Preferred First-order Answer Set Programs

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

In this paper, 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

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 formalisms, 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, e.g., [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, i.e., [Gebser et al. 2007].

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 paper, 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 paper, we make the following original contributions towards the development of preferred first-order answer set programming:

Authors’ address: V. Asuncion, Y. Zhang, 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.

*: The corresponding author.

Permission to make digital or hard copies of part or all 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 show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© YYYY ACM 1539-9087/YYY/01-ARTA $15.00
DOI: http://dx.doi.org/10.1145/0000000.0000000

- 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 paper is organized as follows. Section 2 provides logical preliminaries that we will need throughout this paper. 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 paper with some remarks.

2. BASIC CONCEPTS
In this section, we first introduce necessary logical concepts and notions that we will need in this paper, 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 \( A \) of vocabulary \( \tau \) (or a \( \tau \)-structure) is a tuple \( A = (A, c^A_1, \ldots, c^A_m; P^A_1, \ldots, P^A_n) \), where \( A \) is a nonempty set called the domain of \( A \) (sometimes we use \( \text{Dom}(A) \) to denote \( A \)’s domain), \( c^A_i \) (\( 1 \leq i \leq m \)) is an element in \( A \) for every constant \( c_i \) in \( \tau \), and \( P^A_j \) (\( 1 \leq j \leq n \)) is a \( k \)-ary relation over \( A \) for every \( k \)-ary predicate \( P_j \) in \( \tau \). \( P^A_j \) is also called the interpretation of \( P_j \) in \( A \). A structure is finite if its domain is a finite set. In this paper, 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 \( A \) of \( \tau_1 \), we say that the restriction of \( A \) on \( \tau_2 \), denoted by \( A \mid \tau_2 \), is a structure of \( \tau_2 \) whose domain is the same as \( A \)’s, and for each constant \( c \) and relation symbol \( R \) in \( \tau_2 \), \( c^A \) and \( R^A \) are in \( A \mid \tau_2 \). On the other hand, if we are given a structure \( A' \) of \( \tau_2 \), a structure \( A \) of \( \tau_1 \) is an expansion of \( A' \) to \( \tau_1 \), if \( A \) has the same domain of \( A' \) 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 \( c^{\mathcal{A}} \), where \( c \) is an arbitrary constant. Let \( P(\overline{x}) \) be an atom and \( \eta \) an assignment in structure \( \mathcal{A} \). We also use \( P(\overline{x}) \eta \in \mathcal{A} \) to denote \( \eta(\overline{x}) \in P^{\mathcal{A}} \). 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 \( \overline{x} \) be the set of free variables occurring in a formula \( \phi \). Then, the satisfaction relation only relies on the assignment of \( \overline{x} \). In this case, we write \( \mathcal{A} \models \phi(\overline{x}/\overline{a}) \) to denote the satisfaction relation, where \( \overline{a} \) is a tuple of elements in \( \mathcal{A} \). 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 \( \mathcal{A} \cup \{ P(\overline{x}) \eta \} \) 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., \( P^{\mathcal{A}} \) to \( P \cup \{ \eta(\overline{x}) \} \)). If \( \mathcal{P} \) is a set of predicates, we simply write \( \mathcal{P} \eta \subseteq \mathcal{A} \) if for each \( P \in \mathcal{P}, P(\overline{x}) \eta \in P^{\mathcal{A}} \). Similarly, we write \( \mathcal{P} \eta \cap \mathcal{A} = \emptyset \) if for each \( P \in \mathcal{P}, P(\overline{x}) \eta \cap P^{\mathcal{A}} = \emptyset \).

Let \( \mathcal{A}_1 \) and \( \mathcal{A}_2 \) be two \( \tau \)-structures sharing the same domain, i.e., \( \text{Dom}(\mathcal{A}_1) = \text{Dom}(\mathcal{A}_2) \), and for each constant \( c \) in \( \tau \), \( c^{\mathcal{A}_1} = c^{\mathcal{A}_2} \). By \( \mathcal{A}_1 \subseteq \mathcal{A}_2 \), we simply mean that for each predicate \( P \) in \( \tau \), \( P^{\mathcal{A}_1} \subseteq P^{\mathcal{A}_2} \). By \( \mathcal{A}_1 \subseteq \mathcal{A}_2 \), we mean that \( \mathcal{A}_1 \subseteq \mathcal{A}_2 \) but \( \mathcal{A}_2 \not\subseteq \mathcal{A}_1 \). 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 \( \tau \), \( P^{\mathcal{A}_1 \cup \mathcal{A}_2} = P^{\mathcal{A}_1} \cup P^{\mathcal{A}_2} \).

2.2. An overview of first-order answer set programs

In this paper, 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 paper. A rule is of the form:

\[
\alpha \leftarrow \beta_1, \ldots, \beta_k, \text{not } \gamma_1, \ldots, \text{not } \gamma_l,
\]

where \( \alpha \) is a proper atom or the falsity \( \bot \) (i.e., an empty head), and \( \beta_1, \ldots, \beta_k, \gamma_1, \ldots, \gamma_l \) \((k, l \geq 0)\) are atoms. Here, \( \alpha \) is called the head, \( \{\beta_1, \ldots, \beta_k\} \) the positive body and \( \{\text{not } \gamma_1, \ldots, \text{not } \gamma_l\} \), the negative body of the rule, respectively. Sometimes, for convenience we use \( \text{Head}(r) \), \( \text{Pos}(r) \) and \( \text{Neg}(r) \) to denote the sets \( \{\alpha\} \), \( \{\beta_1, \ldots, \beta_k\} \) and \( \{\gamma_1, \ldots, \gamma_l\} \), 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 \( \Pi \), \( \tau_{\text{int}}(\Pi) \) to denote the vocabulary containing all intentional predicates in \( \Pi \), and \( \tau_{\text{ext}}(\Pi) \) to denote the vocabulary containing all extensional predicates and constants in \( \Pi \). We also use notions \( \mathcal{P}(\Pi), \mathcal{P}_{\text{int}}(\Pi), \) and \( \mathcal{P}_{\text{ext}}(\Pi) \) to denote the sets of all predicates, intentional, and extensional predicates in \( \Pi \), respectively. A proper atom \( P(\overline{t}) \) is extensional (intentional) if \( P \) is extensional (intentional). Furthermore, given a set \( \mathcal{X} \) of atoms, we denote by \( \text{Pred}(\mathcal{X}) \) the set of all the predicate symbols occurring in \( \mathcal{X} \). For instance, given a rule \( r \) of the form (1), \( \text{Pred}(\text{Pos}(r)) \) denotes the set of all the predicate symbols occurring in the set \( \{\beta_1, \ldots, \beta_k\} \). Obviously, we use \( \text{Head}(\Pi), \text{Pos}(\Pi) \) and \( \text{Neg}(\Pi) \) to denote the unions of all \( \text{Head}(r), \text{Pos}(r) \) and \( \text{Neg}(r) \) 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:

\[
\bigwedge_{r \in \Pi} \forall \overline{x}_r (\beta_1 \land \ldots \land \beta_k \land \neg \gamma_1 \land \ldots \land \neg \gamma_l \rightarrow \alpha),
\]

where \( \alpha \leftarrow \beta_1, \ldots, \beta_k, \text{not } \gamma_1, \ldots, \text{not } \gamma_l \).
where for a rule \( r \in \Pi \), \( \bar{x}_r \) denotes the tuple of distinguishable variables occurring in \( r \). The formula \( \hat{\Pi} \) is usually referred to as the universal closure of \( \Pi \). Then, a \( \tau(\Pi) \)-structure \( A \) is an answer set of \( \Pi \) iff \( A \) is a model of the following SO sentence \( \varphi \):

\[
\hat{\Pi} \land \exists \bar{u} ( \bar{u} < \bar{P} \land \hat{\Pi}^*(\bar{u})),
\]

(3)

where:

- \( \bar{P} = P_1 \ldots P_k \) denotes the tuple of intentional predicates \( P_1, \ldots, P_k \) of \( \Pi \);
- \( \bar{u} = u_1 \ldots u_k \) denotes the tuple of predicates variables \( u_1, \ldots, u_k \) such that for each \( i \) (\( 1 \leq i \leq k \)), the arity of \( u_i \) matches that of \( P_i \);
- \( \bar{u} < \bar{P} \) is an SO formula:

\[
\bigwedge_{P_i \in \bar{P}} \forall \bar{x} (u_i(\bar{x}) \rightarrow P_i(\bar{x})) \land \neg \bigwedge_{P_i \in \bar{P}} \forall \bar{x} (P_i(\bar{x}) \rightarrow u_i(\bar{x}));
\]

- and finally, \( \hat{\Pi}^*(\bar{u}) \) is an SO formula obtained from \( \hat{\Pi} \) by replacing every \( P_i \) occurring among \( \alpha, \beta_1, \ldots, \beta_k \) in formula (2) with the predicate variable \( u_i \) (\( 1 \leq i \leq k \)) on \( \Pi \).

The above 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 above 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:

\[
P(x) \leftarrow \neg Q(x),
\]

\[
Q(x) \leftarrow Q(x).
\]

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

3. A PROGRESSION-BASED SEMANTICS

Let \( \Pi \) be an FO program, and \( \prec \subseteq \Pi \times \Pi \) an irreflexive and transitive relation. Then \( (\Pi, \prec) \) is called a preferred FO program. Intuitively, if \( r_1, r_2 \in \Pi \) and \( (r_1, r_2) \in \prec \) (we may write \( r_1 \prec r_2 \)), 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:

\[
\Sigma(\Pi, \mathcal{M}) = \{ (r, \eta) \mid r \in \Pi \text{ and } \eta : \text{Term}(r) \rightarrow \text{Dom}(\mathcal{M}) \}.
\]

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:

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

where \( c_i^{M^0} = c_i^M \) for each constant \( c_i \) of \( \tau(\Pi) \) (\( 1 \leq i \leq r \)), \( P_j^{M^0} = P_j^M \) for each extensional predicate \( P_j \) in \( \tau_{ex}(\Pi) \) (\( 1 \leq j \leq s \)), and \( Q_k^{M^0} = \emptyset \) for each intentional predicate \( Q_k \) in \( \tau_{int}(\Pi) \).
and stage, generated from consequence operator (i.e., operator generated should be satis-
ified by preferred generally speaking, this strategy ensures that when we consider a rule for application, other
earlier; and (b) these rules' positive bodies are already satis-
fied by the structure obtained from the previous evaluation stages, and the
negative body should not be satis-
fied by the structure from the previous stage.

Secondly, we consider the structure \( \lambda_{\mathcal{M}\eta}(\Gamma^t(\Pi)_{\mathcal{M}}) \), 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 \( \lambda_{\mathcal{M}\eta}(\Gamma^t(\Pi)_{\mathcal{M}}) \) 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 \( \lambda_{\mathcal{M}\eta}(\Gamma^t(\Pi)_{\mathcal{M}}) \) 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 \( T_{(\Pi, <)_{\mathcal{Y}}} \) 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
\[
\begin{align*}
  r_1: & \text{Flies}(x) \leftarrow \text{Bird}(x), \text{not Cannot\_fly}(x), \\
  r_2: & \text{Cannot\_fly}(x) \leftarrow \text{Penguin}(x), \text{not Flies}(x),
\end{align*}
\]

We consider a finite structure \(\mathcal{M}\), where
\[
\text{Dom}(\mathcal{M}) = \{\text{cody, tweety}\},
\]
\[
\text{Bird}^\mathcal{M} = \{\text{cody, tweety}\},
\]
\[
\text{Penguin}^\mathcal{M} = \{\text{tweety}\},
\]
\[
\text{Flies}^\mathcal{M} = \{\text{cody}\},
\]
\[
\text{Cannot\_fly}^\mathcal{M} = \{\text{tweety}\}.
\]

In \(\Pi_1\), \text{Bird} and \text{Penguin} are extensional predicates, and \text{Flies} and \text{Cannot\_fly} are intentional. According to Definition 3.1, we have
\[
\begin{align*}
  \Gamma^0(\Pi_1)_{\mathcal{M}} &= \{(r_2, \eta) \mid \eta : \{x\} \rightarrow \{\text{tweety}\}\}, \\
  \Gamma^2(\Pi_1)_{\mathcal{M}} &= \Gamma^1(\Pi_1)_{\mathcal{M}} = \Gamma^0(\Pi_1)_{\mathcal{M}} \cup \{(r_1, \eta') \mid \eta' : \{x\} \rightarrow \{\text{cody}\}\}.
\end{align*}
\]

Then from Definition 3.2, we have
\[
\lambda_{\mathcal{M}^0}(\Gamma^2(\Pi_1)_{\mathcal{M}}) = \mathcal{M}^0 \cup \{\text{Cannot\_fly(tweety), Flies(cody)}\} = \mathcal{M}.
\]

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

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 \(t \geq 0\), \(\lambda_{\mathcal{M}^t}(\Gamma^t(\Pi)_{\mathcal{M}}) = \mathcal{M}^{t+1}(\Pi)\). Therefore, \(\lambda_{\mathcal{M}^t}(\Gamma^\infty(\Pi)_{\mathcal{M}}) = \mathcal{M}^\infty(\Pi)\). Then from Theorem 1 by Zhang and Zhou [2010], we conclude that \(\mathcal{M}\) is a preferred answer set of \((\Pi, <)\) iff \(\mathcal{M}\) is an answer set of \(\Pi\).

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. See appendix.

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. Firstly, 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:
\[ r_1 : P(x) \leftarrow Q(x), \]
\[ r_2 : Q(x) \leftarrow, \]
\[ r_1 < r_2. \]

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 \( \lambda_{\mathcal{M}}(\Gamma^\infty(\Pi_2)_\mathcal{M}) = \mathcal{M}' \) 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:
\[ r_1 : P(y) \leftarrow P(x), Q(x), \]
\[ r_2 : P(x) \leftarrow Q(x), \]
\[ r_1 <_3 r_2. \]

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 \( P(\overline{x}) \in \mathcal{P}_{\text{ext}}(\Pi) \cup \mathcal{P}_{\text{int}}(\Pi) \) 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 non-preferred programs usually are not applicable to preferred programs. Consider the following program:
\[ r_1 : P(x) \leftarrow Q(x), \]
\[ r_2 : Q(x) \leftarrow R, \]
\[ r_3 : R \leftarrow, \]
\[ r_1 < r_3. \]

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
\[ r_1 : P(x) \leftarrow Q(x), \]
\[ r_2' : Q(x) \leftarrow, \]
\[ r_3 : R \leftarrow, \]
\[ r_1 < r_3. \]

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 (Π₄, <₄):

\[
\begin{align*}
r_1 & : P(x) \leftarrow Q(y), \\
r_2 & : P(z) \leftarrow Q(z), \\
r_1 & <₄ r_2.
\end{align*}
\]

If we simply ground Π 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 Π with a corresponding tag predicate, as showed in the following.

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

\[r_i : α ← β_1, \ldots, β_k, \text{not} γ_1, \ldots, \text{not} γ_l,\]

Π' contains its tagged rule of the form:

\[r'_i : α ← β_1, \ldots, β_k, \text{not} γ_1, \ldots, \text{not} γ_l, \text{Tag}_i(\overline{x}),\]

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

Let \((Π, <)\) be a preferred answer set program on vocabulary \(τ(Π)\) and \((Π', <')\) the tagged preferred program of \((Π, <)\) on vocabulary \(τ(Π') = τ(Π) \cup \{\text{Tag}_1, \ldots, \text{Tag}_k\}\). Given a \(τ\)-structure \(ℳ\), we construct an expansion \(ℳ'\) of \(ℳ\) to \(τ(Π')\) as follows:

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

Let \(r' \in Π'\) and \(η\) an assignment on structure \(ℳ'\) of \(τ(Π')\). We use the notation \(r' η\) to denote the ground instance of \(r'\) based on \(η\). Now we are ready to define the grounding of a preferred answer set program.

Definition 4.3. Let \((Π, <)\) be a preferred answer set program, \(ℳ\) a structure of \(τ(Π)\), \((Π', <')\) the tagged answer set program of \((Π, <)\), and \(ℳ'\) the expansion of \(ℳ\) to \(τ(Π')\) as described above. We say that a pair \((\text{Ground}(Π)_ℳ, <')\) is the grounded preferred answer set program of \((Π, <)\) based on \(ℳ\), if

1. \(\text{Ground}(Π)_ℳ = \{r^* \mid \text{Head}(r)η ← Body(r)η, \text{Tag}_i(\overline{x})η, \text{where } r \in Π \text{ and } η \text{ is an assignment on } ℳ'\} \cup \{P(\overline{a}) ← | P \in P_{\text{ext}}(Π) \text{ and } P(\overline{a}) \in P^{ℳ}\} \cup \{\text{Tag}_i(\overline{a}) ← | \text{Tag}_i(\overline{a}) \in Tag_i^{ℳ'} (1 ≤ i ≤ k)\} \cup \{a = a ← | a \text{ is an element of } \text{Dom}(ℳ')\}\);
2. \(<' = \{(r^*_i, r^*_j) \mid r^*_i, r^*_j \in \text{Ground}(Π)_ℳ \text{ for some } r^*_k, r^*_l \in Π' \text{ such that there exist assignments } η \text{ and } η' \text{ on } ℳ' \text{ satisfying } r^*_i = r^*_k η, r^*_j = r^*_l η' \text{ and } r^*_k <' r^*_l\}\).

---

1We assume that \(Π\) contains rules \(r_1, \ldots, r_k\).

2Here, we view \(a = a\) as a propositional atom.
Example 4.4. We consider the preferred program \((\Pi_1, <_1)\) from Example 3.3 again. Under the structure \(M\) given in Example 3.3, the grounded preferred program of \((\Pi_1, <_1)\) is as follows:

\[
\begin{align*}
\sigma_1^* & : \text{Flies}(\text{cody}) \leftarrow \text{Bird}(\text{cody}), \text{not Cannot}_\star(\text{fly}(\text{cody})), \text{Tag}(\text{cody}), \\
\sigma_2^* & : \text{Flies}(\text{tweety}) \leftarrow \text{Bird}(\text{tweety}), \text{not Cannot}_\star(\text{fly}(\text{tweety})), \text{Tag}(\text{tweety}), \\
\sigma_3^* & : \text{Cannot}_\star(\text{fly}(\text{cody})) \leftarrow \text{Penguin}(\text{cody}), \text{not Flies}(\text{cody}), \text{Tag}(\text{cody}), \\
\sigma_4^* & : \text{Cannot}_\star(\text{fly}(\text{tweety})) \leftarrow \text{Penguin}(\text{tweety}), \text{not Flies}(\text{tweety}), \text{Tag}(\text{tweety}), \\
\sigma_5^* & : \text{Bird}(\text{cody}) \leftarrow, \\
\sigma_6^* & : \text{Bird}(\text{tweety}) \leftarrow, \\
\sigma_7^* & : \text{Penguin}(\text{tweety}) \leftarrow, \\
\sigma_8^* & : \text{Tag}(\text{cody}) \leftarrow, \\
\sigma_9^* & : \text{Tag}(\text{tweety}) \leftarrow, \\
\sigma_{10}^* & : \text{Tag}(\text{cody}) \leftarrow, \\
\sigma_{11}^* & : \text{Tag}(\text{tweety}) \leftarrow, \\
\sigma_{12}^* & : \text{cody} = \text{cody} \leftarrow, \\
\sigma_{13}^* & : \text{tweety} = \text{tweety} \leftarrow, \\
\sigma_{14}^* & : \text{tweety} ^{a}\text{tweety} ^{r} \\
\sigma_{15}^* & : \text{tweety} ^{a}\text{tweety} ^{r} \\
\sigma_{16}^* & : \text{tweety} ^{a}\text{tweety} ^{r} \\
\sigma_{17}^* & : \text{tweety} ^{a}\text{tweety} ^{r}.
\end{align*}
\]

Now let us consider the preferred program \((\Pi_4, <_4)\) we discussed earlier under the finite structure \(M = \{\{a\}, \{P(a)\}, \{Q(a)\}\}\). It is clear that \(M\) 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:

\[
\begin{align*}
\sigma_1^* & : P(a) \leftarrow Q(a), \text{Tag}(a), \\
\sigma_2^* & : P(a) \leftarrow Q(a), \text{Tag}(a), \\
\sigma_3^* & : Q(a) \leftarrow, \\
\sigma_4^* & : \text{Tag}(a) \leftarrow, \\
\sigma_5^* & : \text{Tag}(a) \leftarrow, \\
\sigma_6^* & : a = a \leftarrow, \\
\sigma_7^* & : a \leftarrow.
\end{align*}
\]

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:

\[
\begin{align*}
\Delta^0(\Pi)_S & = \{r \mid (1) \text{Pos}(r) = \emptyset \text{ and } \text{Neg}(r) \cap S = \emptyset; \\
& (2) \text{there does not exist a rule } r' \in \Pi \text{ such that } r' < r, \text{Pos}(r') \subseteq S \text{ and } \text{Neg}(r') \cap \text{Head}(r) = \emptyset\}; \\
\Delta^{i+1}(\Pi)_S & = \Delta^i(\Pi)_S \cup \{r \mid (1) \text{Pos}(r) \subseteq \text{Head}(\Delta^i(\Pi)_S) \\
& \text{and } \text{Neg}(r) \cap S = \emptyset; \\
& (2) \text{there does not exist a rule } r' \in \Pi \text{ such that } r' < r, r' \not\in \Delta^i(\Pi)_S \text{ and } \text{Pos}(r') \subseteq S \text{ and } \text{Neg}(r') \cap \text{Head}(\Delta^i(\Pi)_S) = \emptyset\}.
\end{align*}
\]
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 $(\Pi, \prec)$ based on $S$. $S$ is called a preferred answer set of $(\Pi, \prec)$ 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 $(\Pi_1, \prec_1)$ has a unique preferred answer set:

$$\{\text{Bird}(\text{cody}), \text{Bird}(\text{tweety}), \text{Penguin}(\text{tweety}), \text{Flies}(\text{cody}), \text{Cannot\_fly}(\text{tweety}), T \text{ag}_1(\text{cody}), T \text{ag}_2(\text{cody}), T \text{ag}_1(\text{tweety}), T \text{ag}_2(\text{tweety}), cody = \text{cody}, \text{tweety = tweety}\}.$$

On the other hand, the grounded preferred program of $(\Pi_4, \prec_4)$ has a unique preferred answer set $\{P(a), Q(a), T \text{ag}_1(a), T \text{ag}_2(a), a = a\}$.

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

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

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

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

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

The other direction can be proved with similar arguments. □

### 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 $(\Pi, \prec)$ 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 $\text{Pos}(r) \subseteq S$ and $\text{Neg}(r) \cap S = \emptyset$. Now consider $(\Pi, \prec)$ to be an FO preferred answer set program and $M$ a structure of $\tau(\Pi)$, then we say that a rule $r \in \Pi$ is a generating rule of $M$ under the assignment $\eta$.
if $M \models \text{Pos}(r) \eta \land \neg \text{Neg}(r) \eta$, where $\text{Pos}(r)$ and $\neg \text{Neg}(r)$ denote the formulas $\beta_1 \land \ldots \land \beta_k$ and

$$\neg \gamma_1 \land \ldots \land \neg \gamma_k,$$

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 $\text{Pos}(r) \subseteq \text{Head}(\Delta^\infty(\Pi)_S)$ and $\text{Neg}(r) \cap S = \emptyset$. Since $\text{Head}(\Delta^\infty(\Pi)_S) = S$, 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 $<$. Firstly, 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 $\text{Pos}(r) = \emptyset$, 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) $\text{Pos}(r) \notin \text{Head}(\Delta^{t-1}(\Pi)_S)$ or $\text{Neg}(r) \cap S \neq \emptyset$; or (2) there exists some $t' \in \Pi$ such that $r' < r, r' \notin \Delta^{t-1}(\Pi)_S, \text{Pos}(r') \subseteq S$ and $\text{Neg}(r') \cap \text{Head}(\Delta^{t-1}(\Pi)_S) = \emptyset$. Since $r$ is a generating rule of $S$ and $S = \text{Head}(\Delta^\infty(\Pi)_S)$, 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 $t' > (t - 1)^3$, no more rules from $\Pi$ can be added into $\Delta^t(\Pi)_S$, i.e. $\Delta^{t-1}(\Pi)_S = \Delta^t(\Pi)_S$ 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 $\text{Pos}(r') \subseteq S$ and $\text{Neg}(r') \cap \text{Head}(\Delta^{t-1}(\Pi)_S) = \emptyset$, 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 $\text{Pos}(r) \subseteq S$ and $\text{Neg}(r) \cap S = \emptyset$. Also, since $S$ is an answer set of $\Pi$, it follows $\text{Head}(r) \in S$. So $\text{Head}(\Delta^\infty(\Pi)_S) \subseteq S$. Now we can see $S \subseteq \text{Head}(\Delta^\infty(\Pi)_S)$, consider any $r \in \Pi$ such that $\text{Head}(r) \in S$. Since $S$ is an answer set of $\Pi$, there must exist a rule $r' \in \Pi$ such that $\text{Head}(r) = \text{Head}(r'), \text{Pos}(r') \subseteq S$ and $\text{Neg}(r') \cap S = \emptyset$. This means that $r'$ is a generating rule of $S$. From the condition, we know that $r' \in \Delta^\infty(\Pi)_S$. It follows $\text{Head}(r) = \text{Head}(r') \in \text{Head}(\Delta^\infty(\Pi)_S)$. That is $S \subseteq \text{Head}(\Delta^\infty(\Pi)_S)$. Therefore, we have $S = \text{Head}(\Delta^\infty(\Pi)_S)$. So $S$ is an answer set of $(\Pi, <)$. □

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 $M$ an answer set of $\Pi$. Then, the following two statements are equivalent:

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

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

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 $t > N$, $\Delta^t(\Pi)_S = \Delta^{t+1}(\Pi)_S$.}

11
Result 1: \((r, \eta) \in \Gamma^\infty(\Pi)_S\) if and only if \(r^* \in \Delta^\infty(\Pi)_S\), where 
\[
r^*: \text{Head}(r) \eta \leftarrow \text{Body}(r) \eta, \text{Tag}(\overline{x}) \eta,
\]
Result 2: \(r\) is a generating rule of \(\mathcal{M}\) under \(\eta\) if \(r^*\) is a generating rule of \(S\).

From the definitions of \(\Gamma^\infty(\Pi)_M, \Delta^\infty(\Pi)_S\) and \((\text{Ground}(\Pi)_M, \prec^*)\), together with Theorem 4.7, Results 1 and 2 are easily to be proved.

**Proposition 4.10.** For FO preferred answer set programs \((\Pi, \prec_1)\) and \((\Pi, \prec_2)\) where \(\prec_1 \subseteq \prec_2\), and a structure \(\mathcal{M}\) of \(\tau(\Pi)\), if \(\mathcal{M}\) is a preferred answer set of \((\Pi, \prec_2)\), then \(\mathcal{M}\) is also a preferred answer set of \((\Pi, \prec_1)\).

**Proof.** By Theorem 4.7, it is sufficient to prove the propositional version of the result. So we can assume that \((\Pi, \prec_1)\) and \((\Pi, \prec_2)\) are two propositional preferred programs and \(\prec_1 \subseteq \prec_2\). Suppose \(S\) is a preferred answer set of \((\Pi, \prec_2)\). Firstly, 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 \((\Pi, \prec_2)\), we have \(S = \text{Head}(\Delta^\infty(\Pi S^2))\). Now we show that \(S\) must also be a preferred answer set of \((\Pi, \prec_1)\).

To show this, we simply consider special preferred answer set program \((\Pi, \prec_1)\), where \(\eta = \emptyset\). 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 a preferred answer set of \((\Pi, \prec_1)\).

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 \(\Gamma^\infty(\Pi)_M\) 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 \(\Gamma^\infty(\Pi)_M\). The following lemma and theorem are towards this purpose.

**Lemma 4.11.** A grounded preferred answer set program \((\Pi, \prec)\) has a preferred answer set if there exists an answer set \(S\) of \(\Pi\) such that for each \((r_1, r_2) \in \prec\) 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 \((\Pi, \prec)\). 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 \(\text{Head}(r) \subseteq S\). 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 \(r \notin \Delta^t(\Pi)_S\), 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 \(\text{Head}(r) \subseteq S\).

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 \(\prec\). Firstly, suppose that there does not exist a rule \(r^* \in \Pi\) such that \(r^* \prec r\) 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 \(r^* \prec r\) and those \(r^*\) satisfying the condition of this lemma, \(r^* \in \Delta^\infty(\Pi)_S\). Now we consider \(r\). Since for any \(r'\) where \(r \not\prec r'\) 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 \(r\) must exist generating rules \(r_1, \ldots, r_k\) of \(S\) such that \(r_1, \ldots, r_k\) 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.

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

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

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 paper, 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 $P(x_1, \ldots, x_{i-1}, c, x_{i+1}, \ldots, x_n)$ occurs in some rule $r$ where $c$ is a constant, then we simply replace $P(x_1, \ldots, x_{i-1}, c, x_{i+1}, \ldots, x_n)$ by $P(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n)$ and then add $x_i = c$ into $r$'s positive body. In this way, we can simply write $x_P$ and $x_r$ 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 $x_P = xyz$. Now let $x_P = x_1 \ldots x_n$ and $y_P = y_1 \ldots y_n$, then we also write $x_P = y_P$ 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), $x_{r_1}$ and $x_{r_2}$ 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 $x_{r_1}$ and $x_{r_2}$ in a formula for which they have to be disjoint, we can always relabel the variables in $x_{r_2}$ 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
Now we propose an SO sentence that will simulate the progression semantics we de

5.3. Formalizing progression

(4) and lastly, the SO axiom:

i.e., those generating rules under the structure

Given an FO program

5.2. Well-orderings on generating rules in terms of structures

Then we have

Note that when

this is not the case when

what corresponds to a strict total-order, when only considering

Example 5.1. Consider the program \( \Pi_5 \) consisting of the following rules:

\[
\begin{align*}
r_1 : S(x_1, y_1) & \leftarrow E(z_1, y_1), \ x_1 = a, \ z_1 = a, \\
r_2 : S(x_2, z_2) & \leftarrow S(x_2, y_2), \ E(y_2, z_2'), \ z_2 = a, \ z_2' = b.
\end{align*}
\]

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

where we assume that \( \overline{x_P} \) is a tuple of distinguishable variables disjoint from \( \overline{y_P} \).

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:

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

i.e., 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 lastly, 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 non-empty 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 non-empty 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.
Definition 5.2. Given an FO answer set program II, we define an SO formula $\varphi^\text{PRO}_\Pi(\preceq, S)$ as follows (here “PRO” stands for progression):\footnote{In the following formulas (6), (7), and (8), in the case that $r_1 = r_2, r_1 = r_3, \text{ or } r_2 = r_3$, we assume a relabeling of $x_{r_1}, x_{r_2}, \text{ or } x_{r_3}$ (however appropriate) such that they will now be disjoint from each other.}

\begin{align*}
&\bigwedge_{r_1, r_2, r_3 \in \Pi} \forall x_{r_1} x_{r_2} x_{r_3} \left( \prec_{r_1 r_2} (x_{r_1}, x_{r_2}) \land \prec_{r_2 r_3} (x_{r_2}, x_{r_3}) \rightarrow \prec_{r_1 r_3} (x_{r_1}, x_{r_3}) \right) \tag{6} \\
&\land \bigwedge_{r_1, r_2 \in \Pi} \forall x_{r_1} x_{r_2} \left( \prec_{r_1 r_2} (x_{r_1}, x_{r_2}) \rightarrow \prec_{r_2 r_1} (x_{r_2}, x_{r_1}) \right) \tag{7} \\
&\land \bigwedge_{r_1, r_2 \in \Pi} \forall x_{r_1} x_{r_2} \left( \prec_{r_1 r_2} (x_{r_1}, x_{r_2}) \rightarrow \varphi^\text{GEN}_{r_1} (x_{r_1}) \land \varphi^\text{GEN}_{r_2} (x_{r_2}) \right) \tag{8} \\
&\land \bigwedge_{r \in \Pi} \forall x_r \left( \varphi^\text{GEN}_{r} (x_r) \rightarrow \varphi^\text{SUP}_r (\preceq, x_r) \right) \tag{9} \\
&\land \varphi^\text{WELL}_\Pi (\preceq, S), \tag{10}
\end{align*}

where:

- for each $r_1, r_2 \in \Pi$, the symbol $\prec_{r_1 r_2}$ is a predicate variable of arity $|x_{r_1}| + |x_{r_2}|$;\footnote{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.}
- $\preceq$ denotes the distinguishable tuple of predicate variables of the set $\{\prec_{r_1 r_2} | r_1, r_2 \in \Pi\}$;
- $\varphi^\text{SUP}_r (\preceq, x_r)$ denotes the following formula (here “SUP” stands for support):

\begin{equation}
\bigwedge_{P(\bar{x}^\prime) \in Pos(r), \ P \in P_{\Pi, r}(\Pi)} \bigvee_{r' \in \Pi, \ H(\bar{y}^\prime) = P(\bar{y}^\prime)} \exists \bar{x}^\prime_r (\prec_{r' r} (x_r^\prime, x_r) \land x_r^\prime = y_r^\prime) \tag{11}
\end{equation}

(where in the case that $r' = r$, we simply assume a relabeling of $x_r^\prime$ such that $x_r^\prime$ will be disjoint from $x_r^\prime$);
- $\varphi^\text{WELL}_\Pi (\preceq, S)$ denotes the following formula (here “WELL” stands for well-ordered):

\begin{equation}
\bigwedge_{r \in \Pi} \forall x_r (\tilde{S}_r (x_r) \rightarrow \varphi^\text{GEN}_{r} (x_r)) \land \bigvee_{r' \in \Pi} \exists x_r^\prime (\tilde{S}_{r'} (x_r^\prime) \land \forall y_r^\prime (\tilde{S}_{r'} (y_r^\prime) \land y_r^\prime \neq x_r^\prime \rightarrow \prec_{r' r} (x_r^\prime, y_r^\prime))) \tag{12}
\end{equation}

where $\tilde{S}$ denotes the distinguishable tuple of predicate variables of the set $\{\tilde{S}_r | r \in \Pi\}$ such that for each $r \in \Pi$, the arity of $\tilde{S}_r$ is $|x_r^\prime|$, and $y_r^\prime$ (in the consequent above) is a relabeling of the distinct variables of $x_r^\prime$ such that $y_r^\prime$ is now disjoint from $x_r^\prime$.

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

\[ \Gamma(\Pi)_\mathcal{M} = \{(r, \eta) \mid r \in \Pi, Pos(r) \eta \subseteq \mathcal{M} \text{ and } Neg(r) \eta \cap \mathcal{M} = \emptyset \text{ for an assignment } \eta \text{ of } \mathcal{M}\}, \]
which eventually establishes a correspondence to the sequence of progression sets \( \Gamma^0(\Pi)_M \), \( \Gamma^1(\Pi)_M \), \ldots, 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., \( \varphi^\text{SUP}_r(\preceq, \overline{x}_r) \)). Finally formula (10) enforces a well-order (i.e., \( \varphi^\text{WELL}_r(\preceq, \overline{S}_r) \)).

Specifically, (10) is fulfilled by formula (12), which encodes that each non-empty subset of generating rules has a least element. Indeed, formula (8) expresses the condition that the well-order only involves those of generating rules, while formula (9) expresses the condition that the well-order only involves those of generating rules, and \( \vee_{r \in \Pi} \exists \overline{x}_r \varphi^\text{GEN}_r(\overline{x}_r) \) encodes that the extents of each \( \overline{S}_r \) for \( r \in \Pi \) are only those of generating rules, and \( \vee_{r \in \Pi} \exists \overline{x}_r \varphi^\text{GEN}_r(\overline{x}_r) \) encodes that at least one of these subsets is non-empty, while the consequence

\[
\bigvee_{r' \in \Pi} \exists \overline{x}_{r'}(\overline{S}_{r'}(\overline{x}_{r'})) \land \forall y_{r'}(\overline{S}_{r'}(y_{r'})) \land y_{r'} \neq \overline{x}_{r'} \rightarrow \preceq_{r'}(\overline{x}_{r'}, y_{r'})
\]

\[
\land \bigwedge_{r'' \in \Pi, r'' \neq r'} \forall \overline{x}_{r''}(\overline{S}_{r''}(\overline{x}_{r''})) \rightarrow \preceq_{r''}(\overline{x}_{r''}, \overline{x}_{r'})
\]

encodes the existence of the least element.

**Proposition 5.3.** The SO formula \( \varphi^\text{PRO}_\Pi(\preceq, \overline{S}) \) 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 \( E \):

\[
r_1 : T(x_1, y_1) \leftarrow E(x_1, y_1),
\]

\[
r_2 : T(x_2, z_2) \leftarrow T(x_2, y_2), E(y_2, z_2).
\]

Then:

\[
\varphi^\text{COMP}_\Pi_6 = \forall xy(T(x, y) \leftarrow \exists x_1 y_1 (\varphi^\text{GEN}_{r_1}(x_1, y_1) \land x = x_1 \land y = y_1)
\]

\[
\land \exists x_2 y_2 z_2 (\varphi^\text{GEN}_{r_2}(x_2, y_2, z_2) \land x = x_2 \land y = z_2),
\]

where:

\[
\varphi^\text{GEN}_{r_1}(x_1, y_1) = E(x_1, y_1),
\]

\[
\varphi^\text{GEN}_{r_2}(x_2, y_2, z_2) = T(x_2, y_2) \land E(y_2, z_2).
\]

In particular, we also have that:

\[
\varphi^\text{SUP}_{r_1}(\preceq, x_1, y_1) = T,
\]

\[
\varphi^\text{SUP}_{r_2}(\preceq, x_2, y_2, z_2) = \exists x_3 y_3 z_3(\preceq_{r_2}(x_3, y_3, z_3) \land x_2 = x_3 \land y_2 = z_3)
\]

\[
\land \exists x_1 y_1(\preceq_{r_1}(x_1, y_1, x_2, y_2, z_2) \land x_2 = x_1 \land y_2 = y_1),
\]

where the tuple \( x_3 y_3 z_3 \) is a (disjoint) relabeling of \( x_2 y_2 z_2 \) and \( \preceq \) is the tuple \( \preceq_{r_1} \preceq_{r_1} \preceq_{r_2} \) of predicate variables.
About the formula $\varphi^\text{WELLO} (\prec, \overline{S})$, we further have:

\[
\forall x_1 y_1 (\overline{S}_{r_1} (x_1, y_1) \rightarrow \varphi^\text{GEN}_{f_1} (x_1, y_1)) \land \forall x_2 y_2 z_3 (\overline{S}_{r_2} (x_2, y_2, z_2) \rightarrow \varphi^\text{GEN}_{f_1} (x_2, y_2, z_2)) \\
\land (\exists x_1 y_1 \overline{S}_{r_1} (x_1, y_1) \lor \exists x_2 y_2 z_3 \overline{S}_{r_2} (x_2, y_2, z_2)) \\
(\exists x_1 y_1 (\overline{S}_{r_1} (x_1, y_1) \land \forall x y z (\overline{S}_{r_1} (x, y) \land (x_1 \neq x \lor y_1 \neq y) \rightarrow \overline{<}_{r_1} (x_1, x, y))) \\
\land \forall x_2 y_2 z_3 (\overline{S}_{r_2} (x_2, y_2, z_2) \rightarrow \overline{<}_{r_1} (x_1, x, y))) \\
\lor \exists x_2 y_2 z_3 (\overline{S}_{r_2} (x_2, y_2, z_2) \land \forall x y z (\overline{S}_{r_2} (x, y, z) \land (x_2 \neq x \lor y_2 \neq y \lor z_2 \neq z) \rightarrow \overline{<}_{r_2} (x_2, y_2, z_2, x, y, z))) \\
\land \forall x_1 y_1 (\overline{S}_{r_1} (x_1, y_1) \rightarrow \overline{<}_{r_2} (x_2, y_2, z_2, x, y, z)))
\]

where $\overline{S}$ is the tuple $\overline{S}_{r_1}, \overline{S}_{r_2}$ of predicate variables. Then $\forall \overline{S} \varphi^\text{WELLO} (\prec, \overline{S})$ expresses that each non-empty subset of $\Gamma (\Pi)_{\mathcal{M}}$ possesses a least element as induced by the relations of the predicates $\overline{<}_{r_1}, \overline{<}_{r_2}, \overline{<}_{r_1}, \overline{<}_{r_2}, \overline{<}_{r_1}$ and $\overline{<}_{r_2}$. Note that the non-empty subsets of $\Gamma (\Pi)_{\mathcal{M}}$ are implicitly encoded via the universally quantified predicate variables $\overline{S}_{r_1}$ and $\overline{S}_{r_2}$.

**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 $\mathcal{M} \models \exists \overline{S} \varphi^\text{WELLO} (\prec, \overline{S})$, where $\varphi^\text{II} (\prec, \overline{S}) = \varphi^\text{PRO} (\prec, \overline{S}) \land \varphi^\text{COMP} (\prec, \overline{S})$.

**Proof.** See appendix. \(\square\)

### 5.4. Incorporating preference

We have shown that the SO sentence $\exists \overline{S} \varphi^\text{II} (\prec, \overline{S})$ proposed in Theorem 5.5 precisely captures the answer set semantics for FO answer set programs. In this subsection, we further extend this formula by embedding preferences. We first define the following formula.

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

\[
\bigwedge_{r \in \Pi} \forall \overline{x}_r (\varphi^\text{GEN}_{f_r} (\overline{x}_r) \rightarrow \bigwedge_{r' < r} \forall \overline{x}_r' ( (\varphi^\text{GEN}_{f_{r'}} (\overline{x}_r') \rightarrow \overline{<}_{r'} (\overline{x}_r', \overline{x}_r)) \\
\land (\neg \varphi^\text{GEN}_{f_{r'}} (\overline{x}_r') \rightarrow ( \varphi^\text{POS}_{r'} (\overline{x}_r') \lor \varphi^\text{DEF}_{r'} (\prec, \overline{x}_r', \overline{x}_r))))
\]

where:

- $\varphi^\text{POS}_{r'} (\overline{x}_r')$ denotes the following formula (“POS” stands for positive body not satisfied):

\[
\bigvee_{P(\overline{x}_p) \in \text{Pos}(r')} \neg P(\overline{x}_{\overline{p}}); \tag{14}
\]

- $\varphi^\text{DEF}_{r'} (\prec, \overline{x}_r', \overline{x}_r)$ (“DEF” stands for defeated) denotes the formula

\[
(\bigvee_{P(\overline{x}_p) \in \text{Neg}(r')} P(\overline{x}_{\overline{p}})) \lor (\bigvee_{P(\overline{x}_p) \in \text{Neg}(r')} \bigvee_{P(\overline{x}_p) \in \text{Neg}(r')} \exists \overline{x}_p' (\prec_{r''} (\overline{x}_p', \overline{x}_r') \land \overline{x}_{\overline{p}} = \overline{y}_{\overline{p}})). \tag{15}
\]

Formula (13) 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, i.e., by (14), or it is defeated by some other rule $r''$.
such that \( r'' \) is already derived earlier than \( r \) in the progression stages, i.e., by (15), 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 \( M \) a \( \tau(\Pi) \)-structure. \( M \) is a preferred answer set of \( (\Pi, <) \) iff \( M \models \exists \overline{x} \forall \overline{y} S \varphi_{(\Pi, <)}(\overline{x}, \overline{y}) \), where \( \varphi_{(\Pi, <)}(\overline{x}, \overline{y}) = \varphi_{\text{PRE}}^{\text{PRO}}(\overline{x}, \overline{y}) \land \varphi_{\Pi}^{\text{COMP}} \land \varphi_{\Pi}^{\text{TOTAL}}(\overline{x}) \land \varphi_{\Pi}^{\text{WELL}}(\overline{x}, \overline{y}) \).

**Proof.** See appendix.

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

**Proof.** We obtain an \( \exists \text{SO} \) formula \( \exists \overline{z} \psi_{(\Pi, <)}(\overline{z}) \) from \( \exists \overline{x} \forall \overline{y} S \varphi_{(\Pi, <)}(\overline{x}, \overline{y}) \) by substituting the SO formula \( \varphi_{\Pi}^{\text{TOTAL}}(\overline{x}) \) for \( \varphi_{\Pi}^{\text{WELL}}(\overline{x}, \overline{y}) \) in \( \varphi_{(\Pi, <)}(\overline{x}, \overline{y}) \) where

\[
\varphi_{\Pi}^{\text{TOTAL}}(\overline{x}) = \bigwedge_{r \in \Pi} \forall \overline{x}_r \overline{y}_r (\varphi_{r}^{\text{GEN}}(\overline{x}_r) \land \varphi_{r}^{\text{GEN}}(\overline{y}_r) \land \overline{x}_r \neq \overline{y}_r \rightarrow (\overline{<}_{rr}(\overline{x}_r, \overline{y}_r) \lor \overline{<}_{rr}(\overline{y}_r, \overline{x}_r)))
\]

which, in conjunction with formulas (6), (7), and (8), expresses a strict total-order on \( \Gamma(\Pi)_M \). Then, since well-orders are strict total-orders on finite structures, the result follows.

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 \( A \) a finite structure on \( \tau(\Pi) \). Then there exists a first-order sentence \( \Phi(\Pi, <) \) where \( \tau(\Pi) \subseteq \tau(\Phi(\Pi, <)) \), such that \( A \) is a preferred answer set of \( (\Pi, <) \) iff there is a model \( M \) of \( \Phi(\Pi, <) \) and \( A \) is a restriction of \( M \) on \( \tau(\Pi) \).

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

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:

\[
\begin{align*}
  r_1 & : P(x_1) \iff Q(x_1), \\
  r_2 & : Q(x_2) \iff r_1 \leq _2 r_2.
\end{align*}
\]

\(^7\)Where "TOTALOR" stands for total-order.

\(^8\)Note that for our purpose here, we relabel the variables in rules \( r_1 \) and \( r_2 \) to make them disjoint.
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 \( \varphi_{\Pi_2}^{\text{Pref}} (\prec) \) to be:
\[
\forall x_1 (\varphi_{r_1}^{\text{GEN}} (x_1) \rightarrow \top) \\
\land \forall x_2 (\varphi_{r_2}^{\text{GEN}} (x_2) \rightarrow \forall x_1 ((\varphi_{r_1}^{\text{GEN}} (x_1) \rightarrow \prec_{r_1} (x_1, x_2)) \\
\land (\neg \varphi_{r_1}^{\text{GEN}} (x_1) \rightarrow (\varphi_{r_1}^{\text{POS}} (x_1) \lor \varphi_{r_2}^{\text{Def}} (\prec, x_2, x_1))))).
\]
(16)

Then, from \( \varphi_{\Pi_2}^{\text{Proj}} (\prec) \), we further have:
\[
\forall x_1 (\varphi_{r_1}^{\text{GEN}} (x_1) \rightarrow \exists x_2 (\prec_{r_1} (x_2, x_1) \land \forall x_1 = x_2)),
\]
i.e., the support for \( r_1 \). Then, since we also have
\[
\forall x_1 x_2 (\prec_{r_1 r_2} (x_1, x_2) \rightarrow \varphi_{r_1}^{\text{GEN}} (x_1) \land \varphi_{r_2}^{\text{GEN}} (x_2))
\]
by \( \varphi_{\Pi_2}^{\text{Proj}} (\prec) \) (i.e., via (8)), then (17) is equivalent to:
\[
\forall x_1 (\varphi_{r_1}^{\text{GEN}} (x_1) \rightarrow \exists x_2 (\prec_{r_1} (x_2, x_1) \land \varphi_{r_2}^{\text{GEN}} (x_2) \land \forall x_1 = x_2)).
\]
(18)

Then, from (16), we further get that (19) is equivalent to:
\[
\forall x_1 (\varphi_{r_1}^{\text{GEN}} (x_1) \rightarrow \exists x_2 (\prec_{r_1} (x_2, x_1) \land \varphi_{r_2}^{\text{GEN}} (x_2) \land \prec_{r_1 r_2} (x_1, x_2) \land \forall x_1 = x_2)),
\]
i.e., simply adding the atom, \( \prec_{r_1 r_2} (x_1, x_2) \), to the consequent. Then by (6), i.e., the transitive property, we further have that (20) is equivalent to
\[
\forall x_1 (\varphi_{r_1}^{\text{GEN}} (x_1) \rightarrow \exists x_2 (\prec_{r_2} (x_2, x_1) \land \varphi_{r_2}^{\text{GEN}} (x_2) \land \prec_{r_1 r_2} (x_1, x_2) \land \forall x_1 = x_2\).
\]
(21)
i.e., adding the atom, \( \prec_{r_2 r_2} (x_2, x_2) \), to the consequent. Then, by the asymmetry as expressed by formula (7) (i.e., which implies we cannot have \( \prec_{r_2 r_2} (x_2, x_2) \)), we further have that
\[
\forall x_1 (\varphi_{r_1}^{\text{GEN}} (x_1) \rightarrow \bot) \equiv \forall x_1 \neg \varphi_{r_1}^{\text{GEN}} (x_1).
\]
(22)

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 \( \exists \exists \psi (\Pi_2, <) \prec \) is unsatisfiable, which corresponds to the preferred program \( (\Pi_2, <) \) 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 \( r (\Pi) \)-structure \( M \), \( M \models \exists \exists \forall \bar{S} \varphi_{(\Pi, <)} (\prec, \bar{S}) \) iff \( M \) is an answer set of \( \Pi \).

**Proof.** When \( <= \emptyset \), \( \exists \exists \forall \bar{S} \varphi_{(\Pi, <)} (\prec, \bar{S}) \) reduces to \( \exists \exists \forall \bar{S} \varphi_{(\Pi, <)} (\prec, \bar{S}) \), since \( \varphi_{(\Pi, <)} (\prec) \) as given by:
\[
\bigwedge_{r \in \Pi} \forall \bar{x}_r (\varphi_r^{\text{GEN}} (\bar{x}_r)) \rightarrow \bigwedge_{r < r'} \forall \bar{x}_r ((\varphi_r^{\text{GEN}} (\bar{x}_r) \rightarrow \prec_{r'} (\bar{x}_r, \bar{x}_r))
\land (\neg \varphi_r^{\text{GEN}} (\bar{x}_r) \rightarrow (\varphi_r^{\text{POS}} (\bar{x}_r) \lor \varphi_r^{\text{Def}} (\prec, \bar{x}_r, \bar{x}_r))))),
\]
becomes
\[
\bigwedge_{r \in \Pi} \forall \bar{x}_r (\varphi_r^{\text{GEN}} (\bar{x}_r) \rightarrow \top) \equiv \top.
\]
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 paper 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 \( T_{\Pi,Y}X \) as follows:

\[
T_{\Pi,Y}X = \{ \text{Head}(r) \mid r \in \Pi, Pos(r) \subseteq X \text{ and } Neg(r) \cap Y = \emptyset \}.
\] (23)

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

\[
T_{(\Pi,<),Y}X = \{ \text{Head}(r) \mid (1) \text{ Pos}(r) \subseteq X \text{ and } Neg(r) \cap Y = \emptyset; \]

(2) there does not exist a rule \( r' \in \Pi \) where \( r' < r \),

\[
\text{Pos}(r') \subseteq Y \text{ and } Neg(r') \cap X = \emptyset \},
\] (24)

where (as mentioned in [Schaub and Wang 2003]), \( \text{Rule}(.) \) 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 (24) by requiring that more preferred rules, i.e., \( r' \) in (24), have effectively been applied.

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

\[
T_{(\Pi,<),Y}^{i+1}X = T_{(\Pi,<),Y}(T_{(\Pi,<),Y}^iX).
\]

Also note that in the case of preference free programs, a set \( X \) of atoms is an answer set of \( \Pi \) iff \( T_{\Pi,\emptyset}^\infty X = X \), i.e., 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 \( T_{(\Pi,<),\emptyset}^\infty X = X \).

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, i.e., 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 \(T_{(\Pi, <), X}^\infty \emptyset = X\) iff \(\text{Head}^{\infty}(\Pi|_X) = X\).

**Proof.** The result will be easily proved by induction that \(T_{(\Pi, <), X}^i \emptyset = \text{Head}^i(\Pi|_X)\) for all \(i \geq 0\).

### 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 \(T_{(\Pi, <), Y} X\), 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 \(W T_{(\Pi, <), Y} X\) (i.e., with a \(W\) left-superscript for “weak”) denote the consequence operator defined as follows:

\[
W T_{(\Pi, <), Y} X = \{\text{Head}(r) | (1) \text{Pos}(r) \subseteq X \text{ and } \neg \text{Pos}(r) \cap Y = \emptyset; \\
(2) \text{there does not exist a rule } r' \in \Pi \text{ where } r' < r, \\
\text{Head}(r') \notin X, \text{ and } \text{Pos}(r') \subseteq Y \text{ and } \\
\neg \text{Pos}(r') \cap X = \emptyset\}. \tag{25}
\]

Then, compared to the \(T_{(\Pi, <), Y} X\) operator, the application of rules in (25) 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 \(W T_{(\Pi, <), Y} X = X\) (with the iterative applications of \(W T_{(\Pi, <), Y} X\) defined as above).

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 \(M\) a corresponding \(\tau(\Pi)\)-structure. We define

\[
\Gamma_W^0(\Pi, M) = \{(r, \eta) | (1) \text{Pos}(r) \eta \subseteq M^0(\Pi) \text{ and } \neg \text{Pos}(r) \eta \cap M = \emptyset; \\
(2) \text{there does not exist a rule } r' \in \Pi \text{ and an assignment } \eta' \text{ such that } \\
r' < r, \text{Pos}(r') \eta' \subseteq M \text{ and } \neg \text{Pos}(r') \eta' \cap M^0(\Pi) = \emptyset\};
\]

and for \(t \geq 0\),

\[
\Gamma_W^{t+1}(\Pi, M) = \Gamma_W^t(\Pi, M) \cup \{(r, \eta) | (1) \text{Pos}(r) \eta \subseteq \lambda_M(\Gamma_W^t(\Pi, M)) \text{ and } \neg \text{Pos}(r) \eta \cap M = \emptyset; \\
(2) \text{there does not exist a rule } r' \in \Pi \text{ and an assignment } \eta' \text{ such that } \\
r' < r, \text{Head}(r') \eta' \notin \lambda_M(\Gamma_W^t(\Pi, M)), \text{ and } \\
\text{Pos}(r') \eta' \subseteq M \text{ and } \neg \text{Pos}(r') \eta' \cap \lambda_M(\Gamma_W^t(\Pi, M)) = \emptyset\}. \tag{26}
\]

22
Let $\Gamma_{W}^{\infty}(\Pi, M) = \bigcup_{t=0}^{\infty} \Gamma_{W}^{t}(\Pi, M)$.

**Definition 6.3.** Let $\langle \Pi, < \rangle$ be a preferred FO program and $M$ a $\tau(\Pi)$-structure. $M$ is called a weak preferred answer set of $\langle \Pi, < \rangle$ iff $\lambda_{M^{P}}(\Gamma_{W}^{\infty}(\Pi, M)) = M$.

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

**Proposition 6.4.** Let $\langle \Pi, < \rangle$ be a preferred program. If a $\tau(\Pi)$-structure $M$ is a weak preferred answer set of $\langle \Pi, < \rangle$, then $M$ is an answer set of $\Pi$.

**Proof.** See appendix. $\square$

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:

$$\exists x \forall S \varphi_{\langle \Pi, < \rangle}(\langle x, S \rangle) = \varphi_{\Pi}^{\text{PREF}}(\langle x \rangle) \land \varphi_{\Pi}^{\text{PRO}}(\langle x, S \rangle) \land \varphi_{\Pi}^{\text{COMP}}$$

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

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

$$\varphi_{\Pi}^{\text{SUP}}(\langle x, r \rangle) = \bigvee_{\left. x \in \Pi \right. \text{Head}(r) = \text{P}(\langle y \rangle)} \exists x \forall S \varphi_{\Pi}^{\text{SUP}}(\langle x, r \rangle) \land \varphi_{\Pi}^{\text{GEN}}(\langle x, r \rangle)$$

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

Generally speaking, $\varphi_{\Pi}^{\text{SUP}}(\langle x, r \rangle)$ as defined in (27) 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 $\varphi_{\Pi}^{\text{PRO}}(\langle x, S \rangle)$ (here “W-PRO” for weak progression), we now denote the formula obtained from $\varphi_{\Pi}^{\text{SUP}}(\langle x, r \rangle)$ with $\varphi_{\Pi}^{\text{SUP}}(\langle x, r \rangle)$ with $\varphi_{\Pi}^{\text{SUP}}(\langle x, r \rangle)$.

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

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

$$\bigwedge_{r \in \Pi} \forall x^{r}_{r} (\varphi_{r}^{\text{GEN}}(\langle x^{r}_{r} \rangle) \rightarrow \left( \bigwedge_{r < r'} \forall x^{r}_{r'} (\varphi_{r'}^{\text{GEN}}(\langle x^{r}_{r'} \rangle) \rightarrow \langle x^{r'}_{r} \rangle(\langle x^{r}_{r} \rangle, \langle x^{r}_{r'} \rangle) \right)$$

$$\land \left( \neg \varphi_{r}^{\text{GEN}}(\langle x^{r}_{r} \rangle) \rightarrow (\varphi_{r}^{\text{POS}}(\langle x^{r}_{r} \rangle) \lor \varphi_{r}^{\text{DEF}}(\langle x^{r}_{r} \rangle, \langle x^{r}_{r} \rangle) \lor \varphi_{r}^{\text{HEAD}}(\langle x^{r}_{r} \rangle, \langle x^{r}_{r} \rangle, \langle x^{r}_{r} \rangle)) \right))$$

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

$$\bigvee_{r'' \in \Pi} \exists x^{r''}_{r''}(\langle x^{r''}_{r''} \rangle, \langle x^{r''}_{r''} \rangle) \land \varphi_{r''}^{\text{DEF}}(\langle x^{r''}_{r''} \rangle, \langle x^{r''}_{r''} \rangle, \langle x^{r''}_{r''} \rangle)$$

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

23
Intuitively, $\varphi_{(\Pi, <)}^{W\text{-PREF}}(\vec{z})$ differs from $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$ in the sense that it is enough for a more preferred non-generating rule to have its head already derived by some generating rule from the previous stages, i.e., as enforced by $\varphi_{r, r'}^{\text{HEAD}}(\vec{z}, x_r, x_{r'})$ where $r'$ is a non-generating rule. This is in contrast with $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$ since it requires all non-generating rule to be either explicitly non-derivable as its positive body cannot be derived (i.e., via the $\varphi_r^{-\text{POS}}(x_{r'})$), or is already defeated by some rule derived from the earlier stages (i.e., via the $\varphi_{r, r'}^{\text{DEF}}(\vec{z}, x_r, x_{r'})$) formula. Thus, it is not difficult to see that the only real difference between $\varphi_{(\Pi, <)}^{W\text{-PREF}}(\vec{z})$ and $\varphi_{(\Pi, <)}^{\text{PREF}}(\vec{z})$ is the addition of the other choice for the non-generating rules (i.e. via $\varphi_{r, r'}^{\text{HEAD}}(\vec{z}, x_r, x_{r'})$).

As in the original definition of the FO preferred answer sets, we also have 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

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

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

**Proof.** See appendix. □

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

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

**Proof.** Similarly to the proof of Proposition 5.8, we obtain an $\exists\text{SO}$ formula from

$$
\exists \vec{z} \forall \vec{S} \left( \varphi_{(\Pi, <)}^{W\text{-PREF}}(\vec{z}) \land \varphi_{\Pi}^{W\text{-PRO}}(\vec{z}, \vec{S}) \land \varphi_{\Pi}^{\text{COMP}} \right)
$$

by substituting the SO formula $\varphi_{\Pi}^{\text{TOTALOR}}(\vec{z})$ for $\varphi_{\Pi}^{\text{WELLOR}}(\vec{z}, \vec{S})$ in $\varphi_{\Pi}^{W\text{-PRO}}(\vec{z}, \vec{S})$. Then since total-orders are well-orders on finite sets, the result follows. □

### 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 $(r_i)_{i \in I}$ of $\Gamma(\Pi)_M$ satisfying the following properties. For every $i, j \in I$: \(^9\)

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

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

\(^9\)Note that the original definition as it was discussed by Delgrande et al. [2003] only concerned finite propositional programs.
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 \(W = (\Gamma(\Pi)_{\mathcal{M}}, <^W)\) 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 \((r_1, \eta_1) <^W (r_2, \eta_2)\);
2. \((r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}, (r', \eta') \notin \Gamma(\Pi)_{\mathcal{M}},\) and \(r' < r\) implies that either:
   (a) \(\text{Pos}(r')\eta' \not\subseteq \mathcal{M}\) or
   (b) \(\text{Neg}(r')\eta' \cap \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^W (r, \eta)\} \neq \emptyset\) or
3. \(\text{Head}(r')\eta' \in \mathcal{M}\).

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 \(\varphi_{\text{BE-PREF}}^{\Pi, <} (\preceq, \overline{S})\) as follows (here “BE-PREF” stands for Brewka-Eiter preference):

\[
\bigwedge_{r \in \Pi} \forall \overline{x_r} (\varphi_{r}^{\text{GEN}}(\overline{x_r}) \rightarrow \bigwedge_{r' < r} \forall \overline{x_{r'}} (\varphi_{r'}^{\text{GEN}}(\overline{x_{r'}}) \rightarrow \preceq_{r'r}(\overline{x_{r'}}, \overline{x_r}))
\]

\[
\land \bigwedge_{r_1, r_2, r_3 \in \Pi} \forall \overline{x_{r_1}} \overline{x_{r_2}} \overline{x_{r_3}} ((\varphi_{r_1}^{\text{GEN}}(\overline{x_{r_1}})) \land \preceq_{r_1r_2}(\overline{x_{r_1}}, \overline{x_{r_2}}) \rightarrow \preceq_{r_1r_3}(\overline{x_{r_1}}, \overline{x_{r_3}}))
\]

\[
\land \bigwedge_{r_1, r_2 \in \Pi} \forall \overline{x_{r_1}} \overline{x_{r_2}} (\preceq_{r_1r_2}(\overline{x_{r_1}}, \overline{x_{r_2}}) \rightarrow \neg \preceq_{r_1r_2}(\overline{x_{r_1}}, \overline{x_{r_2}}))
\]

\[
\land \bigwedge_{r_1, r_2, r_3 \in \Pi} \forall \overline{x_{r_1}} \overline{x_{r_2}} \overline{x_{r_3}} (\varphi_{r_1}^{\text{GEN}}(\overline{x_{r_1}}) \land \varphi_{r_2}^{\text{GEN}}(\overline{x_{r_2}}))
\]

\[
\land \varphi_{\Pi}^{\text{WELLOR}} (\preceq, \overline{S}),
\]

such that \(\varphi_{r}^{\text{POS}}(\overline{x_r})\) and \(\varphi_{r}^{\text{DEF}}(\preceq, \overline{x_r}, \overline{x_r})\) are defined as in Definition 5.6, \(\varphi_{\Pi}^{\text{WELLOR}} (\preceq, \overline{S})\) is defined as in Definition 5.2, and \(\varphi_{r}^{\text{BE-HEAD}}(\overline{x_r})\) stands for the formula

\[
P(\overline{x_r}),
\]

where we assume that \(\text{Head}(r')\) is of the form \(P(\overline{x_r})\).

Let us take a closer look at Definition 6.9. In a nutshell, about the Formula (30), for each generating rule \((r, \eta) \in \Gamma(\Pi)_{\mathcal{M}}\), we have that each other rule \((r', \eta')\) (can be non-generating) 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, i.e., as enforced by \(\preceq_{r'r}(\overline{x_{r'}}, \overline{x_r})\). On the other hand, in the case that \((r', \eta')\) is a non-generating rule, we must have that either: \((r', \eta')\) cannot possibly be derived (since \(\text{Pos}(r'')\eta' \not\subseteq \mathcal{M}\) as encoded by \(\varphi_{r''}^{\text{POS}}(\overline{x_{r''}}))\), as it is already defeated in the well-order (via \(\varphi_{r''}^{\text{DEF}}(\preceq, \overline{x_{r'}}, \overline{x_r})\)), or that \(\text{Head}(r')\eta' \in \mathcal{M}\) (as enforced by the formula \(\varphi_{r'}^{\text{BE-HEAD}}(\overline{x_{r'}}) = P(\overline{x_r})\) where we assume that \(\text{Head}(r')\) is \(P(\overline{x_r})\)). Finally, Formulas (31), (32), (33), and (34) enforces the well-order (where \(\varphi_{\Pi}^{\text{WELLOR}} (\preceq, \overline{S})\) 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

\[
\mathcal{M} \models \exists \overline{z_1} \overline{z_2} \forall S_1 S_2 (\varphi_{\text{BE-PREF}}^{\Pi, <} (\preceq_1, \overline{S_1}) \land \varphi_{\Pi}^{\text{PRO}} (\preceq_2, \overline{S_2}) \land \varphi_{\Pi}^{\text{COMP}})
\]

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 \(\varphi_{\text{BE-PREF}}^{\Pi, <} (\preceq_1, \overline{S_1})\) are independent from that of \(\varphi_{\Pi}^{\text{PRO}} (\preceq_2, \overline{S_2})\). This is
in contrast with both the SW and the “weak” SW since they do not dispose of the groundedness requirement.

**Proof.**

\[ M \models \exists <_{1} \forall S_{1} S_{2} (\varphi_{\Pi, <}^{\text{BE-PREF}} (\vec{S}_{1}, \vec{S}_{1}) \land \varphi_{\Pi}^{\text{PRO}} (\vec{S}_{2}, \vec{S}_{2}) \land \varphi_{\Pi}^{\text{COMP}}) \]

iff \[ M \models \exists <_{2} \forall S_{2} (\varphi_{\Pi}^{\text{PRO}} (\vec{S}_{2}, \vec{S}_{2}) \land \varphi_{\Pi}^{\text{COMP}}) \] and \[ M \models \exists <_{1} \forall S_{1} \varphi_{\Pi}^{\text{BE-PREF}} (\vec{S}_{1}, \vec{S}_{1}) \] iff \( M \) is an answer set of \( \Pi \) (by Theorem 5.5) and there exists a BE preference preserving well-order of \( \Gamma(\Pi) \) (by Definitions 6.8 and 6.9) iff \( M \) is a BE preferred answer set of \((\Pi, <)\). □

**Proposition 6.11.** On finite structures, the BE preferred answer set framework can also be captured by an \( \exists \text{SO} \) sentence.

**Proof.** As in the proofs of Propositions 5.8 and 6.7, simply replace the occurrences of \( \varphi_{\Pi}^{\text{WELL-OR}} (\vec{S}, \vec{S}) \) by \( \varphi_{\Pi}^{\text{TOTAL-OR}} (\vec{S}) \). Then since total-orders are well-orders on finite sets, the result follows. □

### 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. Firstly, it is our current task to develop a preferred first-order ASP solver under the progression-based preference semantics. As discussed in subsection 6.5, our approach will be based on the logical characterization result proved in Section 5.

Secondly, as an application of preferred ASP, there have been some works on updates through preferred logic programs, e.g., [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”, i.e., 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.

**Acknowledgement**

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

**REFERENCES**


Appendix: Proofs

Proof of Proposition 3.5

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 \( \Omega_\Pi = \{Q_1, \ldots, Q_n\} \) the set of all the intentional predicates of \( \Pi \). Consider a structure \( M \) of \( \tau(\Pi) \). The \( t \)-th simultaneous evolution stage of \( \Pi \) based on \( M \) \((t \geq 0)\), denoted as \( M^t(\Pi) \), is a structure of \( \tau(\Pi) \) defined inductively as follows:

\[
M^0(\Pi) = (\text{Dom}(M), c^M_1, \ldots, c^M_n, P^M_1, \ldots, P^M_s, Q^M_1, \ldots, Q^M_n),
\]

where \( c^M_i = c^\Pi_i \) for each constant \( c_i \) of \( \tau \) \((1 \leq i \leq r)\),

\[
P^M_j = P^\Pi_j \text{ for each extensional predicate } P_j \text{ in } \tau_{ext}(\Pi),
\]

\((1 \leq j \leq s), \text{ and } Q^M_k = \emptyset \text{ for each intentional predicate } Q_k \text{ in } \tau_{int}(\Pi) \((1 \leq k \leq n)\);

\[
M^{t+1}(\Pi) = M^t(\Pi) \cup \{Q_i(\overline{x}) \eta \mid \text{ there exists a rule } Q_i(\overline{x}) \leftarrow \beta_1, \ldots, \beta_m, \text{ not } \gamma_1, \ldots, \text{ not } \gamma_l \in \Pi \text{ and an assignment } \eta \text{ such that for all } j \geq (1 \leq j \leq m), \beta_j \eta \in M^t(\Pi), \text{ and for all } k \geq (1 \leq k \leq l), \gamma_k \eta \notin M\};
\]

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

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

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

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

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

(a) \( Pos(r') \eta' \subseteq M \) and \( Neg(r') \eta' \cap \lambda_{M^0}(\Gamma^n(\Pi)_M) = \emptyset \);

(b) \((r', \eta') \notin \Gamma^n(\Pi)_M \),

i.e. 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

\[
B^0(r^*, \eta^*) = \{(r', \eta') \mid r' < r^*, Pos(r') \eta' \subseteq M \text{ and } Neg(r') \eta' \cap M^0(\Pi) = \emptyset \}\text{ and } B^k(r^*, \eta^*) = \{(r', \eta') \mid (a) r' < r^*; \}
\]

(b) \( Pos(r') \eta' \subseteq M \) and \( Neg(r') \eta' \cap \lambda_{M^0}(\Gamma^k(\Pi)_M) = \emptyset \);

(c) \((r', \eta') \notin \Gamma^k(\Pi)_M \)
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 $\text{Neg}(r'')\eta'' \cap \lambda_{\mathcal{M}^0}(\Gamma^{k+l}(\Pi)_{\mathcal{M}}) \neq \emptyset$) 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 $B^{k+l}(r^*, \eta^*) \supseteq B^k(r^*, \eta^*)$. 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 $r'' \neq r'$ (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 $\text{Pos}(r')\eta' \subseteq \mathcal{M}$ and $\text{Neg}(r')\eta' \cap \lambda_{\mathcal{M}^0}(\Gamma^k(\Pi)_{\mathcal{M}}) = \emptyset$.

Also, since $\lambda_{\mathcal{M}^0}(\Gamma^\infty(\Pi)_{\mathcal{M}}) = \mathcal{M}$ by assumption, then for some $l \geq 0$, we have $\text{Pos}(r')\eta' \subseteq \lambda_{\mathcal{M}^0}(\Gamma^{k+l}(\Pi)_{\mathcal{M}})$. Now there can only be two possibilities:

**Case 1.** $\text{Neg}(r')\eta' \subseteq \mathcal{M}$.

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 $(r', \eta') \in \mathcal{M}^{k+l}(\Gamma^{k+l}(\Pi)_{\mathcal{M}})$. In particular, the definition of $\Gamma^{k+l}(\Pi)_{\mathcal{M}}$ since $(r', \eta')$ will now be applicable at this stage (i.e., since there are no pairs $(r'', \eta'')$
Proof of Theorem 5.5

This completes our proof of $M \subseteq \lambda_{M^0}(\Gamma^\infty(\Pi)_{M})$. \square

Proof of Theorem 5.5
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 $\mathcal{M} \models \exists \Xi \forall \Sigma \varphi_{\Pi}(\Xi, \Sigma)$.

$(\Rightarrow)$ 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

$$(\mathcal{M}^*)_t(\Pi) = \{ (r, \eta) \mid Pos(r, \eta) \subseteq M^{t+1}(\Pi) \text{ and } Neg(r, \eta) \cap M = \emptyset \}.$$  \hfill (36)

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:

$$\Lambda^0(\Pi, \mathcal{M}) = (\mathcal{M}^*)_0(\Pi);$$

$$\Lambda^{t+1}(\Pi, \mathcal{M}) = (\mathcal{M}^*)^t(\Pi) \setminus (\mathcal{M}^*_t(\Pi)).$$

Then clearly, we also have $\bigcup_{t \geq 0} \Lambda^t(\Pi, \mathcal{M}) = (\mathcal{M}^*)^\infty(\Pi)$. Moreover, by the definition of $\Lambda^t(\Pi, \mathcal{M})$, it is not difficult to see that the sets $\Lambda^0(\Pi, \mathcal{M}), \Lambda^1(\Pi, \mathcal{M}), \ldots, \Lambda^\infty(\Pi, \mathcal{M})$ 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 $\Lambda^t(\Pi, \mathcal{M})$, by the well-ordering theorem (every set can be well-ordered), there exists a well-ordering on the elements of $\Lambda^t(\Pi, \mathcal{M})$. Set such a well-ordering as $\mathcal{W}_t = (\Lambda^t(\Pi, \mathcal{M}), <^{\mathcal{W}_t})$.

2. Then we define the well-order $\mathcal{W} = (\Gamma(\Pi, \mathcal{M}), <^{\mathcal{W}})$ on $\Gamma(\Pi, \mathcal{M})$ by setting

$$<^{\mathcal{W}} = \bigcup_{t \geq 0} <^{\mathcal{W}_t} \cup \{(r_1, \eta_1), (r_2, \eta_2) \mid (r_1, \eta_1) \in \Lambda^t_{\mathcal{M}}(\Pi), (r_2, \eta_2) \in \Lambda^{t+1}_{\mathcal{M}}(\Pi), t_1 < t_2\}.$$  

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

Now we denote by $\forall \overline{S} \varphi_{\Pi}(\overline{S})$ the sentence obtained from $\forall \overline{S} \varphi_{\Pi}(\overline{S})$ by treating the existentially quantified predicate variables in $\overline{S}$ as predicate constants. Then to show $\mathcal{M} \models \exists \overline{S} \forall \overline{S} \varphi_{\Pi}(\overline{S})$, we show that there exists an expansion $\mathcal{M}'$ of $\mathcal{M}$ such that $\mathcal{M}' \models \forall \overline{S} \varphi_{\Pi}(\overline{S})$.

In the following, we denote the treatment of each predicate variable, $<_{r_1, r_2}$, as a predicate constant by simply denoting it as $<_{r_1, r_2}$ (just removing the "\-" accent). Now set $\mathcal{M}'$ to be an expansion of $\mathcal{M}$ such that $\mathcal{M}'$ is of the extended signature $\tau(\Pi) \cup \{ <_{r_1, r_2} \mid r_1, r_2 \in \Pi \}$ of $\mathcal{M}$ and where each predicate symbol $<_{r_1, r_2}$ (for $r_1, r_2 \in \Pi$) is interpreted by:

$$<^{\mathcal{M}'}_{r_1, r_2} = \{ (\eta_1(u_1), \ldots, \eta_1(u_k), \eta_2(v_1), \ldots, \eta_2(v_l)) \mid \langle u_1, \ldots, u_k \rangle = \overline{x_{r_2}}, \langle v_1, \ldots, v_l \rangle = \overline{x_{r_1}},$$

and $(r_1, \eta_1) <^{\mathcal{W}} (r_2, \eta_2)$

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

Claim 1. $\mathcal{M}' \models \forall \overline{S} \varphi_{\Pi}^{\mathcal{M}'}(\overline{S})$.

---

10Here $\infty$ denotes an arbitrary order-type.
Proof of Claim 1. To prove this claim, it is sufficient to show the following:

1. $M' = \bigwedge_{r_1, r_2, r_3 \in \Pi} \forall \bar{x}_{r_1}, \bar{x}_{r_2}, \bar{x}_{r_3} \left( <_{r_1, r_2} (\bar{x}_{r_1}, \bar{x}_{r_2}) \land <_{r_2, r_3} (\bar{x}_{r_2}, \bar{x}_{r_3}) \rightarrow <_{r_1, r_3} (\bar{x}_{r_1}, \bar{x}_{r_3}) \right)$

2. $M' = \bigwedge_{r_1, r_2 \in \Pi} \forall \bar{x}_{r_1}, \bar{x}_{r_2} \left( <_{r_1, r_2} (\bar{x}_{r_1}, \bar{x}_{r_2}) \rightarrow \neg <_{r_2, r_1} (\bar{x}_{r_2}, \bar{x}_{r_1}) \right)$

3. $M' = \bigwedge_{r_1, r_2 \in \Pi} \forall \bar{x}_{r_1}, \bar{x}_{r_2} \left( <_{r_1, r_2} (\bar{x}_{r_1}, \bar{x}_{r_2}) \rightarrow \varphi_r^{\text{GEN}}(\bar{x}_{r_1}) \land \varphi_r^{\text{GEN}}(\bar{x}_{r_2}) \right)$

4. $M' = \bigvee_{r \in \Pi} \varphi_r^{\text{WELL}}(\bar{S})$

Now we show each of the above statements.

1. $M' \models \bigwedge_{r_1, r_2, r_3 \in \Pi} \forall \bar{x}_{r_1}, \bar{x}_{r_2}, \bar{x}_{r_3} \left( <_{r_1, r_2} (\bar{x}_{r_1}, \bar{x}_{r_2}) \land <_{r_2, r_3} (\bar{x}_{r_2}, \bar{x}_{r_3}) \rightarrow <_{r_1, r_3} (\bar{x}_{r_1}, \bar{x}_{r_3}) \right)$

Suppose for some assignment $\alpha$, we have $M' \models <_{r_1, r_2} (\bar{a}_{r_1}, \bar{a}_{r_2}) \land <_{r_2, r_3} (\bar{a}_{r_2}, \bar{a}_{r_3})$ such that $\bar{a}_{r_1}$ and $\bar{a}_{r_2}$ denote the tuples obtained from $\bar{x}_{r_1}$ and $\bar{x}_{r_2}$, and $\bar{x}_{r_3}$ respectively by replacing each of the variable $x$ in $\bar{x}_{r_1}$ with $\alpha(x)$. We show $M' \models <_{r_1, r_3} (\bar{a}_{r_1}, \bar{a}_{r_3})$. From the definition of $<_{r_1, r_2}$, we have that there exists $(r_1, \eta_1) \in \Pi(\Pi)_M$ and $(r_2, \eta_2) \in \Pi(\Pi)_M$ with $(r_1, \eta_1) <_{r_1} (r_2, \eta_2)$ such that if $u_1, \ldots, u_k = \bar{x}_{r_2}$ and $\langle v_1, \ldots, v_l \rangle = \bar{x}_{r_3}$, then $\langle \eta_1(u_1), \ldots, \eta_1(u_k) \rangle = \bar{a}_{r_2}$ and $\langle \eta_2(v_1), \ldots, \eta_2(v_l) \rangle = \bar{a}_{r_3}$. Similarly, by the definition of $<_{r_2, r_3}$, there exists $(r_2, \eta'_2) \in \Pi(\Pi)_M$ and $(r_3, \eta_3) \in \Pi(\Pi)_M$ with $(r_2, \eta'_2) <_{r_2} (r_3, \eta_3)$ such that if $w_1, \ldots, w_m = \bar{x}_{r_3}$ and $\langle w_1, \ldots, w_m \rangle = \bar{x}_{r_3}$, then $\langle \eta'_2(w_1), \ldots, \eta'_2(w_m) \rangle = \bar{a}_{r_2}$ and $\langle \eta_3(w_1), \ldots, \eta_3(w_m) \rangle = \bar{a}_{r_3}$. Then, since $\langle \eta'_2(v_1), \ldots, \eta'_2(v_l) \rangle = \langle \eta_2(v_1), \ldots, \eta_2(v_l) \rangle$, we must have $\eta'_2 = \eta_2$. Then we have $(r_1, \eta_1) <_{r_1} (r_3, \eta_3)$ by transitivity since $(r_2, \eta'_2) = (r_2, \eta_2)$. Then from the definition of $<_{r_2, r_3}$, it follows that $\langle \bar{a}_{r_1}, \bar{a}_{r_3} \rangle \in <_{r_1, r_3}$, and hence, that $M' \models <_{r_1, r_3} (\bar{a}_{r_1}, \bar{a}_{r_3})$.

2. $M' \models \bigwedge_{r_1, r_2 \in \Pi} \forall \bar{x}_{r_1}, \bar{x}_{r_2} \left( <_{r_1, r_2} (\bar{x}_{r_1}, \bar{x}_{r_2}) \rightarrow \neg <_{r_2, r_1} (\bar{x}_{r_2}, \bar{x}_{r_1}) \right)$

Towards a contradiction, assume for some assignment $\alpha$, such that $M' \models <_{r_1, r_2} (\bar{a}_{r_1}, \bar{a}_{r_2}) \land <_{r_2, r_1} (\bar{a}_{r_2}, \bar{a}_{r_1})$. Then we also have $<_{r_1, r_1} (\bar{a}_{r_1}, \bar{a}_{r_1})$ by the transitivity axiom (which was already shown above to be satisfied by $M'$). Then by the definition of $<_{r_1, r_1}$, there exists $(r_1, \eta_1) \in \Lambda^t(\Pi)_M$ and $(r_1, \eta_2) \in \Lambda^t(\Pi)_M$ with $t_1 < t_2$ such that if $\langle u_1, \ldots, u_k \rangle = \bar{x}_{r_1}$, we have $\langle \eta_1(u_1), \ldots, \eta_1(u_k) \rangle = \bar{a}_{r_1}$ and $\langle \eta_2(u_1), \ldots, \eta_2(u_k) \rangle = \bar{a}_{r_1}$. Then this implies $\eta_1 = \eta_2$, and since there is a unique $t$ for which $(r_1, \eta) \in \Lambda^t(\Pi)_M$ 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. $M' \models \bigwedge_{r_1, r_2 \in \Pi} \forall \bar{x}_{r_1}, \bar{x}_{r_2} \left( <_{r_1, r_2} (\bar{x}_{r_1}, \bar{x}_{r_2}) \rightarrow \varphi_r^{\text{GEN}}(\bar{x}_{r_1}) \land \varphi_r^{\text{GEN}}(\bar{x}_{r_2}) \right)$

This follows from the “interpretations,” $<_{r_1, r_2}$, of the predicates $<_{r_1, r_2}$ (for $r_1, r_2 \in \Pi$) and where it is a “representation” of the well-order $W = (\Pi(\Pi)_M, <)$ on $\Pi(\Pi)_M$.

4. $M' \models \bigwedge_{r \in \Pi} \forall \bar{x}_{r} (\varphi_r^{\text{GEN}}(\bar{x}_{r}) \rightarrow \varphi_r^{\text{SUP}}(\bar{x}_{r}))$

Suppose for some assignment $\alpha$, we have $M' \models \varphi_r^{\text{GEN}}(\bar{a}_{r})$ such that $\bar{a}_{r}$ is the tuple obtained from $\bar{x}_{r}$ via $\alpha$ as above. Then we show $M' \models \forall \bar{a}_{r} \in \bar{a}_{r} \in P(os(\bar{r})) \left( \bigvee_{P \in P(os(\bar{r}))) H(c(e(\bar{r}))) = \bar{P}(\bar{r}) \right)$

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

(5) $M' \models \forall \overline{S} \varphi^{welor}_P(\overline{S})$. That is, we show that $M'$ satisfies

$$\forall \overline{S}((\bigwedge_{r \in \Pi} \exists \overline{x_r}(\overline{S}(\overline{x_r}) \rightarrow \varphi^{gen}_r(\overline{x_r}))) \rightarrow \left( \bigvee_{r \in \Pi} \exists \overline{x_r}(\overline{S}(\overline{x_r}) \land \forall \overline{y_r}(\overline{S}(\overline{y_r}) \land \overline{y_r} \neq \overline{x_r} \rightarrow <_{r'}(\overline{x_r}, \overline{y_r})) \right) \land \left( \bigwedge_{r \in \Pi} \forall \overline{x_r}(\overline{S}(\overline{x_r}) \rightarrow <_{r'}(\overline{x_r}, \overline{x_r})) \right):$$

This simply follows from the assumption that $W$ is a well-ordering on $\Gamma(\Pi)_M$ since a well-order implies that each non-empty subset contains a least element. Indeed, $\bigwedge_{r \in \Pi} \exists \overline{x_r}(\overline{S}(\overline{x_r}) \rightarrow \varphi^{gen}_r(\overline{x_r}))$ encodes the condition that we only consider tuples corresponding to the generating rules in the well-order, $\bigvee_{r \in \Pi} \exists \overline{x_r}(\overline{S}(\overline{x_r})$ encodes the condition that the subset of $\Gamma(\Pi)_M$ that we are considering is non-empty, and lastly,

$$\bigwedge_{r \in \Pi} \forall \overline{x_r}(\overline{S}(\overline{x_r}) \rightarrow <_{r'}(\overline{x_r}, \overline{x_r}))$$

encodes the condition that for such a (non-empty) subset, a least element exists.

This ends the proof of Claim 1.

Claim 2. $M' \models \varphi^{comp}_P$.

Proof of Claim 2. From [Zhang and Zhou 2010], $M^{\infty}(\Pi) = M$ iff $M \models \varphi$, where $\varphi$ is the SO sentence as defined in Section 2.3. Moreover, if $M \models \varphi$, then $M \models \varphi^{comp}_P$. This ends the proof of Claim 2.

So we have shown that $M \models \exists \overline{S} \forall \overline{S} \varphi^{welor}_P(\overline{S}, \overline{S})$.

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

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

Step. Assume that for $t' \leq t$, we have $M^{t'}(\Pi) \subseteq M$.

Then let $P(\overline{a_p}) \in M^{t'+1}(\Pi)$ such that $P(\overline{a_p}) \notin M^t(\Pi)$ (i.e., for if $P(\overline{a_p}) \in M^t(\Pi)$ then
the result is clear by the inductive hypothesis. We will now show $P(\overline{a_\phi}) \in \mathcal{M}$. Indeed, since $P(\overline{a_\phi}) \in \mathcal{M}^{r+1}(\Pi)$ with $P(\overline{a_\phi}) \notin \mathcal{M}(\Pi)$, then there exists a rule $r \in \Pi$ and an assignment $\eta$ such that $Head(r)\eta = P(\overline{a_\phi})$, $Pos(r)\eta \subseteq \mathcal{M}(\Pi)$, and $Neg(r)\eta \cap \mathcal{M} = \emptyset$. Then since $\mathcal{M}(\Pi) \subseteq \mathcal{M}$ by assumption, $\mathcal{M}^{r}_{\tau(\Pi)} = \mathcal{M}$ and $\mathcal{M} \models \phi_{W}^{COMP}$ (i.e., since $\phi_{W}^{COMP}$ only involves those symbols in $\tau(\Pi)$), it follows that $P(\overline{a_\phi}) \in \mathcal{M}$.

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

$$
\text{Dom}(W) = \Gamma(\Pi, \mathcal{M}): \quad <^W = \{ ((r_1, \eta_1), (r_2, \eta_2)) \mid r_1, r_2 \in \Pi, \eta_1 \text{ and } \eta_2 \text{ are assignments such that if } \overline{x_{r_1}} = \langle u_1, \ldots, u_k \rangle \text{ and } \overline{x_{r_2}} = \langle v_1, \ldots, v_l \rangle \text{ then } (\eta_1(u_1), \ldots, \eta_1(u_k), \eta_2(v_1), \ldots, \eta_2(v_l)) \in \mathcal{M}^r \}.
$$

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

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

where $\text{BOT}(\mathcal{W})$ denotes the least element of $\text{Dom}(\mathcal{W})$ under $W$; $\text{SUCC}(\alpha)$ the successor element of $\alpha$ under $W$; and $\text{ORD}(W)$, the order type of $W$ (that is, $\text{ORD}(W)$ is equal to the size of $\text{Dom}(W)$). As $W$ 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 $\mathcal{M} = \forall \overline{S} \phi_{W}^{\mathcal{M}}(\overline{S})$, 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 $\mathcal{M} \models \phi_{W}^{\text{COMP}}$ (which implies that $\mathcal{M}^{r}_{\tau(\Pi)} = \mathcal{M}$ is logically closed under $\Pi$), it follows from the definition of $\mathcal{M}^{1}(\Pi)$ that $Head(r)\eta \in \mathcal{M}^{1}(\Pi) \subseteq \mathcal{M}^\infty(\Pi)$. Hence, by the definition of $\mathcal{W}^{\text{BOT}(\mathcal{W})}(\Pi)$, we have $\mathcal{W}^{\text{BOT}(\mathcal{W})}(\Pi) = \{ Head(r)\eta \} \subseteq \mathcal{M}^\infty(\Pi)$.

**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

$$
\mathcal{M}^r = \bigwedge_{r \in \Pi} \forall \overline{x_r}(\phi_r^{\text{GEN}}(\overline{x_r}) \rightarrow \phi_r^{\text{SUP}}(\overline{x_r}))
$$

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

**Step.** Assume for $\text{BOT}(\mathcal{W}) \leq^W \beta \leq^W \alpha$, 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 above, it follows that $Head(r)\eta \in \mathcal{M}^{1}(\Pi) \subseteq \mathcal{M}^\infty(\Pi)$, which implies $\mathcal{W}^{\text{SUCC}(\alpha)}(\Pi) = \mathcal{W}^{\alpha}(\Pi) \cup \{ Head(r)\eta \} \subseteq \mathcal{M}^\infty(\Pi)$.

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

Then as

$$
\mathcal{M}^r = \bigwedge_{r \in \Pi} \forall \overline{x_r}(\phi_r^{\text{GEN}}(\overline{x_r}) \rightarrow \phi_r^{\text{SUP}}(\overline{x_r}))
$$
(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 $P(\overline{a}) \in \mathcal{M}$ where $P \in \mathcal{P}_{\text{int}}(\Pi)$ (i.e., for if $P \notin \mathcal{P}_{\text{int}}(\Pi)$, then it immediately follows that $P(\overline{a}) \in \mathcal{M}^{0}(\Pi) \subseteq \mathcal{M}^{1}(\Pi)$). As $\mathcal{M} \models \varphi_{\text{COMP}}^{\Pi}$ implies $\mathcal{M} \models \varphi_{\text{COMP}}^{\Pi}$ (since $\varphi_{\text{COMP}}^{\Pi}$ only involves the symbols occurring in $\tau(\Pi)$), then for some rule $r \in \Pi$ and assignment $\eta$, we have $\text{Head}(r)\eta = P(\overline{a})$, $\text{Pos}(r)\eta \subseteq \mathcal{M}$ 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 $\text{Head}(r)\eta = P(\overline{a}) \in \mathcal{M}^{1}(\Pi) \subseteq \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \cup \mathcal{M}^{1}(\Pi)$.

**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 $P(\overline{a}) \in \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \subseteq \mathcal{W}^{\text{ORD}(\mathcal{W})}(\Pi) \cup \mathcal{M}^{1}(\Pi)$.

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

**Proof of Theorem 5.7**

**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 $\mathcal{M} \models \exists \exists S \varphi_{(\Pi, <)}(\overline{S}, \overline{S})$ in two steps:

(1) Given $\lambda_{\mathcal{M}^{0}}(\Gamma^{\infty}(\Pi),\mathcal{M}) = \mathcal{M}$, we show there exists a well-order $\mathcal{W} = (\Gamma(\Pi), <^{\mathcal{W}})$ 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) $\text{Pos}(r)\eta \subseteq \mathcal{W}^{0}(\Pi) \cup \{\text{Head}(r)\eta' | (r', \eta') <^{\mathcal{W}} (r, \eta)\}$;

(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. $\text{Neg}(r') \eta' \cap \mathcal{M}^{0}(\Pi) \cup \{\text{Head}(r''\eta'') | (r'', \eta'') <^{\mathcal{W}} (r, \eta)\} \neq \emptyset$.

(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 $\mathcal{M}' \models \forall \exists S \varphi_{(\Pi, <)}(\overline{S})$ where $\forall \exists S \varphi_{(\Pi, <)}(\overline{S})$ is the sentence obtained from $\mathcal{M} \models \forall \exists S \varphi_{(\Pi, <)}(\overline{S})$ by treating the predicate variables in $\overline{S}$ 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, $(r, \eta) \in \Gamma(\Pi),\mathcal{M}$ if $r$ 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 $\mathcal{W} = (\Gamma^{t}(\Pi), <^{\mathcal{W}})$ on $\Gamma^{t}(\Pi),\mathcal{M}$ with the following properties. For each $(r, \eta) \in \Gamma^{t}(\Pi),\mathcal{M}$:
Thus, ultimately, since $\Gamma^\infty(\Pi, M) = \Gamma(\Pi, M)$ 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 $\mathcal{W} = (\Gamma^0(\Pi, M), <^W)$ on $\Gamma^0(\Pi, M)$. 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 $(r, \eta)$, $(r', \eta') \in \Gamma^0(\Pi, M), r < r'$ implies $(r, \eta) <^W (r', \eta')$. To see this, note that we can partition $\Gamma^0(\Pi, M)$ into a sequence of sets $S_{r_1}, \ldots, S_{r_n}$ where $r_1 < r_2 < \cdots < r_n$. Then, by the well-ordering theorem, there exists a well-order $\mathcal{W}_i = (S_{r_i}, <^{W_i})$ of each of the sets $S_{r_i}$. Thus, we can just set $\mathcal{W} = (\Gamma^0(\Pi, M), <^W)$ by setting $<^W = (\bigcup_{1 \leq i \leq n} <^{W_i}) \cup \{(r_i, \eta_i, (r_j, \eta_j)) | (r_i, \eta_i) \in S_{r_i}, (r_j, \eta_j) \in S_{r_j}, i < j\}$. 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.** $(r, \eta) \in \Gamma^0(\Pi, M)$ implies $\text{Pos}(r, \eta) \subseteq \mathcal{M}^0(\Pi)$ and $\{\text{Head}(r', \eta') | (r', \eta') <^W (r, \eta)\}$.

**Proof of Claim 2.** Clearly by the definition of $\Gamma^0(\Pi, M), (r, \eta) \in \Gamma^0(\Pi, M)$ implies $\text{Pos}(r, \eta) \subseteq \mathcal{M}^0(\Pi)$. This completes the proof of Claim 2.

**Claim 3.** If $(r, \eta) \in \Gamma^0(\Pi, M)$, then $(r', \eta') \in \Gamma^0(\Pi, M)$, where $r < r'$ implies $(r', \eta') <^W (r, \eta)$.

**Proof of Claim 3.** Follows from the description of $\mathcal{W}$. This ends the proof of Claim 3.

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

1. $\text{Pos}(r', \eta') \notin \mathcal{M}$ or
2. $\text{Neg}(r', \eta') \cap \{\text{Head}(r'', \eta'') | (r'', \eta'') <^W (r, \eta)\} \neq \emptyset$.

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

**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 $\mathcal{W}' = (\Gamma^{t+1}(\Pi, M), <^{W'})$ on $\Gamma^{t+1}(\Pi, M)$ satisfying the conditions of the preference preserving well-order. Moreover, by the well-ordering theorem, there also exists a well-order $\mathcal{W}'' = (\Lambda^{t+1}(\Pi, M), <^{W''})$ on $\Lambda^{t+1}(\Pi, M)$ where $\Lambda^{t+1}(\Pi, M) = \Gamma^{t+1}(\Pi, M) \setminus \Gamma^0(\Pi, M)$. 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 $(r, \eta), (r', \eta') \in \Lambda^{t+1}(\Pi, M), r < r'$ implies $(r, \eta) <^{W''} (r', \eta')$ (i.e. using the same argument as above). Now we specify $\mathcal{W}'' = (\Gamma^{t+1}(\Pi, M), <^{W''})$ to be a well-order on $\Gamma^{t+1}(\Pi, M)$ by setting $<^{W''} = \mathcal{W} \cup \mathcal{W}' \cup (\Gamma(\Pi, M) \times \Lambda^{t+1}(\Pi, M))$.

**Claim 5.** $(r, \eta) \in \Gamma^{t+1}(\Pi, M)$ implies $\text{Pos}(r, \eta) \subseteq \mathcal{M}^0(\Pi)$ and $\{\text{Head}(r', \eta') | (r', \eta') <^{W''} (r, \eta)\}$.

**Proof of Claim 5.** We consider the possibilities:

**Case 1.** $(r, \eta) \in \Gamma^t(\Pi, M)$.

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


Case 2. \((r, \eta) \in \Lambda^{t+1}(\Pi)\_M\).

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

This completes the proof of Claim 5.

Claim 6. If \((r, \eta) \in \Gamma^{t+1}(\Pi)\_M\), then \((r', \eta') \in \Gamma^{t+1}(\Pi)\_M\) where \(r' < r\) implies \((r', \eta') < W''(r, \eta)\).

Proof of Claim 6. Towards a contradiction, assume \((r, \eta), (r', \eta') \in \Gamma^{t+1}(\Pi)\_M\), \(r' < r\), where \((r, \eta) < W''(r', \eta')\). We also consider all possibilities:

Case 1. \((r, \eta), (r', \eta') \in \Gamma^t(\Pi)\_M\).

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

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

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

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

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

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

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

Case 4. \((r, \eta), (r', \eta') \in \Lambda^{t+1}(\Pi)\_M\).

Then since \(W < W''\) and by the construction of the well-order \(W''\) on \(\Lambda^{t+1}(\Pi)\_M\), then we must have \((r', \eta') < W''(r, \eta)\) since \(r' < r\) by assumption. This is contrary to the initial assumption \((r, \eta) < W''(r', \eta')\).

This ends the proof of Claim 6.

Claim 7. If \((r, \eta) \in \Gamma^{t+1}(\Pi)\_M\), then \((r', \eta') \notin \Gamma(\Pi)\_M\) where \(r' < r\) implies that either:

1. \(\text{Pos}(r')\eta' \subseteq \mathcal{M}\) or
2. \(\text{Neg}(r')\eta' \cap (\mathcal{M}^0(\Pi) \cup \{\text{Head}(r')\eta'' \mid (r'', \eta'') < W''(r, \eta)\}) \neq \emptyset\).

Proof of Claim 7. We again consider the possibilities:

Case 1. \((r, \eta) \in \Gamma^t(\Pi)\_M\).

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

Case 2. \((r, \eta) \in \Lambda^{t+1}(\Pi)\_M\).

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

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

Then this contradicts the assumption \((r, \eta) \in \Lambda^{t+1}(\Pi)_M = \Gamma^{t+1}(\Pi)_M \setminus \Gamma'(\Pi)_M\) (i.e. \((r, \eta)\) is applied in \(\Gamma^{t+1}(\Pi)_M\)) since by the definition of \(\Gamma^{t+1}(\Pi)_M\), 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 \(M \models \exists \exists \forall S \varphi_{(\Pi, <)}(\exists, S)\). As mentioned in the beginning, based on the result of the first part (i.e. that there exists a preference preserving well-order \(W = (\Gamma(\Pi)_M, <^W)\) on the set \(\Gamma(\Pi)_M\), we now construct an expansion \(M'\) of \(M\) based on the well-order \(W\). Thus, set the structure \(M'\) to be an expansion of \(M\) on the signature \(\tau(\Pi) \cup \{<_{r_1r_2} | r_1, r_2 \in \Pi\}\) (i.e., where \(M\) is of the signature \(\tau(\Pi)\)) such that for each (new) predicate symbol \(<_{r_1r_2}\) (i.e. where \(r_1, r_2 \in \Pi\) and \(r_1\) and \(r_2\) could be the same), we have

\[
<^M_{r_1r_2} = \{ (\eta_1(u_1), \ldots, \eta_1(u_k), \eta_2(v_1), \ldots, \eta_2(v_l)) \mid (u_1, \ldots, u_k, v_1, \ldots, v_l) = \bar{x}^r_{r_2}, (r_1, \eta_1) <^W (r_2, \eta_2) \}.
\]

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

\((\Leftarrow)\) Assume \(M \models \exists \exists \forall S \varphi_{(\Pi, <)}(\exists, S)\). We show \(\lambda_M(\Gamma(\Pi)_M) = M\) by showing \(\lambda_M(\Gamma(\Pi)_M) \subseteq M\) and \(M \subseteq \lambda_M(\Gamma(\Pi)_M)\). Indeed, since \(M \models \exists \exists \forall S \varphi_{(\Pi, <)}(\exists, S)\) implies \(M \models \exists \exists \forall S (\varphi^\Pi_{\text{PRO}}(\exists, S) \land \varphi^\Pi_{\text{COMP}})\), then \(M\) is an answer set of \(\Pi\) by Theorem 5.5. Thus, from [Zhang and Zhou 2010], we have \(\lambda_M(\Gamma(\Pi)_M) = M\). Then since \(\lambda_M(\Gamma(\Pi)_M) \subseteq M\), it follows that \(\lambda_M(\Gamma(\Pi)_M) = M\). Therefore, it is only left for us to show \(M \subseteq \lambda_M(\Gamma(\Pi)_M)\).

Now, since \(M \models \exists \exists \forall S \varphi_{(\Pi, <)}(\exists, S)\), there exists an expansion \(M'\) of \(M\), on the signature \(\tau(\Pi) \cup \{<_{r_1r_2} | r_1, r_2 \in \Pi\}\), such that \(M' \models \forall S \varphi_{(\Pi, <)}(\exists, S)\) where \(\forall S \varphi_{(\Pi, <)}(\exists, S)\) is the sentence obtained from \(\forall S \varphi_{(\Pi, <)}(\exists, S)\) by treating the predicate variables in \(\exists\) as constants. Then since \(M' \models \forall S \varphi_{(\Pi, <)}(\exists, S)\) (i.e. which implies a well-order on \(\Gamma(\Pi)_M\), we construct a well-order \(W = (\Gamma(\Pi)_M, <^W)\) of \(\Gamma(\Pi)_M\) by setting

\[
<^W = \{(r_1, \eta_1), (r_2, \eta_2) \mid r_1, r_2 \in \Pi, \bar{x}^r_{r_2} = \{u_1, \ldots, u_k\}, \bar{x}^r_{r_1} = \{v_1, \ldots, v_l\}, (\eta_1(u_1), \ldots, \eta_1(u_k), \eta_2(v_1), \ldots, \eta_2(v_l)) \in <^M_{r_1r_2}\}.
\]

Hence, for an \(\alpha \in \Gamma(\Pi)_M\), define \(W^\alpha(\Pi)\) inductively as follows:

- \(W^\text{BOT}(\Pi) = \{\text{BOT}(W)\}\);
- \(W^\text{SUC}(\alpha)(\Pi) = W^\alpha(\Pi) \cup \{\text{SUC}(\alpha)\}\),

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

\text{Claim 1. } W^\text{ORD}(\Pi) \subseteq \Gamma(\Pi)_M.$

\text{Proof of Claim 1.} We prove by induction on \(\alpha\) for \(\alpha \geq \text{BOT}(W)\).

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

1. \(\text{Pred}(\text{Pos}(r)) \cap \text{Pout}(\Pi) = \emptyset\) (i.e., since \((r, \eta)\) is the least element under \(W\) of \(\Gamma(\Pi)_M\) and that the well-order \(W\) satisfies the notion of \text{support}\), since \(M' \models \Lambda_{\forall \bar{x}_r} (\varphi^\text{SUC}(\bar{x}_r) \rightarrow \varphi^\text{SUP}(\bar{x}_r))\);
(2) Any rule \( r' < r \) and assignment \( \eta' \) implies \( (r', \eta') \notin \Gamma(\Pi)_M \). For suppose \((r', \eta') \in \Gamma(\Pi)_M \), then we must have \((r', \eta') <_W (r, \eta) \) since \( r' < r \), \((r, \eta) \) and \((r', \eta') \) are both in \( \Gamma(\Pi)_M \), and

\[
\mathcal{M}' \models \forall \overrightarrow{x}_r (\varphi^{\text{GEN}}_r (\overrightarrow{x}_r) \rightarrow \bigwedge_{r' < r} \forall \overrightarrow{x}_{r'} (\varphi^{\text{GEN}}_{r'} (\overrightarrow{x}_{r'}) \rightarrow <_{r'} (\overrightarrow{x}_{r'}, \overrightarrow{x}_r)))
\]

(i.e., since \( \mathcal{M}' \models \varphi^{\text{PREF}}_{\langle r, r' \rangle} \)), which is absurd since \((r, \eta)\) is the least under \( W \);

(3) For each rule \( r' < r \) and assignment \( \eta' \) with \((r', \eta') \notin \Gamma(\Pi)_M \), we have either:

(a) \( \text{Pos}(r')\eta' \not\subseteq M \) or

(b) \( \text{Neg}(r')\eta' \cap (\mathcal{M}^0(\Pi)_M \cup \{ \text{Head}(r')\eta'' \mid (r'', \eta'') <_W (r, \eta) \}) \neq \emptyset \)

since

\[
\mathcal{M}' \models \forall \overrightarrow{x}_r (\varphi^{\text{GEN}}_r (\overrightarrow{x}_r) \rightarrow \bigwedge_{r' < r} \forall \overrightarrow{x}_{r'} (\neg \varphi^{\text{GEN}}_{r'} (\overrightarrow{x}_{r'}) \rightarrow (\Phi^{\text{POS}}_r (\overrightarrow{x}_r) \lor \Phi^{\text{DEF}}_r (\overrightarrow{x}_r, \overrightarrow{x}_{r'})))).
\]

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

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

Step. Assume for \( \text{Bot}(W) \leq W \beta \leq W \alpha \), we have \( W^\beta(\Pi) \subseteq \Gamma^\alpha(\Pi)_M \).

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

Subclaim 1.: There does not exists a rule \( r' < r \) and assignment \( \eta' \) such that

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

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. \((r', \eta') \in \Gamma(\Pi)_M \).

Then as

\[
\mathcal{M}' \models \forall \overrightarrow{x}_r (\varphi^{\text{GEN}}_r (\overrightarrow{x}_r) \rightarrow \bigwedge_{r' < r} \forall \overrightarrow{x}_{r'} (\varphi^{\text{GEN}}_{r'} (\overrightarrow{x}_{r'}) \rightarrow <_{r'} (\overrightarrow{x}_{r'}, \overrightarrow{x}_r)))
\]

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

Case 2. \((r', \eta') \notin \Gamma(\Pi)_M \).

Then since

\[
\mathcal{M}' \models \forall \overrightarrow{x}_r (\varphi^{\text{GEN}}_r (\overrightarrow{x}_r) \rightarrow \bigwedge_{r' < r} \forall \overrightarrow{x}_{r'} (\neg \varphi^{\text{GEN}}_{r'} (\overrightarrow{x}_{r'}) \rightarrow (\Phi^{\text{POS}}_r (\overrightarrow{x}_r) \lor \Phi^{\text{DEF}}_r (\overrightarrow{x}_r, \overrightarrow{x}_{r'}))))
\]

there can only be two possibilities:

Subcase 1. \( \text{Pos}(r')\eta' \not\subseteq M \).

Then this contradicts the assumption \( \text{Pos}(r')\eta' \subseteq M \).
Subcase 2. \( \neg \text{Neg}(r') \land \lambda_{M^0}(W^\alpha(\Pi)) \neq \emptyset. \)

Then this contradicts the assumption \( \neg \text{Neg}(r') \land \lambda_{M^0}(\Gamma^t(\Pi)) = \emptyset \) since \( W^\alpha(\Pi) \subseteq \Gamma^t(\Pi) \), by the inductive hypothesis.

This completes the proof of Subclaim 1.

Hence, by Subclaim 1, we must have \((r, \eta) \in \Gamma^{t+1}(\Pi)\) (i.e., since there is no pair \((r', \eta')\) blocking \((r, \eta)\) from being applied at stage \(t+1\)), which then implies \( W^{\SUCC(\alpha)}(\Pi) \subseteq \Gamma^\infty(\Pi) \).

This completes the proof of Claim 1.

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

1. \( \text{Head}(r) \eta = P(\bar{a}_P) \);
2. \( \text{Pos}(r) \eta \subseteq \lambda_{M^0}(\Pi) = \mathcal{M} \).

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

This completes the proof of the theorem.

**Proof of Proposition 6.4**

**Proof.**

1. Using the assumption \( \lambda_{M^0}(\Gamma^W)(\Pi)) = \mathcal{M} \), we show \( \lambda_{M^0}(\Gamma^W(\Pi)) \subseteq \mathcal{M}^{\infty}(\Pi) \) and \( \mathcal{M}^{\infty}(\Pi) \subseteq \lambda_{M^0}(\Gamma^W(\Pi)) \). First we show \( \lambda_{M^0}(\Gamma^W(\Pi)) \subseteq \mathcal{M}^{\infty}(\Pi) \) by induction.

   **Base.** Let \( P(\lambda \bar{a}_P) \in \lambda_{M^0}(\Gamma^W(\Pi)) \). Then by the definition of \( \lambda_{M^0}(\Gamma^W(\Pi)) \), there exists some \((r, \eta) \in \Gamma^W(\Pi) \) with \( \text{Head}(r) \eta = P(\bar{a}_P) \), \( \text{Pos}(r) \eta \subseteq \mathcal{M}^{\infty}(\Pi) \), and \( \text{Neg}(r) \eta \cap \mathcal{M} = \emptyset \). Then by the definition of \( \mathcal{M}^{\infty}(\Pi) \), we have \( P(\lambda \bar{a}_P) \in \mathcal{M}^{\infty}(\Pi) \).

   **Step.** Assume for \( 1 \leq t' \leq t \) we have \( \lambda_{M^0}(\Gamma^{t'}(\Pi)) \subseteq \mathcal{M}^{\infty}(\Pi) \).

   Now, let \( P(\lambda \bar{a}_P) \in \lambda_{M^0}(\Gamma^{t'+1}(\Pi)) \) where \( P(\lambda \bar{a}_P) \notin \lambda_{M^0}(\Gamma^t(\Pi)) \) (i.e., since if \( P(\lambda \bar{a}_P) \in \lambda_{M^0}(\Gamma^t(\Pi)) \), the result immediately follows by the inductive hypothesis). Then there exists an \( (r, \eta) \) with \( \text{Head}(r) \eta = P(\lambda \bar{a}_P) \) such that \( \text{Pos}(r) \eta \subseteq \lambda_{M^0}(\Gamma^t(\Pi)) \) and \( \text{Neg}(r) \eta \cap \mathcal{M} = \emptyset \). Now, since \( \lambda_{M^0}(\Gamma^t(\Pi)) \subseteq \mathcal{M}^{\infty}(\Pi) \) by assumption, then for some \( n \geq 0 \), we have \( \lambda_{M^0}(\Gamma^{t+n}(\Pi)) \subseteq \mathcal{M}^{\infty}(\Pi) \). Then this implies \( \text{Pos}(r) \eta \subseteq \mathcal{M}^{\infty}(\Pi) \) and \( \text{Neg}(r) \eta \cap \mathcal{M} = \emptyset \), which further implies \( \text{Head}(r) \eta = P(\lambda \bar{a}_P) \in \mathcal{M}^{\infty}(\Pi) \) by the definition of \( \mathcal{M}^{\infty}(\Pi) \). Hence, we also have \( \lambda_{M^0}(\Gamma^{t+1}(\Pi)) \subseteq \mathcal{M}^{\infty}(\Pi) \).

2. Now we show \( \mathcal{M}^{\infty}(\Pi) \subseteq \lambda_{M^0}(\Gamma^W(\Pi)) \). Thus, by induction on \( t \) for \( 0 \leq t \), we show \( \mathcal{M}^{t}(\Pi) \subseteq \lambda_{M^0}(\Gamma^W(\Pi)) \) by showing that:

   (a) \( \mathcal{M}^0(\Pi) \subseteq \lambda_{M^0}(\Gamma^W(\Pi)) \), which immediately follows from the definition of \( \lambda_{M^0}(\Gamma^W(\Pi)) \) since \( \lambda_{M^0}(\Gamma^W(\Pi)) = \mathcal{M}^0(\Pi) \cup \{ \text{Head}(r) \eta \mid (r, \eta) \in \Gamma^W(\Pi) \}; \)

   (b) Assuming \( \mathcal{M}^t(\Pi) \subseteq \lambda_{M^0}(\Gamma^W(\Pi)) \) for \( 0 \leq t' \leq t \), we show \( \mathcal{M}^{t+1}(\Pi) \subseteq \lambda_{M^0}(\Gamma^W(\Pi)) \) by deriving a contradiction in a similar manner to the proof of Proposition 3.5 under the assumption \( \mathcal{M}^{t+1}(\Pi) \subseteq \lambda_{M^0}(\Gamma^W(\Pi)) \). That is, under that assumption, there must exist the rule \( r \) and assignment \( \eta \) such that \( \text{Pos}(r) \eta \subseteq \mathcal{M}^{t}(\Pi) \) and \( \text{Neg}(r) \eta \cap \mathcal{M} = \emptyset \) with \( \text{Head}(r) \eta \notin \lambda_{M^0}(\Gamma^W(\Pi)) \), which then implies \( (r, \eta) \notin \Gamma^W(\Pi) \). Now,
under the inductive hypothesis $M^t(\Pi) \subseteq \lambda_{M^0}(\Gamma_{W}(\Pi,M))$, there must be some ordinal $n \geq 1$ such that $M^t(\Pi) \subseteq \lambda_{M^0}(\Gamma_{W}(\Pi,M))$, which implies $Pos(r)\eta \subseteq \lambda_{M^0}(\Gamma_{W}(\Pi,M))$. Then since $(r, \eta) \notin \Gamma_{W}^{n+1}(\Pi,M)$ (i.e., since $(r, \eta) \notin \Gamma_{W}^{n}(\Pi,M)$ by assumption), there must be a pair $(r', \eta')$ with $r' < r$ such that:

(i) $Pos(r')\eta' \subseteq M$ and $Neg(r')\eta' \cap \lambda_{M^0}(\Gamma_{W}(\Pi,M)) = \emptyset$;

(ii) $Head(r')\eta' \notin \lambda_{M^0}(\Gamma_{W}(\Pi,M))$.

That is, $(r', \eta')$ is blocking $(r, \eta)$ from being applied at stage $n + 1$. Then similarly as in the proof of Proposition 3.5, for a $k \geq 1$, we set $B^k(r, \eta)$ as those pairs $(r', \eta')$ blocking $(r, \eta)$ only this time, $B^k(r, \eta)$ is set to be

$$B^1(r, \eta) := \{(r', \eta') \mid r' < r, Pos(r')\eta' \subseteq M \text{ and } Neg(r')\eta' \cap \mathcal{M}^0(\Pi) = \emptyset \} \text{ and } B^k(r, \eta) := \{(r', \eta') \mid (a) r' < r;$$

(b) $Pos(r')\eta' \subseteq M \text{ and } Neg(r')\eta' \cap \lambda_{M^0}(\Gamma_{W}(\Pi,M)) = \emptyset$;

(c) $Head(r')\eta' \notin \lambda_{M^0}(\Gamma_{W}(\Pi,M))$\}

for $k > 1$. Thus, through the assumption $\lambda_{M^0}(\Gamma_{W}(\Pi,M)) = \mathcal{M}$ and the fact that $(r', \eta') \in \Gamma_{W}^k(\Pi)$ implies $Head(r')\eta' \in \lambda_{M^0}(\Gamma_{W}(\Pi,M))$, it can also be shown as in Claim 3 in the proof of Proposition 3.5 that there exists some ordinal $n' > n$ for which we have $B^{n'}(r, \eta) = \emptyset$, which renders $(r, \eta)$ to be applicable at stage $n' + 1$ in $\Gamma_{W}^{n'+1}(\Pi,M)$ (i.e. $(r, \eta) \in \Gamma_{W}^{n'+1}(\Pi,M)$). Then this contradicts the assumption $Head(r)\eta \notin \lambda_{M^0}(\Gamma_{W}(\Pi,M))$ since $(r, \eta) \in \Gamma_{W}^n(\Pi,M)$ implies $Head(r)\eta \in \lambda_{M^0}(\Gamma_{W}(\Pi,M))$ by the definition of $\lambda_{M^0}(\Gamma_{W}(\Pi,M))$. Hence, we must have $M^{t+1}(\Pi) \subseteq \lambda_{M^0}(\Gamma_{W}(\Pi,M))$.

Therefore, as we could show $M^t(\Pi) \subseteq \lambda_{M^0}(\Gamma_{W}(\Pi,M))$ for all $t \geq 0$, then we conclude $M^{\infty}(\Pi) \subseteq \lambda_{M^0}(\Gamma_{W}(\Pi,M))$ as well. This completes the proof of Proposition 6.4.

\section*{Proof of Theorem 6.6}

**Proof.**

To prove this theorem, we provide an alternative reformulation of $\Gamma_{W}(\Pi,M)$ and show that they are “essentially equivalent”. Let us denote the alternative reformulation by $\Lambda_{W}(\Pi,M)$ and define it inductively as follows:

$$\Lambda_{W}(\Pi,M) = \{(r, \eta) \mid (1) Pos(r)\eta \subseteq \mathcal{M}^{0}(\Pi) \text{ and } Neg(r)\eta \cap \mathcal{M} = \emptyset;$$

(2) there do not exist a rule $r' \in \Pi$ and an assignment $\eta'$ such that $r' < r, Pos(r')\eta' \subseteq M \text{ and } Neg(r')\eta' \cap \mathcal{M}^{0}(\Pi) = \emptyset\};$

$$\Lambda_{W}^{t+1}(\Pi,M) = \Lambda_{W}(\Pi,M) \cup \{(r, \eta) \mid (1) (a) Pos(r)\eta \subseteq \lambda_{M^0}(\Lambda_{W}(\Pi,M)) \text{ and }$$

$Neg(r)\eta \cap \mathcal{M} = \emptyset \text{ or,}$

(b) $(r, \eta) \in \Gamma(\Pi,M)$ and $Head(r)\eta \in \lambda_{M^0}(\Lambda_{W}(\Pi,M));$

(2) there do not exist a rule $r' \in \Pi$ and an assignment $\eta'$ such that $r' < r, Head(r')\eta' \notin \lambda_{M^0}(\Lambda_{W}(\Pi,M))$, and $Pos(r')\eta' \subseteq M \text{ and } Neg(r')\eta' \cap \lambda_{M^0}(\Lambda_{W}(\Pi,M)) = \emptyset\}.$

Note that $\Gamma_{W}(\Pi,M)$ really only differs from $\Lambda_{W}(\Pi,M)$ in that $\Lambda_{W}(\Pi,M)$ has a more liberal condition since generating pairs $(r, \eta) \in \Gamma(\Pi,M)$ can be derived at stage $t + 1$ so long as their heads are already derived in the earlier stages.
Claim 1: \( \lambda_{\mathcal{M}^0}(\Lambda^t_W(\Pi_M)) = \lambda_{\mathcal{M}^0}(\Gamma^t_W(\Pi_M)) \) for all \( t \geq 0 \).

Proof of Claim 1: We prove this theorem by induction on \( t \).

**Basis:** Clearly, we have that \( \lambda_{\mathcal{M}^0}(\Lambda^0_W(\Pi_M)) = \lambda_{\mathcal{M}^0}(\Gamma^0_W(\Pi_M)) \) by the definitions of \( \Lambda^0_W(\Pi_M) = \Lambda_0 \) and \( \Gamma^0_W(\Pi_M) = \Gamma_0 \).

**Step:** Assume for \( 0 \leq t' \leq t \) that \( \lambda_{\mathcal{M}^0}(\Lambda^t_W(\Pi_M)) = \lambda_{\mathcal{M}^0}(\Gamma^t_W(\Pi_M)) \) holds. We show that this also holds for \( t + 1 \).

Thus, let \( P(\overline{a}) \in \lambda_{\mathcal{M}^0}(\Lambda^{t+1}_W(\Pi_M)) \) such that \( P(\overline{a}) \notin \lambda_{\mathcal{M}^0}(\Lambda^t_W(\Pi_M)) \) (otherwise it immediately holds by the inductive hypothesis). Then there must exist pair \( (r, \eta) \) where \( Head(r) = P(\overline{a}) \) such that \( (r, \eta) \) was derived in \( \lambda_{\mathcal{M}^0}(\Lambda^t_W(\Pi_M)) \) at stage \( t + 1 \). Then given that \( \lambda_{\mathcal{M}^0}(\Lambda^t_W(\Pi_M)) = \lambda_{\mathcal{M}^0}(\Gamma^t_W(\Pi_M)) \) by the inductive hypothesis, it follows that \( (r, \eta) \in \Lambda^t_W(\Pi_M) \) by the definitions of both \( \Lambda^t_W(\Pi_M) \) and \( \Gamma^t_W(\Pi_M) \), which implies that \( Head(r) = P(\overline{a}) \in \lambda_{\mathcal{M}^0}(\Gamma^{t+1}_W(\Pi_M)) \). Now let \( P(\overline{a}) \in \lambda_{\mathcal{M}^0}(\Gamma^{t+1}_W(\Pi_M)) \) such that \( P(\overline{a}) \notin \lambda_{\mathcal{M}^0}(\Gamma^t_W(\Pi_M)) \). Then similarly, there exists a pair \( (r, \eta) \) with \( Head(r) = P(\overline{a}) \) such that \( (r, \eta) \) was derived in \( \lambda_{\mathcal{M}^0}(\Gamma^{t+1}_W(\Pi_M)) \) at stage \( t + 1 \). Then, since \( \lambda_{\mathcal{M}^0}(\Lambda^t_W(\Pi_M)) = \lambda_{\mathcal{M}^0}(\Gamma^t_W(\Pi_M)) \) by assumption and from the definitions of \( \Gamma^{t+1}_W(\Pi_M) \) and \( \Lambda^{t+1}_W(\Pi_M) \), it follows that \( (r, \eta) \in \Lambda^{t+1}_W(\Pi_M) \), which implies that \( Head(r) = P(\overline{a}) \in \lambda_{\mathcal{M}^0}(\Lambda^{t+1}_W(\Pi_M)) \). This completes the proof of Claim 1.

Therefore by Claim 1, to prove it is sufficient to show that the following holds:

\[ \mathcal{M} \models \exists \overline{z} \forall \overline{S} \left( \varphi_{\Pi,<}^W(\overline{z}) \land \varphi_{\Pi}^W(\overline{z}, \overline{S}) \land \varphi_{\Pi}^{COMP} \right) \]

**iff** \( \lambda_{\mathcal{M}^0}(\Lambda^\infty_W(\Pi_M)) = \mathcal{M} \).

**Assume**

\[ \exists \overline{z} \forall \overline{S} \left( \varphi_{\Pi,<}^W(\overline{z}) \land \varphi_{\Pi}^W(\overline{z}, \overline{S}) \land \varphi_{\Pi}^{COMP} \right). \]

We show \( \tau_{\mathcal{M}^0}(\Lambda^\infty_W(\Pi_M)) = \mathcal{M} \) by showing \( \tau_{\mathcal{M}^0}(\Lambda^\infty_W(\Pi_M)) \subseteq \mathcal{M} \) and \( \mathcal{M} \subseteq \tau_{\mathcal{M}^0}(\Lambda^\infty_W(\Pi_M)) \).

Indeed, since \( \mathcal{M} \models \varphi_{\Pi}^{COMP} \) (which implies that \( \mathcal{M} \) is "logically closed" under \( \Pi \)), then it can be shown by easy induction that \( \tau_{\mathcal{M}^0}(\Lambda^\infty_W(\Pi_M)) \subseteq \mathcal{M} \) for all \( t \geq 0 \), since the rules that we are collecting under \( \Lambda^t_W(\Pi_M) \) are only those of generating rules under \( \mathcal{M} \). Therefore, it is only left for us to show that \( \mathcal{M} \subseteq \tau_{\mathcal{M}^0}(\Lambda^\infty_W(\Pi_M)) \).

Now, since

\[ \mathcal{M} \models \exists \overline{z} \forall \overline{S} \left( \varphi_{\Pi,<}^W(\overline{z}) \land \varphi_{\Pi}^W(\overline{z}, \overline{S}) \land \varphi_{\Pi}^{COMP} \right), \]

then there exists an expansion \( \mathcal{M}' \) of \( \mathcal{M} \) on the signature \( \tau(\Pi) \cup \{<_{r_1,r_2} | r_1, r_2 \in \Pi\} \) such that

\[ \mathcal{M}' \models \forall \overline{S} \left( \varphi_{\Pi,<}^W(\overline{z}) \land \varphi_{\Pi}^W(\overline{z}, \overline{S}) \land \varphi_{\Pi}^{COMP} \right). \]

Then we can construct a well-order \( W = (\Gamma(\Pi)_M, <^W) \) on \( \Gamma(\Pi)_M \) by setting

\[
<^W = \{(r_1, \eta_1), (r_2, \eta_2) | r_1, r_2 \in \Pi, \overline{x}_{r_1} = \langle u_1, \ldots, u_k \rangle, \overline{x}_{r_2} = \langle v_1, \ldots, v_i \rangle, \\
\eta_1(u_1), \ldots, \eta_k(u_k), \eta_2(v_1), \ldots, \eta_2(v_i) \in <_{r_1,r_2}\}
\]

since \( \mathcal{M}' \models \varphi_{\Pi}^{W-PRO}(\overline{S}) \) (which enforces a well-order on \( \Gamma(\Pi)_M \)). Moreover, since \( \mathcal{M}' \models \varphi_{\Pi,<}^W \), then it follows that the well-order is also a weak preference preserving well-order (W-PPW), which satisfies the following properties. For each \( (r, \eta) \in \Gamma(\Pi)_M \):

1. (a) \( Pos(r, \eta) \subseteq \mathcal{M}'(\Pi) \cup \{Head(r')\eta' | (r', \eta') <^W (r, \eta)\} \), or
   (b) \( Head(r, \eta) \in \{Head(r')\eta' | (r', \eta') <^W (r, \eta)\} \);
2. (a) \( (r', \eta') \in \Gamma(\Pi)_M \) implies \( (r', \eta') <^W (r, \eta) \);
   (b) \( (r', \eta') \notin \Gamma(\Pi)_M \) implies that either:

43
i. $\text{Pos}(r')\eta' \not\subseteq \mathcal{M}$ or

ii. $\text{Neg}(r')\eta' \cap (\mathcal{M}^0(\Pi) \cup \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^W (r, \eta)\}) \neq \emptyset$ or

iii. $\text{Head}(r'')\eta'' \in \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^W (r, \eta)\}$.

Now, for an $\alpha \in \Gamma(\Pi)_M$, define $\mathcal{R}^\alpha_W(\Pi)$ inductively as follows:

$$\mathcal{R}^\text{bot}(W)_W(\Pi) := \{\text{bot}(W)\};$$

$$\mathcal{R}^{\text{succ}(\alpha)}_W(\Pi) := \mathcal{R}^\text{bot}(W)_W(\Pi) \cup \{\text{succ}(\alpha)\},$$

where $\text{bot}(W)$, $\text{succ}(\alpha)$, and $\text{ord}(W)$ denotes the bottom element under $W$, the successor element of $\alpha$ under $W$, and the order type of $W$ respectively. Intuitively, $\mathcal{R}^\alpha_W(\Pi)$ represents the “gradual” collection of the $(r, \eta)$ pairs of $\Gamma(\Pi)_M$ under the well-order $W$ up to and including $\alpha$.

Now we show by induction on $\alpha$ for $\alpha \geq \text{bot}(W)$ that $\mathcal{R}^\alpha_W(\Pi) \subseteq \Lambda^\alpha_W(\Pi)_M$.

**Basis.** Assume $\text{bot}(W) = (r, \eta)$. Then since $(r, \eta)$ is the least element in $\Gamma(\Pi)_M$ under $W$, by Condition 1 of W-PPW (i.e., the weak-preference preserving well-order), we must have $\text{Pos}(r)\eta \subseteq \mathcal{M}^0(\Pi)$. Moreover, since $(r, \eta) \in \Gamma(\Pi)_M$, then we also have $\text{Neg}(r)\eta \cap \mathcal{M} = \emptyset$. Now, since $(r, \eta)$ is the least element, then for all $(r', \eta')$ with $r' <^W r$, we must have $(r', \eta') \not\in \Gamma(\Pi)_M$ for if $(r', \eta') \in \Gamma(\Pi)_M$, then we must have $(r', \eta') <^W (r, \eta)$ by Condition 2 (a) of W-PPW which is absurd since $(r, \eta)$ is the least element. Therefore, by Condition 2 (b) of W-PPW, we have for all $r' \in \Pi$ with $r' < r$ and assignment $\eta'$ that either:

1. $\text{Pos}(r')\eta' \not\subseteq \mathcal{M}$ or
2. $\text{Neg}(r')\eta' \cap \mathcal{M}^0(\Pi) \neq \emptyset$.

Then this implies there does not exists a rule $r'' \in \Pi$ with $r'' < r$ and assignment $\eta''$ such that $\text{Pos}(r'')\eta'' \subseteq \mathcal{M}$ and $\text{Neg}(r'')\eta'' \cap \mathcal{M} = \emptyset$. This further implies $(r, \eta) \in \Lambda^0_W(\Pi)_M \subseteq \Lambda^\alpha_W(\Pi)_M$ by the definition of $\Lambda^\alpha_W(\Pi)_M$.

**Step.** Assume for bot(W) = $\beta \leq \alpha$ we have $\mathcal{R}^\beta_W(\Pi) \subseteq \Lambda^\beta_W(\Pi)_M$.

Now we show that $\mathcal{R}^{\text{succ}(\alpha)}_W(\Pi) \subseteq \Lambda^\alpha_W(\Pi)_M$. We assume that $\text{succ}(\alpha) = (r, \eta)$. Then by Condition 1 of W-PPW, we have that either:

1. $\text{Pos}(r)\eta \subseteq \mathcal{M}^0(\Pi) \cup \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^W (r, \eta)\}$ or
2. $\text{Head}(r)\eta \subseteq \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^W (r, \eta)\}$.

Additionally, since $(r, \eta) \in \Gamma(\Pi)_M$ (i.e., those involved in the well-order are only of generating rules), then we also have $\text{Neg}(r)\eta \cap \mathcal{M} = \emptyset$. Therefore, since $\mathcal{R}^\text{bot}(W)_W(\Pi) = \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^W (r, \eta)\} \subseteq \Lambda^\alpha_W(\Pi)_M$ by the inductive hypothesis, then for some $t \geq 1$, we have $\mathcal{R}^\alpha_W(\Pi) = \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^W (r, \eta)\} \subseteq \Lambda^\alpha_W(\Pi)_M$. This then implies that:

1. $\text{Pos}(r)\eta \subseteq \mathcal{M}^0(\Pi)_M$ and $\text{Neg}(r)\eta \cap \mathcal{M} = \emptyset$ or
2. $\text{Head}(r)\eta \subseteq \mathcal{M}^0(\Pi)_M$.

Now, assume for some $r'' \in \Pi$ with $r'' < r$ and assignment $\eta''$, we have:

1. $\text{Pos}(r')\eta' \subseteq \mathcal{M}$ and $\text{Neg}(r')\eta' \cap \mathcal{M}^0(\Pi)_M = \emptyset$ or
2. $\text{Head}(r')\eta' \subseteq \mathcal{M}^0(\Pi)_M$.

(i.e. that is, $(r', \eta')$ is blocking $(r, \eta)$ from being applied at stage $t + 1$ in $\Lambda^{t+1}_W(\Pi)_M$). Then we also have $(r'', \eta'') \not\in \Lambda^\alpha_W(\Pi)_M$ from (2) above, i.e., since $\text{Head}(r')\eta' \not\in \mathcal{M}^0(\Pi)_M$. Then there can only be two possibilities:

**Case 1:** $(r', \eta') \not\in \Gamma(\Pi)_M$.

Then $(r', \eta') <^W (r, \eta)$ by Condition 2 (a) of W-PPW since $(r, \eta)$ is also in $\Gamma(\Pi)_M$ and where $r' < r$. Then this implies $(r', \eta') \in \mathcal{R}^{\text{succ}(\alpha)}_W(\Pi) \subseteq \Lambda^\alpha_W(\Pi)_M$, contradicting the assumption $(r', \eta') \not\in \Lambda^\alpha_W(\Pi)_M$ above.

**Case 2:** $(r', \eta') \not\in \Gamma(\Pi)_M$.

Then since $\mathcal{R}^\alpha_W(\Pi) = \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^W (r, \eta)\} \subseteq \Lambda^\alpha_W(\Pi)_M$ (i.e., inductive hypothesis), this is a contradiction since by Condition 2 (b) of W-PPW, we have that either:

1. $\text{Pos}(r')\eta' \not\subseteq \mathcal{M}$ or
2. $\text{Neg}(r')\eta' \cap \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^W (r, \eta)\} \neq \emptyset$ or
(3) $\text{Head}(r')\eta' \in \{(r'', \eta'') \mid (r'', \eta'') \leq^W \alpha <^W (r, \eta)\}$. Therefore, since there is no such pair $(r', \eta')$ blocking $(r, \eta)$ from being applied at stage $t + 1$ in $\Lambda_+^{(t+1)}(\Pi, \mathcal{M})$, then we conclude that $(r, \eta) \in \Lambda_+^{(t+1)}(\Pi, \mathcal{M})$. Thus, we have $\mathcal{R}_{\Lambda_+^{\text{ succ}}(\alpha)}(\Pi) \subseteq \Lambda_+^{(t+1)}(\Pi, \mathcal{M}) \subseteq \Lambda_+^{\infty}(\Pi, \mathcal{M})$ as well. This ends the inductive step.

Therefore, since we have $\mathcal{R}_W^{\text{ ord}}(\Pi) \subseteq \Lambda_+^{\infty}(\Pi, \mathcal{M})$, to show $\mathcal{M} \subseteq \tau_{\mathcal{M}}(\Lambda_+^{\infty}(\Pi, \mathcal{M}))$, it is now enough to only show $\mathcal{M} \subseteq \tau_{\mathcal{M}}(\Pi) \cup \{\text{Head}(r)\eta \mid (r, \eta) \in \mathcal{R}_W^{\text{ ord}}(\Pi)\}$. Thus, let $P(\vec{a}) \in \mathcal{M}$ and where $P(\vec{a}) \notin \mathcal{M}^0(\Pi)$ (i.e. if for $P(\vec{a}) \notin \mathcal{M}^0(\Pi)$ then the result is clear). We will show $P(\vec{a}) \in \{\text{Head}(r)\eta \mid (r, \eta) \in \mathcal{R}_W^{\text{ ord}}(\Pi)\}$. Indeed, since $\mathcal{M} \models \varphi_{\Pi}^{\text{ comp}}$, then there exists some rule $r$ and corresponding assignment $\eta$ such that:

1. $\text{Head}(r)\eta = P(\vec{a})$;
2. $\text{Pos}(r)\eta \subseteq M$ and $\text{Neg}(r)\eta \cap M = \emptyset$,

which implies that $(r, \eta) \in \Gamma(\Pi, \mathcal{M})$. Then this implies that $(r, \eta) \in \mathcal{R}_W^{\text{ ord}}(\Pi)$ (i.e., since $\mathcal{R}_W^{\text{ ord}}(\Pi)$ is constructed from $\Gamma(\Pi, \mathcal{M})$). Therefore, we have $P(\vec{a}) \in \{\text{Head}(r)\eta \mid (r, \eta) \in \mathcal{R}_W^{\text{ ord}}(\Pi)\}$. Hence, we had shown that $\mathcal{M} \subseteq \tau_{\mathcal{M}}(\Lambda_+^{\infty}(\Pi, \mathcal{M}))$.

$(\Leftarrow)$ Assume $\lambda_{\mathcal{M}}(\Lambda_+^{\infty}(\Pi, \mathcal{M})) = \mathcal{M}$. We will show that there exists a weak preference preserving well-order (W-PPW) of $\Gamma(\Pi, \mathcal{M})$. That is, there exists a well-order $\mathcal{W} = (\Gamma(\Pi, \mathcal{M}), <^\mathcal{W})$ on $\Gamma(\Pi, \mathcal{M})$ that satisfies Conditions 1 and 2 of the definition of W-PPW. First we have the following claim.

Claim 2. $\Lambda_+^{\infty}(\Pi, \mathcal{M}) = \Gamma(\Pi, \mathcal{M})$, i.e., $\Lambda_+^{\infty}(\Pi, \mathcal{M})$ contains exactly all the generating rules.

Proof of Claim 2. In fact, since $\Lambda_+^{\infty}(\Pi, \mathcal{M}) \subseteq \Gamma(\Pi, \mathcal{M})$ from the definition of $\Lambda_+^{\infty}(\Pi, \mathcal{M})$ (i.e., since $\Lambda_+^{\infty}(\Pi, \mathcal{M})$ can only contain generating rules under $\mathcal{M}$), then we only need to show that $\Gamma(\Pi, \mathcal{M}) \subseteq \Lambda_+^{\infty}(\Pi, \mathcal{M})$. For the sake of contradiction, assume that there exists some $(r, \eta) \in \Gamma(\Pi, \mathcal{M})$ such that $(r, \eta) \notin \Lambda_+^{\infty}(\Pi, \mathcal{M})$. Since $(r, \eta) \in \Gamma(\Pi, \mathcal{M})$, then we have that $\text{Pos}(r)\eta \subseteq \mathcal{M}$ and $\text{Neg}(r)\eta \cap \mathcal{M} = \emptyset$ (i.e., since $(r, \eta)$ is a generating rule). Moreover, since $\lambda_{\mathcal{M}}(\Lambda_+^{\infty}(\Pi, \mathcal{M})) = \mathcal{M}$ by assumption, then for some $t \geq 0$, we have that $\text{Pos}(r)\eta \subseteq \lambda_{\mathcal{M}}(\Lambda_+^t(\Pi, \mathcal{M}))$. Then since $(r, \eta) \notin \Lambda_+^{(t+1)}(\Pi, \mathcal{M})$ (i.e., since $(r, \eta) \notin \Lambda_+^{\infty}(\Pi, \mathcal{M})$), we have that there exists some rule $r' < r$ and corresponding assignment $\eta'$ such that:

1. $\text{Pos}(r')\eta' \subseteq \mathcal{M}$ and $\text{Neg}(r')\eta' \cap \lambda_{\mathcal{M}}(\Lambda_+^t(\Pi, \mathcal{M})) = \emptyset$;
2. $\text{Head}(r')\eta' \notin \lambda_{\mathcal{M}}(\Lambda_+^{(t+1)}(\Pi, \mathcal{M}))$ (i.e., the rule that is blocking $(r, \eta)$ from being applied at stage $t + 1$). Now set $B^k(r, \eta) := \{(r'', \eta'') \mid (a) r'' < r;
(b) \text{Pos}(r'')\eta'' \subseteq \mathcal{M}$ and $\text{Neg}(r'')\eta'' \cap \lambda_{\mathcal{M}}(\Lambda_+^k(\Pi, \mathcal{M})) = \emptyset$;
(c) $\text{Head}(r'')\eta'' \in \lambda_{\mathcal{M}}(\Lambda_+^k(\Pi, \mathcal{M}))\}$,

to be the set of (more preferred) rules $(r'', \eta'')$ blocking $(r, \eta)$ from being applied at stage $k \geq 1$ as in the proof of Proposition 6.4. Then through the assumption that $\lambda_{\mathcal{M}}(\Lambda_+^k(\Pi, \mathcal{M})) = \mathcal{M}$, and together with the fact that $(r', \eta') \in \Lambda_+^k(\Pi, \mathcal{M})$ implies $\text{Head}(r')\eta' \in \lambda_{\mathcal{M}}(\Lambda_+^k(\Pi, \mathcal{M}))$, we can use similar techniques to that used in the proof of Proposition 3.5 to show that there exists some ordinal $K \geq k + 1$ for which $B^K(r, \eta) = \emptyset$. Then this implies that $(r, \eta) \in \Lambda_+^{K+1}(\Pi, \mathcal{M}) \subseteq \Lambda_+^{\infty}(\Pi, \mathcal{M})$ (i.e., since there are no more rules $(r', \eta')$ blocking $(r, \eta)$ from being applied), which is a contradiction. This completes the proof of Claim 2.

Hence, similarly to the proof of Theorem 5.7, we show by induction on $t \geq 0$ that there exists a well-order $\mathcal{W} = (\Lambda_+^t(\Pi, \mathcal{M}), <^\mathcal{W})$ on $\Lambda_+^t(\Pi, \mathcal{M})$ that satisfies the weak preference preserving well-order (W-PPW) as first defined above. That is, for each $(r, \eta) \in \Lambda_+^t(\Pi, \mathcal{M})$, we have that:
(1) (a) \( \text{Pos}(r) \eta \subseteq M^0(\Pi) \cup \{ \text{Head}(r') \eta' \mid (r', \eta') <^W (r, \eta) \} \), or  
(b) \( \text{Head}(r) \eta \in \{ \text{Head}(r') \eta' \mid (r', \eta') <^W (r, \eta) \} \);  
(2) for each rule \( r' < r \) and assignment \( \eta' \):  
(a) \( (r', \eta') \in \Lambda^W(\Pi)_M \) implies \( (r', \eta') <^W (r, \eta) \);  
(b) \( (r', \eta') \notin \Gamma(\Pi)_M \) implies that either:  
   i. \( \text{Pos}(r') \eta' \notin \mathcal{M} \) or  
   ii. \( \text{Neg}(r') \eta' \cap \{ M^0(\Pi) \cup \{ \text{Head}(r') \eta' \mid (r', \eta') <^W (r, \eta) \} \} \) \( \neq \emptyset \) or  
   iii. \( \text{Head}(r') \eta' \in \{ \text{Head}(r'') \eta'' \mid (r'', \eta'') <^W (r, \eta) \} \).  

Thus ultimately, since \( \Lambda^\infty_W(\Pi)_M = \Gamma(\Pi)_M \) from Claim 2 above, then we also have that there exists a W-PPW of \( \Gamma(\Pi)_M \).

**Basis.** We have by the order-extension principle (OE) [Kaye and Macpherson 1994] that there exists a well-order \( W = (\Lambda^0_W(\Pi)_M, <^W) \) on \( \Lambda^0_W(\Pi)_M \). Moreover, by the finiteness of \( \Pi \), we can also make \( W \) in such a way that the priorities among the rules are respected (i.e., same argument as that used in the proof of Theorem 5.7). We now make the following claims:

**Claim 3.** \( (r, \eta) \in \Lambda^0_W(\Pi)_M \) implies:  
1. \( \text{Pos}(r) \eta \subseteq M^0(\Pi) \cup \{ \text{Head}(r') \eta' \mid (r', \eta') <^W (r, \eta) \} \) or  
2. \( \text{Head}(r) \eta \in \{ \text{Head}(r') \eta' \mid (r', \eta') <^W (r, \eta) \} \).

**Proof of Claim 3.** Indeed, let \( (r, \eta) \in \Lambda^0_W(\Pi)_M \). Then by the definition of \( \Lambda^0_W(\Pi)_M \), we have \( \text{Pos}(r) \eta \subseteq M^0(\Pi) \). This completes the proof of Claim 3.

**Claim 4.** For \( (r, \eta) \), \( (r', \eta') \in \Lambda^0_W(\Pi)_M \) where \( r < r' \), we have that \( (r, \eta) <^W (r', \eta') \).

**Proof of Claim 4.** This follows from the description of \( W \). This ends the proof of Claim 4.

**Claim 5.** For \( (r, \eta) \notin \Gamma(\Pi)_M \) and \( (r', \eta') \in \Lambda^0_W(\Pi)_M \) where \( r < r' \), we have that either:  
1. \( \text{Pos}(r) \eta \subseteq \mathcal{M} \) or  
2. \( \text{Neg}(r) \eta \cap \mathcal{M}^0(\Pi) = \emptyset \).

Then by the definition of \( \Lambda^0_W(\Pi)_M \), this contradicts the assumption \( (r', \eta') \in \Lambda^0_W(\Pi)_M \) since \( (r, \eta) \) is blocking \( (r', \eta') \) from being applied at stage \( 0 \). This completes the proof of Claim 5.

**Step.** Assume for \( 1 \leq t' \leq t \) the hypothesis holds.

We will now show that it also holds for \( t + 1 \). Indeed, by the inductive hypothesis, there exists a well-order \( W = (\Lambda^t_W(\Pi)_M, <^W) \) on \( \Lambda^t_W(\Pi)_M \) satisfying Conditions 1 and 2 of W-PPW. Now set

\[ \Delta^{t+1}_W(\Pi)_M = \Lambda^{t+1}_W(\Pi)_M \setminus \Lambda^t_W(\Pi)_M. \]

Then again by the well-ordering theorem [Enderton 1977], there exists a well-order \( W' = (\Delta^{t+1}_W(\Pi)_M, <^W) \) on \( \Delta^{t+1}_W(\Pi)_M \). Moreover, by the finiteness of \( \Pi \), we can also make \( W' \) in such a way that the priorities among the rules are respected (i.e., same argument as proposed in the proof of Theorem 5.7, which simply follows from the fact that the sum and products of well-ordered types is also well-ordered [Enderton 1977]). Then we now set \( W'' = (\Lambda^{t+1}_W(\Pi)_M, <^W) \) to be a well-order on \( \Lambda^{t+1}_W(\Pi)_M \) by setting \(<^W := <^W \cup <^W \cup (\Lambda^t_W(\Pi)_M \times \Delta^{t+1}_W(\Pi)_M)).

We now make the following claims:

**Claim 6.** \( (r, \eta) \in \Lambda^{t+1}_W(\Pi)_M \) implies:

---

46
Claim 7. Then by the inductive hypothesis, we have:

Case 1. $\text{Claim 6.}$  Then there can be two possibilities:

**Case 1.** $(r, \eta) \in \Lambda^{t+1}_{W}(\Pi, M)$.  Then by the inductive hypothesis, we have:

1. $\text{Pos}(r)\eta \subseteq M \cap \{\text{Head}(r')\eta' : (r', \eta') <^{W''}(r, \eta)\}$
2. $\text{Head}(r)\eta \in \{\text{Head}(r')\eta' : (r', \eta') <^{W''}(r, \eta)\}$.

**Proof of Claim 6.** Then there can be two possibilities:

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

Then by the inductive hypothesis, we have:

1. $\text{Pos}(r)\eta \subseteq M \cap \{\text{Head}(r')\eta' : (r', \eta') <^{W''}(r, \eta)\}$
2. $\text{Head}(r)\eta \in \{\text{Head}(r')\eta' : (r', \eta') <^{W''}(r, \eta)\}$.

Then since $<^{W''} \subseteq <^{W'''}$, we also have:

1. $\text{Pos}(r)\eta \subseteq M \cap \{\text{Head}(r')\eta' : (r', \eta') <^{W''}(r, \eta)\}$
2. $\text{Head}(r)\eta \in \{\text{Head}(r')\eta' : (r', \eta') <^{W''}(r, \eta)\}$.

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

Then by the definition of $\Lambda^{t+1}_{W}(\Pi, M)$ and since $(r, \eta) \notin \Lambda^{t}_{W}(\Pi, M)$ (i.e. which implies $(r, \eta)$ is derived at stage $t + 1$), then we must have either:

1. $\text{Pos}(r)\eta \subseteq \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M))$ and $\text{Neg}(r)\eta \cap M = \emptyset$
2. $(r, \eta) \in \Gamma(\Pi, M)$ and $\text{Head}(r)\eta \in \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M))$.

Since by the construction $\Lambda^{t+1}_{W}(\Pi, M) \times \Delta^{t+1}_{W}(\Pi, M)$ in $<^{W''}$ we have $(r', \eta') <^{W''}(r, \eta)$ for all $(r', \eta') \in \Lambda^{t+1}_{W}(\Pi, M)$, then this implies:

1. $\text{Pos}(r)\eta \subseteq M \cap \{\text{Head}(r')\eta' : (r', \eta') <^{W''}(r, \eta)\}$
2. $\text{Head}(r)\eta \in \{\text{Head}(r')\eta' : (r', \eta') <^{W''}(r, \eta)\}$.

This ends the proof of Claim 6.

**Claim 7.** For $(r, \eta), (r', \eta') \in \Lambda^{t+1}_{W}(\Pi, M)$ where $r < r'$, we have that $(r, \eta) <^{W''} (r', \eta')$.

**Proof of Claim 7.** There can only be three possibilities:

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

Then by the inductive hypothesis, we have $r < r'$ implies $(r, \eta) <^{W} (r', \eta')$. Then since $(r, \eta) <^{W} (r', \eta')$ implies $(r, \eta) <^{W''} (r', \eta')$ (i.e. since $<^{W} \subseteq <^{W''}$), we also have $r < r'$ implies $(r, \eta) <^{W''} (r', \eta')$.

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

Then by the construction $\Lambda^{t}_{W}(\Pi, M) \times \Delta^{t+1}_{W}(\Pi, M)$ in $<^{W''}$, we have $(r, \eta) <^{W''} (r', \eta')$.

For the sake of contradiction, assume $r' < r$. Then since $(r', \eta') \in \Delta^{t+1}_{W}(\Pi, M) = \Lambda^{t+1}_{W}(\Pi, M) \setminus \Lambda^{t}_{W}(\Pi, M)$ (i.e. $(r', \eta')$ is derived at stage $t + 1$), we have $\text{Pos}(r)\eta \subseteq \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M))$ and $\text{Neg}(r)\eta \cap M = \emptyset$. Then since $\lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M)) = M$ by assumption, we also have $\text{Pos}(r)\eta \subseteq M$ and $\text{Neg}(r)\eta \cap \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M)) = \emptyset$. Then there can be two possibilities:

**Subcase 1.** $\text{Head}(r')\eta' \notin \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M))$.

Then we have $(r', \eta')$ is a pair such that:

1. $r' < r$;
2. $\text{Pos}(r')\eta' \subseteq M$ and $\text{Neg}(r')\eta' \cap \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M)) = \emptyset$;
3. $\text{Head}(r')\eta' \notin \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M))$.

for all $1 \leq t' \leq t$. Then this contradicts the assumption $(r, \eta) \in \Lambda^{t}_{W}(\Pi, M)$ since by the definition of $\Lambda^{t}_{W}(\Pi, M)$, we have $(r', \eta')$ is blocking $(r, \eta)$ from being applied for all stages $1 \leq t' \leq t$ in $\Lambda^{t+1}_{W}(\Pi, M)$.

**Subcase 2.** $\text{Head}(r')\eta' \in \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M))$.

First, we assume that there does not exist a pair $(r'', \eta'')$ such that:

1. $(r'') < r'$;
2. $\text{Pos}(r'')\eta'' \subseteq M$ and $\text{Neg}(r'')\eta'' \cap \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M)) = \emptyset$; and
3. $\text{Head}(r'')\eta'' \notin \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M))$ (i.e., $(r'', \eta'')$ is blocking $(r', \eta')$ from application for all stages $t''$ where $1 \leq t'' \leq t$).

for this case, then we also have $r'' < r$ by transitivity (i.e., since we have $r' < r$ by assumption), which then contradicts the assumption $(r, \eta) \in \Lambda^{t}_{W}(\Pi, M)$ since $(r'', \eta'')$ will also be blocking $(r, \eta)$ from application for all stages $t''$ where $1 \leq t'' \leq t$. Thus now, since $\text{Head}(r')\eta' \in \lambda M_{\Pi}(\Lambda^{t}_{W}(\Pi, M))$. 47
then there exists the least $t' \leq t$ such that $\text{Head}(r')\eta' \in \lambda_{M^o}(\Lambda_{W'}^{t+1}(\Pi_{M}))$. Then since $(r', \eta') \in \Gamma(\Pi_{M})$ (i.e., since $(r', \eta') \in \Delta_{W'}^{t+1}(\Pi_{M})$ and where $\Delta_{W'}^{t+1}(\Pi_{M}) \subseteq \Gamma_{W'}^{t+1}(\Pi_{M}) = \Gamma(\Pi_{M})$ by Claim 2) and there is no pair $(r'', \eta'')$ blocking $(r', \eta')$ from application, then we have by the definition of $\Lambda_{W'}^{t+1}(\Pi_{M})$ (i.e., Condition 1 (b) of the definition of $\Lambda_{W'}^{t+1}(\Pi_{M})$). Then since $(r', \eta') \in \Delta_{W'}^{t+1}(\Pi_{M})$, we must have $t' + 1 = t + 1$ (i.e., that is, $t' = t$). Then since $t'$ is the least $t'$ for which we have $\text{Head}(r')\eta' \in \lambda_{M^o}(\Lambda_{W'}^{t}(\Pi_{M}))$, then $\text{Head}(r')\eta' \not\in \lambda_{M^o}(\Lambda_{W'}^{t-1}(\Pi_{M}))$. Then this implies $(r', \eta')$ is a pair such that:

1. $r' < r$;
2. $\text{Pos}(r')\eta' \subseteq M$ and $\text{Neg}(r')\eta' \cap \lambda_{M^o}(\Lambda_{W'}^{t}(\Pi_{M})) = \emptyset$;
3. $\text{Head}(r')\eta' \not\in \lambda_{M^o}(\Lambda_{W'}^{t}(\Pi_{M}))$;

for all $1 \leq t'' < t$. Then this contradicts the assumption $(r, \eta) \in \Lambda_{W'}^{t}(\Pi_{M})$ since $(r', \eta')$ will be blocking $(r, \eta)$ from application for all $1 \leq t'' < t$ in $\Lambda_{W'}^{t}(\Pi_{M})$.

Case 3. $(r, \eta), (r', \eta') \in \Delta_{W'}^{t+1}(\Pi_{M})$.

Follows from the description of the well-order $W'$ on $\Delta_{W'}^{t+1}(\Pi_{M})$.

This completes the proof of Claim 7.

Claim 8. If $(r, \eta) \not\in \Gamma(\Pi_{M})$ and $(r', \eta') \in \Lambda_{W'}^{t+1}(\Pi_{M})$ where $r < r'$ then:

1. $\text{Pos}(r)\eta \not\subseteq M$ or
2. $\text{Neg}(r)\eta \cap (\mathcal{M}^{0}(\Pi) \cup \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^{W'} (r', \eta')\}) \not= \emptyset$ or
3. $\text{Head}(r)\eta \not\in \lambda_{M}(\Lambda_{W'}^{0}(\Pi_{M}))$.

Proof of Claim 8. There can be two possibilities:

Case 1. $(r', \eta') \in \Delta_{W'}^{t}(\Pi_{M})$.

Then by the inductive hypothesis, we have:

1. $\text{Pos}(r')\eta \not\subseteq M$ or
2. $\text{Neg}(r')\eta \cap (\mathcal{M}^{0}(\Pi) \cup \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^{W'} (r', \eta')\}) \not= \emptyset$ or
3. $\text{Head}(r')\eta \not\in \lambda_{M}(\Lambda_{W'}^{0}(\Pi_{M}))$.

Then since $<_W <^{W'}$, we also have $\text{Neg}(r)\eta \cap (\mathcal{M}^{0}(\Pi) \cup \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^{W'} (r', \eta')\}) \not= \emptyset$ or $\text{Head}(r)\eta \not\in \lambda_{M}(\Lambda_{W'}^{0}(\Pi_{M}))$. Then by the construction $\Lambda_{W'}^{t}(\Pi_{M}) \times \Delta_{W'}^{t+1}(\Pi_{M})$ in $<_W$ and where $(r', \eta') \in \Delta_{W'}^{t+1}(\Pi_{M})$, we also have $\text{Neg}(r)\eta \cap \lambda_{M}(\Lambda_{W'}^{0}(\Pi_{M})) = \emptyset$ and $\text{Head}(r)\eta \not\in \lambda_{M}(\Lambda_{W'}^{0}(\Pi_{M}))$.

Case 2. For the sake of contradiction, assume $\text{Pos}(r)\eta \subseteq M$, $\text{Neg}(r)\eta \cap (\mathcal{M}^{0}(\Pi) \cup \{\text{Head}(r'')\eta'' \mid (r'', \eta'') <^{W'} (r', \eta')\}) = \emptyset$, and $\text{Head}(r)\eta \not\in \lambda_{M}(\Lambda_{W'}^{0}(\Pi_{M}))$. Then $(r, \eta)$ is a pair such that:

1. $r < r'$;
2. $\text{Pos}(r)\eta \subseteq M$ and $\text{Neg}(r)\eta \cap \lambda_{M}(\Lambda_{W'}^{0}(\Pi_{M})) = \emptyset$;
3. $\text{Head}(r)\eta \not\in \lambda_{M}(\Lambda_{W'}^{0}(\Pi_{M}))$.

Then this contradicts the assumption $(r', \eta') \in \Delta_{W'}^{t+1}(\Pi_{M}) = \Lambda_{W'}^{t+1}(\Pi_{M}) \setminus \Lambda_{W'}^{t}(\Pi_{M})$ since $(r, \eta)$ is blocking $(r', \eta')$ from being applied in $\Lambda_{W'}^{t+1}(\Pi_{M})$ at stage $t + 1$.

This ends the proof of Claim 8.

Hence, by Claims 6, 7, and 8 above, we have the hypothesis also holds for $t + 1$. This completes our inductive proof.

Therefore, given that $\Gamma(\Pi_{M})$ can be well-ordered into W-PPW, let $\mathcal{W} = (\Gamma(\Pi_{M}), <^{W})$ be such a W-PPW of $\Gamma(\Pi_{M})$. Then we now construct an expansion $\mathcal{M}'$ of $\mathcal{M}$ of the signature $\{<_{r_{1},r_{2}}\}$
\[ r_1, r_2 \in \Pi \cup \tau(\Pi) \text{ (i.e., where } \{<r_1 r_2| r_1, r_2 \in \Pi \} \text{ is the "expansion part") by setting:} \]
\[
<_{r_1 r_2}^M = \{ \langle \eta_1(u_1), \ldots, \eta_1(u_k), \eta_2(v_1), \ldots, \eta_2(v_l) \rangle | \langle u_1, \ldots, u_k \rangle = \overrightarrow{x}_{r_1}, \langle v_1, \ldots, v_l \rangle = \overrightarrow{x}_{r_2}, \text{ and } (r_1, \eta_1) <^W (r_2, \eta_2) \}
\]

for each \( r_1, r_2 \in \Pi \) (they possibly are the same). Then it can be shown (albeit tedious) that \( M' \models \forall S (\varphi_{(\Pi, <)} \wedge \varphi_{\Pi}^{\text{PRO}} (\overrightarrow{\sim}, S) \wedge \varphi_{\Pi}^{\text{COMP}}) \). This completes the proof of Theorem 6.6. \( \square \)