_1) \leftarrow Q(x_1), \\ r_2 &: Q(x_2) \leftarrow, \\ r_1 &<_2 r_2. \end{aligned}$$

We will now use Theorem 5.7 and Proposition 5.8 to show that  $\Pi_2$  does not have a preferred answer set. Thus, given  $\Pi_2$ , we have  $\varphi_{\Pi_2}^{\text{PREF}}(\overrightarrow{\phantom{x}})$  to be:

$$\begin{aligned} &\forall x_1 (\varphi_{r_1}^{\text{GEN}}(x_1) \rightarrow \top) \\ &\wedge \forall x_2 (\varphi_{r_2}^{\text{GEN}}(x_2) \rightarrow \forall x_1 ((\varphi_{r_1}^{\text{GEN}}(x_1) \rightarrow \widetilde{\langle}_{r_1 r_2}(x_1, x_2)) \\ &\quad \wedge (\neg \varphi_{r_1}^{\text{GEN}}(x_1) \rightarrow (\varphi_{r_1}^{\text{POS}}(x_1) \vee \varphi_{r_2 r_1}^{\text{DEF}}(\overrightarrow{\phantom{x}}, x_2, x_1))))). \end{aligned} \quad (17)$$

Then, from  $\varphi_{\Pi_2}^{\text{PRO}}(\overrightarrow{\phantom{x}})$ , we further have:

$$\forall x_1 (\varphi_{r_1}^{\text{GEN}}(x_1) \rightarrow \exists x_2 (\widetilde{\langle}_{r_2 r_1}(x_2, x_1) \wedge x_1 = x_2)), \quad (18)$$

that is, the support for  $r_1$ . Then, since we also have

$$\forall x_1 x_2 (\widetilde{\langle}_{r_1 r_2}(x_1, x_2) \rightarrow \varphi_{r_1}^{\text{GEN}}(x_1) \wedge \varphi_{r_2}^{\text{GEN}}(x_2)) \quad (19)$$

by  $\varphi_{\Pi_2}^{\text{PRO}}(\overrightarrow{\phantom{x}})$  (i.e., via (8)), then (18) is equivalent to:

$$\forall x_1 (\varphi_{r_1}^{\text{GEN}}(x_1) \rightarrow \exists x_2 (\widetilde{\langle}_{r_2 r_1}(x_2, x_1) \wedge \varphi_{r_2}^{\text{GEN}}(x_2) \wedge x_1 = x_2)). \quad (20)$$

Then, from (17), we further get that (20) is equivalent to:

$$\forall x_1 (\varphi_{r_1}^{\text{GEN}}(x_1) \rightarrow \exists x_2 (\widetilde{\langle}_{r_2 r_1}(x_2, x_1) \wedge \varphi_{r_2}^{\text{GEN}}(x_2) \wedge \widetilde{\langle}_{r_1 r_2}(x_1, x_2) \wedge x_1 = x_2)), \quad (21)$$

<sup>9</sup>Note that for our purpose here, we relabel the variables in rules  $r_1$  and  $r_2$  to make them disjoint.

that is, simply adding the atom,  $\widetilde{\prec}_{r_1 r_2}(x_1, x_2)$ , to the consequent. Then by (6), that is, the transitive property, we further have that (21) is equivalent to

$$\forall x_1 (\varphi_{r_1}^{\text{GEN}}(x_1) \rightarrow \exists x_2 (\widetilde{\prec}_{r_2 r_1}(x_2, x_1) \wedge \varphi_{r_2}^{\text{GEN}}(x_2) \wedge \widetilde{\prec}_{r_1 r_2}(x_1, x_2) \wedge \widetilde{\prec}_{r_2 r_2}(x_2, x_2) \wedge x_1 = x_2)), \quad (22)$$

that is, adding the atom,  $\widetilde{\prec}_{r_2 r_2}(x_2, x_2)$ , to the consequent. Then, by the asymmetry as expressed by Formula (7) (i.e., which implies we cannot have  $\widetilde{\prec}_{r_2 r_2}(x_2, x_2)$ ), we further have that

$$\forall x_1 (\varphi_{r_1}^{\text{GEN}}(x_1) \rightarrow \perp) \equiv \forall x_1 \neg \varphi_{r_1}^{\text{GEN}}(x_1). \quad (23)$$

Hence, since  $\varphi_{r_1}^{\text{GEN}}(x_1) = Q(x_1)$ , then this implies that:

$$\forall x_1 \neg Q(x_1).$$

Therefore, since we also have  $\forall x_1 Q(x_1)$  by  $\varphi_{\Pi_2}^{\text{COMP}}$ , the sentence  $\exists \vec{z} \psi_{(\Pi_2, <_2)}(\vec{z})$  is unsatisfiable, which corresponds to the preferred program  $(\Pi_2, <_2)$  not having a preferred answer set as mentioned in Example 4.1.

**PROPOSITION 5.11.** *For an FO preferred program  $(\Pi, <)$  where  $< = \emptyset$  and a  $\tau(\Pi)$ -structure  $\mathcal{M}$ ,  $\mathcal{M} \models \exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  iff  $\mathcal{M}$  is an answer set of  $\Pi$ .*

**PROOF.** When  $< = \emptyset$ ,  $\exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}(\vec{z}, \vec{S})$  reduces to  $\exists \vec{z} \forall \vec{S} \varphi_{\Pi}(\vec{z}, \vec{S})$ , since  $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$  as given by:

$$\bigwedge_{r \in \Pi} \forall \vec{x}_r \left( \varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \bigwedge_{r' < r} \forall \vec{x}_{r'} \left( (\varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow \widetilde{\prec}_{r' r}(\vec{x}_{r'}, \vec{x}_r)) \wedge (\neg \varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow (\varphi_{r'}^{\text{POS}}(\vec{x}_{r'}) \vee \varphi_{r' r'}^{\text{DEF}}(\vec{z}, \vec{x}_r, \vec{x}_{r'})) \right) \right),$$

becomes

$$\bigwedge_{r \in \Pi} \forall \vec{x}_r (\varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \top) \equiv \top. \quad \square$$

## 6. COMPARISONS WITH OTHER PREFERRED LOGIC PROGRAMMING FRAMEWORKS

Handling preferences through propositional ASP has been studied by many researchers. Among the different proposals, the general fixpoint-type framework of handling preferences in logic programs proposed by Schaub and Wang [2003]; that we refer to SW framework whenever no confusion is caused, and the preferred logic programming approach Brewka and Eiter [1999] were considered to be most influential. In this section, we compare our approach developed in this article with these two typical preferred answer set programming frameworks in some detail.

### 6.1. Review of D-Preference Under Schaub and Wang's Fixpoint-Type Characterization

We will first show that our FO preferred semantics is a proper uplifting of Schaub and Wang's prescriptive type framework for propositional programs to the FO case. As illustrated in Schaub and Wang [2003], Schaub and Wang's framework is able to capture several preferred logic programming approaches. In this subsection, we consider the preferred logic programming framework proposed by Delgrande et al. [2003] under Schaub and Wang's characterization, which we will refer to D-preference.

First, we briefly review D-preference framework as reformalized in Schaub and Wang [2003]. With the same basic principle as that of the progression-based FO semantics,

the main idea in [Schaub and Wang 2003] is to characterize preferred answer sets by means of an inductive development (i.e., fixpoint-type) that respects (in some way) the given rule preference relations.

To formally define this inductive characterization, we first introduce the following notions. For a given propositional program  $\Pi$  of signature  $\tau(\Pi)$ , let  $X \subseteq \tau(\Pi)$  and  $Y \subseteq \tau(\Pi)$ . Then we define the *immediate consequence operator*  $\mathcal{T}_{\Pi, Y} X$  as follows:

$$\mathcal{T}_{\Pi, Y} X = \{\text{Head}(r) \mid r \in \Pi, \text{Pos}(r) \subseteq X \text{ and } \text{Neg}(r) \cap Y = \emptyset\}. \quad (24)$$

Now in terms of a preferred program  $(\Pi, <)$ , the immediate consequence operator  $\mathcal{T}_{\Pi, Y} X$  is extended in the SW framework by considering the preference relations among the rules so that  $\mathcal{T}_{(\Pi, <), Y} X$  (i.e., with the preference relations) is now defined as follows:

$$\begin{aligned} \mathcal{T}_{(\Pi, <), Y} X = \{ \text{Head}(r) \mid & (1) \text{Pos}(r) \subseteq X \text{ and } \text{Neg}(r) \cap Y = \emptyset; \\ & (2) \text{ there does not exist a rule } r' \in \Pi \text{ where } r' < r, \\ & r' \notin \text{Rule}(\text{Head}(X)), \text{ and } \text{Pos}(r') \subseteq Y \text{ and} \\ & \text{Neg}(r') \cap X = \emptyset \}, \end{aligned} \quad (25)$$

where (as mentioned in [Schaub and Wang 2003]),  $\text{Rule}(\cdot)$  is a bijective mapping among rule head and rules. That is, we have that  $\text{Rule}(\text{Head}(r)) = r$  and  $\text{Rule}(\{\text{Head}(r) \mid r \in R\}) = R$  (such mappings are defined by distinguishing different occurrences of head atoms). Note that the preference relations among the rules are considered in (25) by requiring that more preferred rules, that is,  $r'$  in (25), have effectively been applied.

Then to obtain the underlying inductive characterization, iterated applications of  $\mathcal{T}_{(\Pi, <), Y} X$  are written as  $\mathcal{T}_{(\Pi, <), Y}^i X$  for  $i \geq 0$ , where  $\mathcal{T}_{(\Pi, <), Y}^0 X = X$  and

$$\mathcal{T}_{(\Pi, <), Y}^{i+1} X = \mathcal{T}_{(\Pi, <), Y}(\mathcal{T}_{(\Pi, <), Y}^i X).$$

Also note that in the case of preference free programs, a set  $X$  of atoms is an answer set of  $\Pi$  iff  $\mathcal{T}_{\Pi, X}^\infty \emptyset = X$ , that is, we start from the empty set then gradually collect the heads of the reduced program  $\Pi^X$  until a fixpoint is achieved. Then, a set  $X$  of atoms is said to be a preferred answer set of the preferred program  $(\Pi, <)$  iff  $\mathcal{T}_{(\Pi, <), X}^\infty \emptyset = X$ .

## 6.2. The FO Preferred Progression Semantics on Propositional Programs

Now we consider the FO preferred progression semantics defined in Definition 3.1 under propositional case. As we discussed earlier, by grounding a FO preferred answer set program, we obtain a grounded preferred answer set program as defined in Definition 4.3. Then we have showed that there is a correspondence between the answer sets of the grounded preferred answer set program and the answer sets of the original FO preferred program, that is, Definition 4.5 and Theorem 4.7.

The following proposition shows that by applying our grounded preferred answer set definition to finite propositional preferred answer set programs, our semantics coincides to D-preference under Schaub and Wang's fixpoint-type characterization.

**PROPOSITION 6.1.** *Let  $(\Pi, <)$  be a propositional preferred program and  $X \subseteq \tau(\Pi)$ . Then  $\mathcal{T}_{(\Pi, <), X}^\infty \emptyset = X$  iff  $\text{Head}(\Delta^\infty(\Pi)_X) = X$ .*

**PROOF.** The result will be easily proved by induction that  $\mathcal{T}_{(\Pi, <), X}^i \emptyset = \text{Head}(\Delta^i(\Pi)_X)$  for all  $i \geq 0$ .  $\square$

## 6.3. Weak Preference Preserving Answer Sets

Now we will show that, similarly to SW framework which captures other important preferred approaches via a slight modification of the immediate consequence operator  $\mathcal{T}_{(\Pi, <), Y} X$ , our FO progression semantics also posses such features.

With respect to the term “groundedness” as used in Schaub and Wang [2003] and [Delgrande et al. 2003], a weak form of the preferred answer set is also proposed [Schaub and Wang 2003]. This so-called weak form has the property that in the application of a rule at any stage, it is enough for the head of a more preferred rule to have already appeared in the “earlier” collection of heads rather than the effective application of the (more preferred) rule itself. This is best illustrated in the following formal elaboration. Let  ${}^W\mathcal{T}_{(\Pi, <), Y}X$  (i.e., with a  $W$  left-superscript for “weak”) denote the consequence operator defined as follows:

$$\begin{aligned} {}^W\mathcal{T}_{(\Pi, <), Y}X = \{ & \text{Head}(r) \mid (1) \text{ Pos}(r) \subseteq X \text{ and } \text{Neg}(r) \cap Y = \emptyset; \\ & (2) \text{ there does not exist a rule } r' \in \Pi \text{ where } r' < r, \\ & \text{Head}(r') \notin X, \text{ and } \text{Pos}(r') \subseteq Y \text{ and} \\ & \text{Neg}(r') \cap X = \emptyset\}. \end{aligned} \quad (26)$$

Then, compared to the  $\mathcal{T}_{(\Pi, <), Y}X$  operator, the application of rules in (26) is considered if it is enough for the head of an “active” (i.e., feasible for application in the current stage) more preferred rule to have been derived in the earlier stages. As shown in Delgrande et al. [2003], this corresponds to the weak notion of “groundedness”. Similarly to that of SW preferred answer sets, a set of propositional atoms  $X$  is an answer set of a program  $\Pi$  iff  ${}^W\mathcal{T}_{(\Pi, <), X}^\infty \emptyset = X$  (with the iterative applications of  ${}^W\mathcal{T}_{(\Pi, <), Y}X$  defined as before).

At the FO level, this notion can be uplifted to our progression-based semantics.

*Definition 6.2.* Let  $(\Pi, <)$  be a preferred FO answer set program and  $\mathcal{M}$  a corresponding  $\tau(\Pi)$ -structure. We define

$$\begin{aligned} \Gamma_W^0(\Pi)_\mathcal{M} = \{ & (r, \eta) \mid (1) \text{ Pos}(r)\eta \subseteq \mathcal{M}^0(\Pi) \text{ and } \text{Neg}(r)\eta \cap \mathcal{M} = \emptyset; \\ & (2) \text{ there does not exist a rule } r' \in \Pi \text{ and an assignment } \eta' \text{ such} \\ & \text{that } r' < r, \text{ Pos}(r')\eta' \subseteq \mathcal{M} \text{ and } \text{Neg}(r')\eta' \cap \mathcal{M}^0(\Pi) = \emptyset\}; \end{aligned}$$

and for  $t \geq 0$ ,

$$\begin{aligned} \Gamma_W^{t+1}(\Pi)_\mathcal{M} = \Gamma_W^t(\Pi)_\mathcal{M} \cup \{ & (r, \eta) \mid (1) \text{ Pos}(r)\eta \subseteq \lambda_{\mathcal{M}^0}(\Gamma_W^t(\Pi)_\mathcal{M}) \text{ and } \text{Neg}(r)\eta \cap \mathcal{M} = \emptyset; \\ & (3) \text{ there does not exist a rule } r' \in \Pi \text{ and an assignment } \eta' \\ & \text{such that } r' < r, \text{ Head}(r')\eta' \notin \lambda_{\mathcal{M}^0}(\Gamma_W^t(\Pi)_\mathcal{M}), \text{ and} \\ & \text{Pos}(r')\eta' \subseteq \mathcal{M} \text{ and } \text{Neg}(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma_W^t(\Pi)_\mathcal{M}) = \emptyset\}. \end{aligned} \quad (27)$$

Let  $\Gamma_W^\infty(\Pi)_\mathcal{M} = \bigcup_{t=0}^\infty \Gamma_W^t(\Pi)_\mathcal{M}$ .

*Definition 6.3.* Let  $(\Pi, <)$  be a preferred FO program and  $\mathcal{M}$  a  $\tau(\Pi)$ -structure.  $\mathcal{M}$  is called a *weak preferred answer set* of  $(\Pi, <)$  iff  $\lambda_{\mathcal{M}^0}(\Gamma_W^\infty(\Pi)_\mathcal{M}) = \mathcal{M}$ .

Similarly to the original progression characterization of FO preferred answer sets, this also possesses the following important property.

**PROPOSITION 6.4.** *Let  $(\Pi, <)$  be a preferred program. If a  $\tau(\Pi)$ -structure  $\mathcal{M}$  is a weak preferred answer set of  $(\Pi, <)$ , then  $\mathcal{M}$  is an answer set of  $\Pi$ .*

The proof of this proposition is in a similar manner of the proof Proposition 3.5. Due to its tediousness, we omit it here.

As in the original definition of the FO preferred answer sets introduced in Section 3, this new weak notion of FO preferred answer sets can also be expressed in SO logic. More precisely, like the way we expressed the original definition of the FO preferred

answer set by the following SO formula in Section 5:

$$\exists \vec{z} \forall \vec{S} \varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z}, \vec{S}) = \varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z}) \wedge \varphi_{\Pi}^{\text{PRO}}(\vec{z}, \vec{S}) \wedge \varphi_{\Pi}^{\text{COMP}},$$

we can also capture this FO counterpart of the weak preferred answer sets via a slight modification of its  $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$  and  $\varphi_r^{\text{SUP}}(\vec{z}, \vec{x}_r)$  (which is in  $\varphi_{\Pi}^{\text{PRO}}(\vec{z}, \vec{S})$ ) subformulas, that is, the ones that explicitly deal with the rules' preference relations and groundedness, respectively.

For this purpose, let us now define  $\varphi_r^{\text{W-SUP}}(\vec{z}, \vec{x}_r)$  (here, “W-SUP” stands for weak support) to be a variant of  $\varphi_r^{\text{SUP}}(\vec{z}, \vec{x}_r)$  as the following formula:

$$\varphi_r^{\text{SUP}}(\vec{z}, \vec{x}_r) \vee \bigvee_{\substack{r' \in \Pi, \\ \text{Head}(r') = P(\vec{y})}} \exists \vec{x}_{r'} (\widetilde{<}_{r'r}(\vec{x}_{r'}, \vec{x}_r) \wedge \vec{x} = \vec{y}), \quad (28)$$

where we assume that  $\text{Head}(r)$  is of the form  $P(\vec{x})$ .

Generally speaking,  $\varphi_r^{\text{W-SUP}}(\vec{z}, \vec{x}_r)$  as defined in (28) expresses the weak form of groundedness by also allowing the sufficient condition that it is enough for the head of a generating rule to have been mentioned in the earlier stages, rather than strictly having its positive body derived. Then, by  $\varphi_{\Pi}^{\text{W-PRO}}(\vec{z}, \vec{S})$  (here “W-PRO” for weak progression), we now denote the formula obtained from  $\varphi_{\Pi}^{\text{PRO}}(\vec{z}, \vec{S})$  (see Definition 5.2) by replacing  $\varphi_r^{\text{SUP}}(\vec{z}, \vec{x}_r)$  with  $\varphi_r^{\text{W-SUP}}(\vec{z}, \vec{x}_r)$ .

In the following definition, we further provide the weak counterpart of the  $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$  formula that explicitly deals with the preference relations.

*Definition 6.5.* For an FO preferred program  $(\Pi, <)$ , define the formula  $\varphi_{(\Pi, <)}^{\text{W-PREF}}(\vec{z})$  as follows (here “W-PREF” stands for weak preference):

$$\bigwedge_{r \in \Pi} \forall \vec{x}_r \left( \varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \bigwedge_{r' < r} \forall \vec{x}_{r'} \left( \left( \varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow \widetilde{<}_{r'r}(\vec{x}_{r'}, \vec{x}_r) \right) \wedge \left( \neg \varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow \left( \varphi_{r'}^{\text{POS}}(\vec{x}_{r'}) \vee \varphi_{rr'}^{\text{DEF}}(\vec{z}, \vec{x}_r, \vec{x}_{r'}) \vee \varphi_{rr'}^{\text{HEAD}}(\vec{z}, \vec{x}_r, \vec{x}_{r'}) \right) \right) \right), \quad (29)$$

where  $\varphi_{r'}^{\text{POS}}(\vec{x}_{r'})$  and  $\varphi_{rr'}^{\text{DEF}}(\vec{z}, \vec{x}_r, \vec{x}_{r'})$  are defined as in Definition 5.6, and  $\varphi_{rr'}^{\text{HEAD}}(\vec{z}, \vec{x}_r, \vec{x}_{r'})$  stands for the formula

$$\bigvee_{\substack{r'' \in \Pi, \\ \text{Head}(r'') = P(\vec{y})}} \exists \vec{x}_{r''} (\widetilde{<}_{r''r'}(\vec{x}_{r''}, \vec{x}_r) \wedge \vec{x} = \vec{y}), \quad (30)$$

where we assume here that  $\text{Head}(r')$  is of the form  $P(\vec{x})$ .

Intuitively,  $\varphi_{(\Pi, <)}^{\text{W-PREF}}(\vec{z})$  differs from  $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$  in the sense that it is enough for a more preferred nongenerating rule to have its head already derived by some generating rule from the previous stages, that is, as enforced by  $\varphi_{rr'}^{\text{HEAD}}(\vec{z}, \vec{x}_r, \vec{x}_{r'})$  where  $r'$  is a nongenerating rule. This is in contrast with  $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$  since it requires all nongenerating rule to be either explicitly nonderivable as its positive body cannot be derived (i.e., via the  $\varphi_{r'}^{\text{POS}}(\vec{x}_{r'})$ ), or is already defeated by some rule derived from the earlier stages (i.e., via the  $\varphi_{rr'}^{\text{DEF}}(\vec{z}, \vec{x}_r, \vec{x}_{r'})$  formula). Thus, it is not difficult to see that the only real difference between  $\varphi_{(\Pi, <)}^{\text{W-PREF}}(\vec{z})$  and  $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$  is the addition of the other choice for the nongenerating rules (i.e., via  $\varphi_{rr'}^{\text{HEAD}}(\vec{z}, \vec{x}_r, \vec{x}_{r'})$ ).

As in the original definition of the FO preferred answer sets, in a similar way of the proof of Theorem 5.7, we can prove the following important property of the weak preferred answer sets.

**THEOREM 6.6.** *For an FO preferred program  $(\Pi, <)$  and a  $\tau(\Pi)$ -structure  $\mathcal{M}$ , we have that*

$$\mathcal{M} \models \exists \vec{z} \forall \vec{S} (\varphi_{(\Pi, <)}^{\text{W-PREF}}(\vec{z}) \wedge \varphi_{\Pi}^{\text{W-PRO}}(\vec{z}, \vec{S}) \wedge \varphi_{\Pi}^{\text{COMP}})$$

iff  $\mathcal{M}$  is a weak preferred answer set of  $(\Pi, <)$ .

The following proposition shows that the FO weak preferred answer sets can be captured by an  $\exists$ SO sentence on finite structures.

**PROPOSITION 6.7.** *On finite structures, every FO weak preferred answer set program is precisely captured by an  $\exists$ SO sentence.*

**PROOF.** Similarly to the proof of Proposition 5.8, we obtain an  $\exists$ SO formula from  $\exists \vec{z} \forall \vec{S} (\varphi_{(\Pi, <)}^{\text{W-PREF}}(\vec{z}) \wedge \varphi_{\Pi}^{\text{W-PRO}}(\vec{z}, \vec{S}) \wedge \varphi_{\Pi}^{\text{COMP}})$  by substituting the SO formula  $\varphi_{\Pi}^{\text{TOTALOR}}(\vec{z})$  for  $\varphi_{\Pi}^{\text{WELLOR}}(\vec{z}, \vec{S})$  in  $\varphi_{\Pi}^{\text{W-PRO}}(\vec{z}, \vec{S})$ . Then since total-orders are well-orders on finite sets, the result follows.  $\square$

#### 6.4. Brewka and Eiter's Framework

There is another important framework for propositional preferred logic programs proposed by Brewka and Eiter [1999]. This preference semantics was also captured in Schaub and Wang [2003] by means of a fixpoint-style characterization. Differently from both Delgrande et al.'s and Schaub and Wang's approaches, this fixpoint characterization requires that the "answer set property" has to be verified separately, and hence, it seems not feasible to define a direct progression-style preference semantics for its first-order case. To get over this unpleasant situation, we bypass the fixpoint characterization and instead go directly to a logical characterization.

To achieve this, we go by appealing to the *preference preserving* formulation as first proposed in Delgrande et al. [2003]. At the propositional case, it is given as follows. Let  $(\Pi, <)$  be a propositional preferred program and  $A$  an answer set of  $\Pi$ . Then  $A$  is also called a BE *preference preserving* answer set of  $(\Pi, <)$  iff there exists an enumeration  $\langle r_i \rangle_{i \in I}$  of  $\Gamma(\Pi)_{\mathcal{M}}$  satisfying the following properties. For every  $i, j \in I$ :<sup>10</sup>

- (1)  $r_i < r_j$  implies  $i < j$ ;
- (2)  $r' \notin \Gamma(\Pi)_{\mathcal{M}}$  and  $r' < r_i$  implies that:
  - (a)  $\text{Pos}(r') \not\subseteq A$ , or
  - (b)  $\text{Neg}(r') \cap \{\text{Head}(r_j) \mid j < i\} \neq \emptyset$ , or
  - (c)  $\text{Head}(r') \in A$ .

We now extend this notion to the FO case and under arbitrary structures as follows.

**Definition 6.8.** Assume  $(\Pi, <)$  to be a preferred FO program and  $\mathcal{M}$  an answer set of  $\Pi$ . Then we say that  $\mathcal{M}$  is also a BE preferred answer set of  $\Pi$  iff there exists a BE *preference preserving well-order*  $\mathcal{W} = (\Gamma(\Pi)_{\mathcal{M}}, <^{\mathcal{W}})$  of  $\Gamma(\Pi)_{\mathcal{M}}$  defined as follows:

- (1)  $(r_1, \eta_1), (r_2, \eta_2) \in \Gamma(\Pi)_{\mathcal{M}}$  and  $r_1 < r_2$  implies  $(r_1, \eta_1) <^{\mathcal{W}} (r_2, \eta_2)$ ;
- (2)  $(r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}$ ,  $(r', \eta') \notin \Gamma(\Pi)_{\mathcal{M}}$ , and  $r' < r$  implies that either:

<sup>10</sup>Note that the original definition as it was discussed by Delgrande et al. [2003] only concerned finite propositional programs.

- (a)  $Pos(r')\eta' \not\subseteq \mathcal{M}$  or
- (b)  $Neg(r')\eta' \cap \{Head(r'')\eta'' \mid (r'', \eta'') <^{\mathcal{W}} (r, \eta)\} \neq \emptyset$  or
- (c)  $Head(r')\eta' \in \mathcal{M}$ .

Now, to capture this in SO logic, we first define the following formula.

**Definition 6.9.** For an FO preferred program  $(\Pi, <)$ , define the formula  $\varphi_{(\Pi, <)}^{\text{BE-PREF}}(\vec{\succ}, \vec{S})$  as follows (here “BE-PREF” stands for Brewka-Eiter preference):

$$\bigwedge_{r \in \Pi} \forall \vec{x}_r \left( \varphi_r^{\text{GEN}}(\vec{x}_r) \rightarrow \bigwedge_{r' < r} \forall \vec{x}_{r'} \left( (\varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow \widetilde{<}_{r'}(\vec{x}_{r'}, \vec{x}_r)) \right. \right. \\ \left. \left. \wedge \left( \neg \varphi_{r'}^{\text{GEN}}(\vec{x}_{r'}) \rightarrow \left( \varphi_{r'}^{\text{-POS}}(\vec{x}_{r'}) \vee \varphi_{rr'}^{\text{DEF}}(\vec{\succ}, \vec{x}_r, \vec{x}_{r'}) \vee \varphi_{r'}^{\text{BE-HEAD}}(\vec{x}_{r'}) \right) \right) \right) \quad (31)$$

$$\wedge \bigwedge_{r_1, r_2, r_3 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} \vec{x}_{r_3} \left( \widetilde{<}_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \wedge \widetilde{<}_{r_2 r_3}(\vec{x}_{r_2}, \vec{x}_{r_3}) \rightarrow \widetilde{<}_{r_1 r_3}(\vec{x}_{r_1}, \vec{x}_{r_3}) \right) \quad (32)$$

$$\wedge \bigwedge_{r_1, r_2 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} \left( \widetilde{<}_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \rightarrow \neg \widetilde{<}_{r_2 r_1}(\vec{x}_{r_2}, \vec{x}_{r_1}) \right) \quad (33)$$

$$\wedge \bigwedge_{r_1, r_2 \in \Pi} \forall \vec{x}_{r_1} \vec{x}_{r_2} \left( \widetilde{<}_{r_1 r_2}(\vec{x}_{r_1}, \vec{x}_{r_2}) \rightarrow \varphi_{r_1}^{\text{GEN}}(\vec{x}_{r_1}) \wedge \varphi_{r_2}^{\text{GEN}}(\vec{x}_{r_2}) \right) \quad (34)$$

$$\wedge \varphi_{\Pi}^{\text{WELLOR}}(\vec{\succ}, \vec{S}), \quad (35)$$

such that  $\varphi_{r'}^{\text{-POS}}(\vec{x}_{r'})$  and  $\varphi_{rr'}^{\text{DEF}}(\vec{\succ}, \vec{x}_r, \vec{x}_{r'})$  are defined as in Definition 5.6,  $\varphi_{\Pi}^{\text{WELLOR}}(\vec{\succ}, \vec{S})$  is defined as in Definition 5.2, and  $\varphi_{r'}^{\text{BE-HEAD}}(\vec{x}_{r'})$  stands for the formula

$$P(\vec{x}), \quad (36)$$

where we assume here that  $Head(r')$  is of the form  $P(\vec{x})$ .

Let us take a closer look at Definition 6.9. In a nutshell, about Formula (31), for each generating rule  $(r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}$ , we have that each other rule  $(r', \eta')$  (can be nongenerating) for which  $r' < r$  is considered into two possible cases. The first case is where  $(r', \eta')$  is also a generating rule itself. In this case, we require that  $(r', \eta')$  must have to be placed earlier than  $(r, \eta)$  in the well-order, that is, as enforced by  $\widetilde{<}_{r'}(\vec{x}_{r'}, \vec{x}_r)$ . On the other hand, in the case that  $(r', \eta')$  is a nongenerating rule, we must have that either:  $(r', \eta')$  cannot possibly be derived (since  $Pos(r')\eta' \not\subseteq \mathcal{M}$  as encoded by  $\varphi_{r'}^{\text{-POS}}(\vec{x}_{r'})$ ), as it is already defeated in the well-order (via  $\varphi_{rr'}^{\text{DEF}}(\vec{\succ}, \vec{x}_r, \vec{x}_{r'})$ ), or that  $Head(r')\eta' \in \mathcal{M}$  (as enforced by the formula  $\varphi_{r'}^{\text{BE-HEAD}}(\vec{x}_{r'}) = P(\vec{x})$  where we assume that  $Head(r') = P(\vec{x})$ ). Finally, Formulas (32), (33), (34), and (35) enforces the well-order (where  $\varphi_{\Pi}^{\text{WELLOR}}(\vec{\succ}, \vec{S})$  is defined as in Definition 5.2).

**THEOREM 6.10.** *Let  $(\Pi, <)$  be an FO preferred and  $\mathcal{M}$  be a  $\tau(\Pi)$ -structures. Then we have that*

$$\mathcal{M} \models \exists \vec{<}_1 \vec{<}_2 \forall \vec{S}_1 \vec{S}_2 \left( \varphi_{(\Pi, <)}^{\text{BE-PREF}}(\vec{<}_1, \vec{S}_1) \wedge \varphi_{\Pi}^{\text{PRO}}(\vec{<}_2, \vec{S}_2) \wedge \varphi_{\Pi}^{\text{COMP}} \right)$$

*iff  $\mathcal{M}$  is a BE preferred answer set of  $(\Pi, <)$ .*

Note that since the answer sets of the BE preferred framework have to be verified separately, the “ordering predicates” of  $\varphi_{(\Pi, <)}^{\text{BE-PREF}}(\vec{<}_1, \vec{S}_1)$  are independent from that of  $\varphi_{\Pi}^{\text{PRO}}(\vec{<}_2, \vec{S}_2)$ . This is in contrast with both the SW and the “weak” SW since they do not dispose of the groundedness requirement.

PROOF.

$$\mathcal{M} \models \exists \vec{<}_1 \vec{<}_2 \forall \vec{S}_1 \vec{S}_2 (\varphi_{(\Pi, <)}^{\text{BE-PREF}}(\vec{<}_1, \vec{S}_1) \wedge \varphi_{\Pi}^{\text{PRO}}(\vec{<}_2, \vec{S}_2) \wedge \varphi_{\Pi}^{\text{COMP}})$$

iff  $\mathcal{M} \models \exists \vec{<}_2 \forall \vec{S}_2 (\varphi_{\Pi}^{\text{PRO}}(\vec{<}_2, \vec{S}_2) \wedge \varphi_{\Pi}^{\text{COMP}})$  and  $\mathcal{M} \models \exists \vec{<}_1 \forall \vec{S}_1 \varphi_{(\Pi, <)}^{\text{BE-PREF}}(\vec{<}_1, \vec{S}_1)$  iff  $\mathcal{M}$  is an answer set of  $\Pi$  (by Theorem 5.5) and there exists a BE preference preserving well-order of  $\Gamma(\Pi)_{\mathcal{M}}$  (by Definitions 6.8 and 6.9) iff  $\mathcal{M}$  is a BE preferred answer set of  $(\Pi, <)$ .  $\square$

PROPOSITION 6.11. *On finite structures, the BE preferred answer set framework can also be captured by an  $\exists$ SO sentence.*

PROOF. As in the proofs of Propositions 5.8 and 6.7, simply replace the occurrences of  $\varphi_{\Pi}^{\text{WELLOR}}(\vec{<}, \vec{S})$  by  $\varphi_{\Pi}^{\text{TOTALOR}}(\vec{<})$ . Then since total-orders are well-orders on finite sets, the result follows.  $\square$

### 6.5. Why Preferred First-Order ASP Further Discussions

In previous subsections, we have shown that by restricting our approach to the propositional case, our progression-based semantics for preferred answer set programming is reduced to Delgrande et al.'s preference approach under Schaub and Wang's fixpoint-style characterization [Delgrande et al. 2003]. We also showed how both Schaub and Wang's and Brewka and Eiter's frameworks [Schaub and Wang 2003; Brewka and Eiter 1999] can be extended to the first-order cases respectively.

Like the case of first-order ASP, in preferred first-order ASP, arbitrary domains are considered, while both finite and infinite domains are allowed. However, in practice we usually only work on finite domains. Then, a key question is: from a practical viewpoint, can we just consider propositional preferred ASP, while grounding is used to handle variables occurring in the underlying programs?

In our view, preferred first-order ASP not only offers a more succinct method to formalize complex problem domains involving preference reasoning, but also provides an alternative to implement effective preferred ASP solvers on finite domains. Note that Corollary 5.9 shows that in our framework, every preferred first-order program can be represented by a first-order sentence on finite structures. With this result, we can possibly develop a preferred first-order ASP solver as follows: (1) for a given preferred program, we can translate it into the corresponding first-order sentence; (2) do necessary formula simplifications; (3) by taking the extensional instances, ground the simplified first-order sentence; and finally (4) call an SAT solver to compute the models, from which the preferred answer sets of the original program are extracted.

As we have demonstrated in our recent work on ordered completion of first-order normal logic programs [Asuncion et al. 2012a], such SAT based first-order ASP solver avoids the grounding of the underlying program directly, instead, it grounds the corresponding first-order formula where many useful optimization techniques can be used. It has noticeable better performance on very large problem instances compared to current existing ASP solvers. We believe that by extending this technology to preferred first-order normal logic programs, we can eventually develop an effective preferred first-order ASP solver with practical applications.

## 7. CONCLUDING REMARKS

Our framework of preferred first-order ASP generalizes the progression semantics for FO ASP by Zhang and Zhou [2010] by incorporating preference, while it also extends Delgrande et al.'s semantics [Delgrande et al. 2003] for preferred propositional ASP to the FO case. On the other hand, the logical characterization of our preferred first-order ASP further reveals the expressive power of our framework from a classical logic viewpoint, and also generalizes Schub and Wang's and Brewka and Eiter's preference



semantics [Schaub and Wang 2003; Brewka and Eiter 1999] to the corresponding first-order cases.

Several related issues are left for our future work. First, it is our current task to develop a preferred first-order ASP solver under the progression-based preference semantics. As discussed in Section 6.5, our approach will be based on the logical characterization result proved in Section 5.

Second, as an application of preferred ASP, there have been some works on updates through preferred logic programs, for instance, Zhang [2006]. It would be an interesting topic to study this problem on the first-order level via preferred FO ASP. Currently, we are considering to use the proposed framework to specify finite structure updates, where the entire update process may be represented by a preferred FO answer set program.

Another interesting work is to extend the preferred FO normal answer set programs to preferred FO disjunctive answer set programs. The recent work of Zhou and Zhang [2011] extends the progression-based semantics of FO normal logic programs to FO disjunctive logic programs. This new semantics may provide a possibility to develop a similar progression-based preference semantics for FO disjunctive programs.

Finally, it is also an important work to investigate the relationships between our preferred FO ASP with other preferred first-order nonmonotonic reasoning formalisms such as prioritized circumscription. Prioritized circumscription [Lifschitz 1985; McCarthy 1986] is an alternative way of introducing circumscription by means of an ordering on tuples of predicates satisfying an axiom (can be any arbitrary first-order sentence). Hence, prioritized circumscription differs from ours in that we do not relate to any ordering of the tuples of predicates, but rather, we relate directly to the so-called “ordering on formulas”, that is, as represented by the universal closures of rules. Thus, our approach is more of a prescriptive analog on an FO level. However, it would be interesting to know whether these two FO preference formulations have certain in-depth connections under some circumstances.

## REFERENCES

- V. Asuncion, F. Lin, Y. Zhang, and Y. Zhou. 2012a. Ordered completion for first-order logic programs on finite structures. *Artif. Intell.* 177–179, 1–24.
- V. Asuncion, F. Lin, Y. Zhang, and Y. Zhou. 2012b. Ordered completion for logic programs with aggregates. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'12)*. 691–697.
- C. Baral. 2003. *Knowledge Representation, Reasoning and Declarative Problem Solving*. MIT Press. 39
- G. Brewka. 2006. Preferences in answer set programming. In *Current Topics in Artificial Intelligence*. Springer, 1–10.
- G. Brewka and T. Eiter. 1999. Preferred answer sets for extended logic programs. *Artif. Intell.* 109, 297–356.
- Y. Chen, F. Lin, Y. Zhang, and Y. Zhou. 2011. Loop-separable programs and their first-order definability. *Artif. Intell.* 175, 890–913.
- J. Delgrande, T. Schaub, and H. Tompits. 2003. A framework for compiling preferences in logic programs. *Theory Pract. Logic Programm.* 3, 129–187.
- J. Delgrande, T. Schaub, and H. Tompits. 2004. A classification and survey of preference handling approaches in nonmonotonic reasoning. *Comput. Intell.* 20, 308–334.
- H. Enderton. 1977. *Elements of Set Theory*. Academic Press.
- P. Ferraris, J. Lee, and V. Lifschitz. 2011. Stable models and circumscription. *Artif. Intell.* 175, 236–263.
- M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. 2007. Conflict-driven answer set solving. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'07)*. 386–391.
- M. Gelfond and V. Lifschitz. 1988. A stable model semantics for logic programming. In *Proceedings of the International Conference for Logic Programming*. 1070–1080.
- J. Lee and Y. Meng. 2009. On reductive semantics of aggregates in answer set programming. In *Proceedings of the International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'09)*. 182–195.

- J. Lee and R. Palla. 2010. Situation calculus as answer set programming. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI'10)*. 309–314.
- V. Lifschitz. 1985. Closed-world databases and circumscription. *Artif. Intell.* 27, 229–235.
- F. Lin and Y. Zhou. 2011. From answer set logic programming to circumscription via logic of GK. *Artif. Intell.* 175, 264–277.
- J. McCarthy. 1986. Applications of circumscription to formalizing common-sense knowledge. *Artif. Intell.* 28, 89–116.
- T. Schaub and K. Wang. 2003. A semantic framework for preference handling in answer set programming. *Theory Pract. Logic Programming* 3, 569–607.
- Y. Zhang. 2006. Logic program-based updates. *ACM Trans. Comput. Log.* 7, 421–472.
- Y. Zhang and Y. Zhou. 2010. On the progression semantics and boundedness of answer set programs. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning*. 518–527.
- Y. Zhou and Y. Zhang. 2011. Progression semantics for disjunctive logic programs. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI'11)*. 286–291.

Received July 2011; revised August 2012, April 2013; accepted September 2013