A semantic framework for preference handling in answer set programming

Torsten Schaub∗
Institut für Informatik, Universität Potsdam
Postfach 60 15 53, D–14415 Potsdam, Germany
torsten@cs.uni-potsdam.de

Kewen Wang†
School of Computing and Information Technology
Griffith University, Brisbane 4111, Australia
K.Wang@cit.gu.edu.au

Abstract

We provide a semantic framework for preference handling in answer set programming. To this end, we introduce preference preserving consequence operators. The resulting fixpoint characterizations provide us with a uniform semantic framework for characterizing preference handling in existing approaches. Although our approach is extensible to other semantics by means of an alternating fixpoint theory, we focus here on the elaboration of preferences under answer set semantics. Alternatively, we show how these approaches can be characterized by the concept of order preservation. These uniform semantic characterizations provide us with new insights about interrelationships and moreover about ways of implementation.

1 Introduction

Preferences constitute a very natural and effective way of resolving indeterminate situations. For example, in scheduling not all deadlines may be simultaneously satisfiable, and in configuration various goals may not be simultaneously met. In legal reasoning, laws may apply in different situations, but laws may also conflict with each other. In fact, while logical preference handling constitutes already an indispensable means for legal reasoning systems (cf. (Gordon 1993; Prakken 1997)), it is also advancing in other application areas such as intelligent agents and e-commerce (Grosof 1999) and the resolution of grammatical ambiguities (Cui and Swift 2002). The growing interest in preferences is also reflected by the large number of proposals in logic programming (Sakama and Inoue 1996; Brewka 1996; Gelfond and Son 1997; Zhang and Foo 1997; Grosof 1997; Brewka and Eiter 1999; Delgrande et al. 2000b; Wang et al. 2000). A common approach is to employ meta-formalisms for characterizing "preferred answer sets". This has led to a diversity of approaches that are hardly comparable due to considerably different ways of formal characterization. Hence, there is no homogeneous account of preference.

∗ Affiliated with the School of Computing Science at Simon Fraser University, Burnaby, Canada.
† This work was done while the second author was with the University of Potsdam.

We address this shortcoming by proposing a uniform semantical framework for extended logic programming with preferences. To be precise, we develop an (alternating) fixpoint theory for so-called *ordered logic programs* (also, prioritized logic programs). An ordered logic program is an extended logic program whose rules are subject to a strict partial order. In analogy to standard logic programming, such a program is then interpreted by means of an associated fixpoint operator. We start by elaborating upon a specific approach to preference handling that avoids some problems of related approaches. We also show how the approaches of Brewka and Eiter (2000) and Delgrande et al. (2000b) can be captured within our framework. As a result, we obtain that the investigated approaches yield an increasing number of answer sets depending on how "tight" they integrate preferences. For obtaining a complementary perspective, we also provide characterizations in terms of the property of order preservation, originally defined in (Delgrande et al. 2000b) for distinguishing "preferred" from "non-preferred" answer sets. Moreover, we show how these approaches can be implemented by the compilation techniques developed in (Delgrande et al. 2000b). As well, we show that all these different preferred answer set semantics correspond to the perfect model semantics on stratified programs. We deal with approaches whose preferred answer sets semantics amounts to a selection function on the standard answer sets of an ordered logic program. In view of our interest in compiling these approaches into ordinary logic programs, we moreover limit our investigation to those guaranteeing polynomial translations. This excludes approach like the ones in (Rintanen 1995; Zhang and Foo 1997) that step outside the complexity class of the underlying logic programming framework. This applies also to the approach in (Sakama and Inoue 1996), where preferences on literals are investigated. While the approach of (Gelfond and Son 1997) remains within NP, it advocates strategies that are non-selective (as discussed in Section 5). Approaches that can be addressed within this framework include those in (Baader and Hollunder 1993; Brewka 1994) that were originally proposed for default logic.

The paper is organized as follows. Once Section 2 has provided formal preliminaries, we begin in Section 3 by elaborating upon our initial semantics for ordered logic programs. Afterwards, we show in Section 4 how this semantics has to be modified in order to account for the two other aforementioned approaches.

2 Definitions and notation

We assume a basic familiarity with alternative semantics of logic programming (Lifschitz 1996). An *extended logic program* is a finite set of rules of the form

$$L_0 \leftarrow L_1, \dots, L_m, not\ L_{m+1}, \dots, not\ L_n, \tag{1}$$

where $n \geq m \geq 0$, and each L_i $(0 \leq i \leq n)$ is a *literal*, ie. either an atom A or the negation $\neg A$ of A. The set of all literals is denoted by Lit. Given a rule r as in (1), we let $head(r)$ denote the *head*, L_0, of r and $body(r)$ the *body*, $\{L_1, \dots, L_m,\ not\ L_{m+1}, \dots, not\ L_n\}$, of r. Further, let $body^+(r) = \{L_1, , \dots, L_m\}$ and $body^-(r) = \{L_{m+1}, \dots, L_n\}$. A program is called *basic* if $body^-(r) = \emptyset$ for all its rules; it is called *normal* if it contains no classical negation symbol $\neg$. The reduct of a rule r is defined as $r^+ = head(r) \leftarrow body^+(r)$; the *reduct*, Π^X, of a program Π *relative to* a set X of literals is defined by

$$\Pi^X = \{r^+ \mid r \in \Pi \text{ and } body^-(r) \cap X = \emptyset\}.$$

A set of literals X is *closed under* a basic program Π iff for any $r \in \Pi$, $head(r) \in X$ whenever $body^+(r) \subseteq X$. We say that X is *logically closed* iff it is either consistent (ie. it does not contain both a literal A and its negation $\neg A$) or equals Lit. The smallest set of literals which is both logically closed and closed under a basic program Π is denoted by $Cn(\Pi)$. With these formalities at hand, we can define *answer set semantics* for extended logic programs: A set X of literals is an *answer set* of a program Π iff $Cn(\Pi^X) = X$. For the rest of this paper, we concentrate on *consistent* answer sets. For capturing other semantics, $Cn(\Pi^X)$ is sometimes regarded as an operator $C_\Pi(X)$. The anti-monotonicity of C_Π implies that C^2_Π is monotonic. As shown in (van Gelder 1993), different semantics are obtained by distinguishing different groups of (alternating) fixpoints of $C^2_\Pi(X)$.

Alternative inductive characterizations for the operators Cn and C_Π can be obtained by appeal to *immediate consequence operators* (Lloyd 1987). Let Π be a basic program and X a set of literals. The *immediate consequence operator* T_Π is defined as follows:

$$T_\Pi X = \{head(r) \mid r \in \Pi \text{ and } body(r) \subseteq X\}$$

if X is consistent, and $T_\Pi X = Lit$ otherwise. Iterated applications of T_Π are written as T^j_Π for $j \geq 0$, where $T^0_\Pi X = X$ and $T^i_\Pi X = T_\Pi T^{i-1}_\Pi X$ for $i \geq 1$. It is well-known that $Cn(\Pi) = \bigcup_{i \geq 0} T^i_\Pi \emptyset$, for any basic program Π. Also, for any answer set X of program Π, it holds that $X = \bigcup_{i \geq 0} T^i_{\Pi^X} \emptyset$. A reduction from extended to basic programs is avoidable with an extended operator: Let Π be an extended program and X and Y be sets of literals. The *extended immediate consequence operator* $T_{\Pi,Y}$ is defined as follows:

$$T_{\Pi,Y} X = \{head(r) \mid r \in \Pi, body^+(r) \subseteq X, \text{ and } body^-(r) \cap Y = \emptyset\} \tag{2}$$

if X is consistent, and $T_{\Pi,Y} X = Lit$ otherwise. Iterated applications of $T_{\Pi,Y}$ are written as those of T_Π. Clearly, we have $T_{\Pi,\emptyset} X = T_\Pi X$ for any basic program Π and $T_{\Pi,Y} X = T_{\Pi^Y} X$ for any extended program Π. Accordingly, we have for any answer set X of program Π that $X = \bigcup_{i \geq 0} T^i_{\Pi,X} \emptyset$. Finally, for dealing with the individual rules in (2), we rely on the notion of *activeness*: Let $X, Y \subseteq Lit$ be two sets of literals in a program Π. A rule r in Π is *active* wrt the pair (X, Y), if $body^+(r) \subseteq X$ and $body^-(r) \cap Y = \emptyset$. Alternatively, we thus have that $T_{\Pi,Y} X = \{head(r) \mid r \in \Pi \text{ is active wrt } (X, Y)\}$.

Lastly, an *ordered logic program* is simply a pair $(\Pi, <)$, where Π is an extended logic program and $< \subseteq \Pi \times \Pi$ is an irreflexive and transitive relation. Given, $r_1, r_2 \in \Pi$, the relation $r_1 < r_2$ is meant to express that r_2 has *higher priority* than r_1. Programs associated with such an external ordering are also referred to as *statically* ordered programs, as opposed to *dynamically* ordered programs whose order relation is expressed through a special-purpose predicate within the program.

3 Preferred fixpoints

We elaborate upon a semantics for ordered logic program that allows us to distinguish the "preferred" answer sets of a program $(\Pi, <)$ by means of fixpoint equations. That is, a set of literals X is a preferred answer set of $(\Pi, <)$, if it satisfies the equation $\mathcal{C}_{(\Pi,<)}(X) = X$ for some operator $\mathcal{C}_{(\Pi,<)}$. In view of the classical approach described above, this makes us investigate semantics that interpret preferences as inducing selection functions on the set of standard answer sets of the underlying non-ordered program Π.

Answer sets are defined via a reduction of extended logic programs to basic programs. Controlling such a reduction by means of preferences is difficult since all conflicts are simultaneously resolved when turning Π into Π^X. Furthermore, we argue that conflict resolution must be addressed among the original rules in order to account for blockage between rules. In fact, once the negative body $body^-(r)$ is eliminated there is no way to detect whether $head(r') \in body^-(r)$ holds in case of $r < r'$. Our idea is thus to characterize preferred answer sets by an inductive development that agrees with the given ordering. In terms of a standard answer set X, this means that we favor its formal characterization as $X = \bigcup_{i\geq 0} T^i_{\Pi,X}\emptyset$ over $X = Cn(\Pi^X)$. This leads us to the following definition.

Definition 1

Let $(\Pi, <)$ be an ordered logic program and let X and Y be sets of literals.

We define the set of immediate consequences of X with respect to $(\Pi, <)$ and Y as

$$\mathcal{T}_{(\Pi,<),Y}X \quad = \quad \left\{ head(r) \;\middle|\; \begin{array}{rl} I. & r \in \Pi \text{ is active wrt } (X,Y) \text{ and} \\ II. & \text{there is no rule } r' \in \Pi \text{ with } r < r' \\ & \text{such that} \\ & (a)\ r' \text{ is active wrt } (Y,X) \text{ and} \\ & (b)\ head(r') \notin X \end{array} \right\}$$

if X is consistent, and $\mathcal{T}_{(\Pi,<),Y}X = Lit$ otherwise.

Note that $\mathcal{T}_{(\Pi,<),Y}$ is a refinement of its classical counterpart $T_{\Pi,Y}$ in (2). The idea behind Condition *II* is to apply a rule r only if the "question of applicability" has been settled for all higher-ranked rules r'. Let us illustrate this in terms of iterated applications of $\mathcal{T}_{(\Pi,<),Y}$. In this case, X accumulates conclusions, while Y comprises the putative answer set. Then, the "question of applicability" is considered to be settled for a higher ranked rule r'

- if the prerequisites of r' will never be derivable, viz. $body^+(r') \not\subseteq Y$, or
- if r' is defeated by what has been derived so far, viz. $body^-(r) \cap X \neq \emptyset$, or
- if r' or another rule with the same head have already applied, viz. $head(r') \in X$.

The first two conditions show why activeness of r' is stipulated wrt (Y, X), as opposed to (X, Y) in Condition *I*. The last condition serves two purposes: First, it detects whether the higher ranked rule r' has applied and, second, it suspends the preference $r < r'$ whenever the head of the higher ranked has already been derived by another rule. This suspension of preference constitutes a distinguishing feature of the approach at hand.

As with $T_{\Pi,Y}$, iterated applications of $\mathcal{T}_{(\Pi,<),Y}$ are written as $\mathcal{T}^j_{(\Pi,<),Y}$ for $j \geq 0$, where $\mathcal{T}^0_{(\Pi,<),Y}X = X$ and $\mathcal{T}^i_{(\Pi,<),Y}X = \mathcal{T}_{(\Pi,<),Y}\mathcal{T}^{i-1}_{(\Pi,<),Y}X$ for $i \geq 1$. The counterpart of operator C_Π for ordered programs is then defined as follows.

Definition 2

Let $(\Pi, <)$ be an ordered logic program and let X be a set of literals.

We define $\mathcal{C}_{(\Pi,<)}(X) = \bigcup_{i\geq 0} \mathcal{T}^i_{(\Pi,<),X}\emptyset$.

Clearly, $\mathcal{C}_{(\Pi,<)}$ is a refinement of C_Π. The difference is that $\mathcal{C}_{(\Pi,<)}$ obtains consequences directly from Π and Y, while C_Π (normally) draws them by appeal to Cn after reducing Π to Π^Y. All this allows us to define preferred answer sets as fixpoints of $\mathcal{C}_{(\Pi,<)}$.

Definition 3

Let $(\Pi, <)$ be an ordered logic program and let X be a set of literals.

We define X as a preferred answer set of $(\Pi, <)$ iff $\mathcal{C}_{(\Pi,<)}(X) = X$.

For illustration, consider the following ordered logic program $(\Pi_3, <)$:

$$\begin{array}{lrcl@{\qquad}lrcl@{\qquad}l}
r_1 : & \neg f & \leftarrow & p, not\ f & r_4 : & b & \leftarrow & p & r_2 < r_1 \\
r_2 : & w & \leftarrow & b, not\ \neg w & r_5 : & p & \leftarrow & & \\
r_3 : & f & \leftarrow & w, not\ \neg f & & & & &
\end{array} \tag{3}$$

Observe that Π_3 admits two answer sets: $X = \{p, b, \neg f, w\}$ and $X' = \{p, b, f, w\}$. As argued in (Baader and Hollunder 1993), X is preferred to X'. To see this, observe that

$$\begin{array}{lcl@{\qquad}lcl}
\mathcal{T}^0_{(\Pi_3,<),X}\emptyset & = & \emptyset & \mathcal{T}^0_{(\Pi_3,<),X'}\emptyset & = & \emptyset \\
\mathcal{T}^1_{(\Pi_3,<),X}\emptyset & = & \{p\} & \mathcal{T}^1_{(\Pi_3,<),X'}\emptyset & = & \{p\} \\
\mathcal{T}^2_{(\Pi_3,<),X}\emptyset & = & \{p, b, \neg f\} & \mathcal{T}^2_{(\Pi_3,<),X'}\emptyset & = & \{p, b\} \\
\mathcal{T}^3_{(\Pi_3,<),X}\emptyset & = & \{p, b, \neg f, w\} & \mathcal{T}^3_{(\Pi_3,<),X'}\emptyset & = & \mathcal{T}^2_{(\Pi_3,<),X'}\emptyset \\
\mathcal{T}^4_{(\Pi_3,<),X}\emptyset & = & \mathcal{T}^3_{(\Pi_3,<),X}\emptyset = X & & \neq & X'
\end{array} \tag{4}$$

We thus get $\mathcal{C}_{(\Pi_3,<)}(X) = X$, while $\mathcal{C}_{(\Pi_3,<)}(X') = \{p, b\} \neq X'$. Note that w cannot be included into $\mathcal{T}^3_{(\Pi_3,<),X'}\emptyset$ since r_1 is active wrt $(X', \mathcal{T}^2_{(\Pi_3,<),X'}\emptyset)$ and r_1 is preferred to r_2.

It is important to see that preferences may sometimes be too strong and deny the existence of preferred answer sets although standard ones exist. This is because preferences impose additional dependencies among rules that must be respected by the resulting answer sets. This is nicely illustrated by programs $\Pi_5 = \{r_1, r_2\}$ and $\Pi'_5 = \{r'_1, r'_2\}$, respectively:

$$\begin{array}{lcccl@{\qquad}lcccl}
r_1 & = & a & \leftarrow & b & r'_1 & = & a & \leftarrow & not\ b \\
r_2 & = & b & \leftarrow & & r'_2 & = & b & \leftarrow &
\end{array} \tag{5}$$

Observe that in Π_5 rule r_1 depends r_2, while in Π'_5 rule r'_1 is defeated by r'_2. But despite the fact that Π_5 has answer set $X = \{a, b\}$ and Π'_5 has answer set $X' = \{b\}$, we obtain no preferred answer set after imposing preferences $r_2 < r_1$ and $r'_2 <' r'_1$, respectively. To see this, observe that $\mathcal{T}^0_{(\Pi_5,<),X}\emptyset = \mathcal{T}^1_{(\Pi_5,<),X}\emptyset = \emptyset \neq X$ and $\mathcal{T}^0_{(\Pi'_5,<'),X'}\emptyset = \mathcal{T}^1_{(\Pi'_5,<'),X'}\emptyset = \emptyset \neq X'$. In both cases, the preferred rules r_1 and r'_1, respectively, are (initially) inapplicable: $a \leftarrow b$ is not active wrt $(\emptyset, \{a, b\})$ and $a \leftarrow not\ b$ is not active wrt $(\emptyset, \{b\})$. And the application of the second rule $b \leftarrow$ is inhibited by Condition II: In the case of $\mathcal{T}^1_{(\Pi_5,<),X}\emptyset$, rule $a \leftarrow b$ is active wrt $(\{a, b\}, \emptyset)$; informally, X puts the construction on the false front that b will eventually be derivable. In the case of $\mathcal{T}^1_{(\Pi_5,<),X}\emptyset$, rule $a \leftarrow not\ b$ is active wrt $(\{b\}, \emptyset)$. This is due to the conception that a higher-ranked rule can never be defeated by a lower-ranked one.

Formal elaboration. We start with the basic properties of our consequence operator:

Theorem 1

Let $(\Pi, <)$ be an ordered program and let X and Y be sets of literals. Then, we have:

1. $\mathcal{T}_{(\Pi,<),Y}X \subseteq T_{\Pi,Y}X$.
2. $\mathcal{T}_{(\Pi,\emptyset),Y}X = T_{\Pi,Y}X$.

For $i = 1, 2$, let X_i and Y_i be sets of literals and $<_i \subseteq \Pi \times \Pi$ be strict partial orders.

3. If $X_1 \subseteq X_2$, then $\mathcal{T}_{(\Pi,<),Y}X_1 \subseteq \mathcal{T}_{(\Pi,<),Y}X_2$.
4. If $Y_1 \subseteq Y_2$, then $\mathcal{T}_{(\Pi,<),Y_2}X \subseteq \mathcal{T}_{(\Pi,<),Y_1}X$.
5. If $<_1 \subseteq <_2$, then $\mathcal{T}_{(\Pi,<_2),Y}X \subseteq \mathcal{T}_{(\Pi,<_1),Y}X$.

The next results show how our fixpoint operator relates to its classical counterpart.

Theorem 2
Let $(\Pi, <)$ be an ordered program and let X be a set of literals. Then, we have:

1. $\mathcal{C}_{(\Pi,<)}(X) \subseteq C_\Pi(X)$.
2. $\mathcal{C}_{(\Pi,<)}(X) = C_\Pi(X)$, if $X \subseteq \mathcal{C}_{(\Pi,<)}(X)$.
3. $\mathcal{C}_{(\Pi,\emptyset)}(X) = C_\Pi(X)$.

We obtain the following two corollaries.

Corollary 3
Let $(\Pi, <)$ be an ordered logic program and X a set of literals.
If X is a preferred answer set of $(\Pi, <)$, then X is an answer set of Π.

Our strategy thus implements a selection function among the standard answer sets of the underlying program. This selection is neutral in the absence of preferences, as shown next.

Corollary 4
Let Π be a logic program and X a set of literals.
Then, X is a preferred answer set of $(\Pi, \emptyset)$ iff X is an answer set of Π.

Of interest in view of an alternating fixpoint theory is that $\mathcal{C}_{(\Pi,<)}$ enjoys *anti-monotonicity*:

Theorem 5
Let $(\Pi, <)$ be an ordered logic program and X_1, X_2 sets of literals.
If $X_1 \subseteq X_2$, then $\mathcal{C}_{(\Pi,<)}(X_2) \subseteq \mathcal{C}_{(\Pi,<)}(X_1)$.

We next show that for any answer set X of a program Π, there is an ordering $<$ on the rules of Π such that X is the unique preferred answer set of $(\Pi, <)$.

Theorem 6
Let Π be a logic program and X an answer set of Π. Then, there is a strict partial order $<$ such that X is the unique preferred answer set of the ordered program $(\Pi, <)$.

Our last result shows that a total order selects at most one standard answer set.

Theorem 7
Let $(\Pi, \ll)$ be an ordered logic program and $\ll$ be a total order.
Then, $(\Pi, \ll)$ has zero or one preferred answer set.

Relationship to perfect model semantics. Any sensible semantics for logic programming should yield, in one fashion or other, the smallest Herbrand model $Cn(\Pi)$ whenever Π is a basic program. A similar consensus seems to exist regarding the *perfect model semantics* of *stratified* normal programs (Apt et al. 1987; Przymusinski 1988). Interestingly, stratified programs can be associated with a rule ordering in a canonical way. We now show that our semantics corresponds to the perfect model semantics on stratified normal programs.

A normal logic program Π is *stratified*, if Π has a partition, called *stratification*, $\Pi = \Pi_1 \cup \ldots \cup \Pi_n$ such that the following conditions are satisfied for $i, j \in \{1, \ldots, n\}$:

1. $\Pi_i \cap \Pi_j = \emptyset$ for $i \neq j$;
2. $\mathit{body}^+(r) \cap (\bigcup_{k=i+1}^{n} \mathit{head}(\Pi_k)) = \emptyset$ and $\mathit{body}^-(r) \cap (\bigcup_{k=i}^{n} \mathit{head}(\Pi_k)) = \emptyset$ for all $r \in \Pi_i$.

That is, whenever a rule r belongs to Π_i, the atoms in $\mathit{body}^+(r)$ can only appear in the heads of $\bigcup_{k=1}^{i} \Pi_k$, while the atoms in $\mathit{body}^-(r)$ can only appear in the heads of $\bigcup_{k=1}^{i-1} \Pi_k$.

A stratification somehow reflects an intrinsic order among the rules of a program. In a certain sense, rules in lower levels are preferred over rules in higher levels, insofar as rules in lower levels should be considered before rules in higher levels. Accordingly, the intuition behind the perfect model of a stratified program is to gradually derive atoms, starting from the most preferred rules. Specifically, one first applies the rules in Π_1, resulting in a set of atoms X_1; then one applies the rules in Π_2 relative to the atoms in X_1; and so on.

Formally, the *perfect model semantics* of a stratified logic program $\Pi = \Pi_1 \cup \ldots \cup \Pi_n$ is recursively defined for $0 < i < n$ as follows (Apt et al. 1987; Przymusinski 1988).

1. $X_0 = \emptyset$
2. $X_{i+1} = \bigcup_{j \geq 0} T^{j}_{\Pi_{i+1}, X_i} X_i$

The *perfect model* $X^\star$ of Π is then defined as $X^\star = X_n$.

Let Π be a stratified logic program and $\Pi = \Pi_1 \cup \ldots \cup \Pi_n$ be a stratification of Π. A natural priority relation $<_s$ on Π can be defined as follows:

$$\text{For any } r_1, r_2 \in \Pi, \text{ we define } r_1 <_s r_2 \text{ iff } r_1 \in \Pi_i \text{ and } r_2 \in \Pi_j \text{ such that } j < i\ .$$

That is, r_2 is preferred to r_1 if the level of r_2 is lower than that of r_1. We obtain thus an ordered logic program $(\Pi, <_s)$ for any stratified logic program Π with a fixed stratification.

Theorem 8
Let $X^\star$ be the perfect model of stratified logic program Π and let $<_s$ be an order induced by some stratification of Π. Then, we have

1. $X^\star = \mathcal{C}_{(\Pi, <_s)}(X^\star)$,
2. If $X \subseteq \mathcal{C}_{(\Pi, <_s)}(X)$, then $X^\star = X$.

These results imply the following theorem.

Corollary 9
Let $X^\star$ be the perfect model of stratified logic program Π and let $<_s$ be an order induced by some stratification of Π. Then , $(\Pi, <_s)$ has the unique preferred answer set $X^\star$.

Interestingly, both programs Π_5 as well as Π'_5 are stratifiable. None of the induced orderings, however, contains the respective preference ordering imposed in (5). In fact, this provides an easy criterion for the existence of (unique) preferred answer sets.

Corollary 10

Let $X^\star$ be the perfect model of stratified logic program Π and let $<_s$ be an order induced by some stratification of Π. Let $(\Pi, <)$ be an ordered logic program such that $< \subseteq <_s$.

Then , $(\Pi, <)$ has the unique preferred answer set $X^\star$.

Implementation through compilation. A translation of ordered logic programs to standard programs is developed in (Delgrande et al. 2000b). Although the employed strategy (cf. Section 4) differs from the one put forward in the previous section, it turns out that the computation of preferred answer sets can be accomplished by means of this translation technique in a rather straightforward way. In the framework of (Delgrande et al. 2000b), preferences are expressed within the program via a predicate symbol $\prec$. A logic program over a propositional language $\mathcal{L}$ is said to be *dynamically* ordered iff $\mathcal{L}$ contains the following pairwise disjoint categories: (i) a set N of terms serving as *names* for rules; (ii) a set At of atoms; and (iii) a set $At_\prec$ of *preference atoms* $s \prec t$, where $s, t \in N$ are names. For a program Π, we need a bijective function $n(\cdot)$ assigning a name $n(r) \in N$ to each rule $r \in \Pi$. We sometimes write n_r instead of $n(r)$. An atom $n_r \prec n_{r'} \in At_\prec$ amounts to asserting that $r < r'$ holds. A (statically) ordered program $(\Pi, <)$ can thus be captured by programs containing preference atoms only among their facts; it is then expressed by the program $\Pi \cup \{(n_r \prec n_{r'}) \leftarrow \ \mid r < r'\}$.

Given $r < r'$, one wants to ensure that r' is considered before r (cf. Condition II in Definition 2). For this purpose, one needs to be able to detect when a rule has been applied or when a rule is defeated. For detecting blockage, a new atom $\mathsf{bl}(n_r)$ is introduced for each r in Π. Similarly, an atom $\mathsf{ap}(n_r)$ is introduced to indicate that a rule has been applied. For controlling application of rule r the atom $\mathsf{ok}(n_r)$ is introduced. Informally, one concludes that it is ok to apply a rule just if it is ok with respect to every $<$-greater rule; for such a $<$-greater rule r', this will be the case just when r' is known to be blocked or applied.

More formally, given a dynamically ordered program Π over $\mathcal{L}$, let $\mathcal{L}^+$ be the language obtained from $\mathcal{L}$ by adding, for each $r, r' \in \Pi$, new pairwise distinct propositional atoms $\mathsf{ap}(n_r)$, $\mathsf{bl}(n_r)$, $\mathsf{ok}(n_r)$, and $\mathsf{rdy}(n_r, n_{r'})$. Then, the translation $\mathbb{T}$ maps an ordered program Π over $\mathcal{L}$ into a standard program $\mathbb{T}(\Pi)$ over $\mathcal{L}^+$ in the following way.

Definition 4

Let $\Pi = \{r_1, \ldots, r_k\}$ be a dynamically ordered logic program over $\mathcal{L}$.

Then, the logic program $\mathbb{T}(\Pi)$ over $\mathcal{L}^+$ is defined as $\mathbb{T}(\Pi) = \bigcup_{r \in \Pi} \tau(r)$, where $\tau(r)$

consists of the following rules, for $L^+ \in \mathit{body}^+(r)$, $L^- \in \mathit{body}^-(r)$, and $r', r'' \in \Pi$:

$$
\begin{array}{rcrcl}
a_1(r) & = & \mathit{head}(r) & \leftarrow & \mathsf{ap}(n_r) \\
a_2(r) & = & \mathsf{ap}(n_r) & \leftarrow & \mathsf{ok}(n_r), \mathit{body}(r) \\
b_1(r, L^+) & = & \mathsf{bl}(n_r) & \leftarrow & \mathsf{ok}(n_r), \mathit{not}\ L^+ \\
b_2(r, L^-) & = & \mathsf{bl}(n_r) & \leftarrow & \mathsf{ok}(n_r), L^- \\
c_1(r) & = & \mathsf{ok}(n_r) & \leftarrow & \mathsf{rdy}(n_r, n_{r_1}), \ldots, \mathsf{rdy}(n_r, n_{r_k}) \\
c_2(r, r') & = & \mathsf{rdy}(n_r, n_{r'}) & \leftarrow & \mathit{not}\ (n_r \prec n_{r'}) \\
c_3(r, r') & = & \mathsf{rdy}(n_r, n_{r'}) & \leftarrow & (n_r \prec n_{r'}), \mathsf{ap}(n_{r'}) \\
c_4(r, r') & = & \mathsf{rdy}(n_r, n_{r'}) & \leftarrow & (n_r \prec n_{r'}), \mathsf{bl}(n_{r'}) \\
c_5(r, r') & = & \mathsf{rdy}(n_r, n_{r'}) & \leftarrow & (n_r \prec n_{r'}), \mathit{head}(r') \\
t(r, r', r'') & = & n_r \prec n_{r''} & \leftarrow & n_r \prec n_{r'}, n_{r'} \prec n_{r''} \\
as(r, r') & = & \neg(n_{r'} \prec n_r) & \leftarrow & n_r \prec n_{r'}
\end{array}
$$

We write $\mathbb{T}(\Pi, <)$ rather than $\mathbb{T}(\Pi')$, whenever Π' is the dynamically ordered program capturing $(\Pi, <)$. The first four rules of $\tau(r)$ express applicability and blocking conditions of the original rules. For each rule $r \in \Pi$, we obtain two rules, $a_1(r)$ and $a_2(r)$, along with n rules of the form $b_1(r, L^+)$ and m rules of the form $b_2(r, L^-)$, where n and m are the numbers of the literals in $\mathit{body}^+(r)$ and $\mathit{body}^-(r)$, respectively. The second group of rules encodes the strategy for handling preferences. The first of these rules, $c_1(r)$, "quantifies" over the rules in Π. This is necessary when dealing with dynamic preferences since preferences may vary depending on the corresponding answer set. The four rules $c_i(r, r')$ for $i = 2..5$ specify the pairwise dependency of rules in view of the given preference ordering: For any pair of rules r, r', we derive $\mathsf{rdy}(n_r, n_{r'})$ whenever $n_r \prec n_{r'}$ fails to hold, or otherwise whenever either $\mathsf{ap}(n_{r'})$ or $\mathsf{bl}(n_{r'})$ is true, or whenever $\mathit{head}(r')$ has already been derived. This allows us to derive $\mathsf{ok}(n_r)$, indicating that r may potentially be applied whenever we have for all r' with $n_r \prec n_{r'}$ that r' has been applied or cannot be applied.

It is instructive to observe how close this specification of $\mathsf{ok}(\cdot)$ and $\mathsf{rdy}(\cdot, \cdot)$ is to Condition II in Definition 1. In fact, given a fixed $r \in \Pi$, Condition II can be read as follows.

$II.$ *for every* $r' \in \Pi$ *with* $r < r'$ *either*
(a) r' *is not active wrt* (Y, X) *or*
(b) $\mathit{head}(r') \in X$

The quantification over all rules $r' \in \Pi$ with $r < r'$ is accomplished by means of $c_1(r)$ (along with $c_2(r, r')$). By definition, r' is not active wrt (Y, X)[1] if either $\mathit{body}^+(r) \not\subseteq Y$ or $\mathit{body}^-(r) \cap X \neq \emptyset$, both of which are detected by rule $c_4(r, r')$. The condition $\mathit{head}(r') \in X$ is reflected by $c_3(r, r')$ and $c_5(r, r')$. While the former captures the case where $\mathit{head}(r')$ was supplied by r' itself,[2] the latter accounts additionally for the case where $\mathit{head}(r')$ was supplied by another rule than r'.

The next result shows that translation $\mathbb{T}$ is a realization of operator $\mathcal{C}$.

[1] Recall that X is supposed to contain the set of conclusions that have been derived so far, while Y provides the putative answer set.
[2] Strictly speaking rule $c_3(r, r')$ is subsumed by $c_5(r, r')$; nonetheless we keep both for conceptual clarity in view of similar translations presented in Section 4.

Theorem 11
Let $(\Pi, <)$ be an ordered logic program over $\mathcal{L}$ and let $X \subseteq \{head(r) \mid r \in \Pi\}$ be a consistent set of literals. Then, there is some set of literals Y over $\mathcal{L}^+$ where $X = Y \cap \mathcal{L}$ such that $\mathcal{C}_{(\Pi,<)}(X) = C_{\mathbb{T}(\Pi,<)}(Y) \cap \mathcal{L}$.

Note that the fixpoints of $\mathcal{C}_{(\Pi,<)}$ constitute a special case the previous theorem.

Theorem 12
Let $(\Pi, <)$ be an ordered logic program over $\mathcal{L}$ and let X and Y be consistent sets of literals. Then, we have that

1. if $\mathcal{C}_{(\Pi,<)}(X) = X$, then there is an answer set Y of $\mathbb{T}(\Pi, <)$ such that $X = Y \cap \mathcal{L}$;
2. if Y is an answer set of $\mathbb{T}(\Pi, <)$, then $\mathcal{C}_{(\Pi,<)}(Y \cap \mathcal{L}) = Y \cap \mathcal{L}$.

4 Other strategies (and characterizations)

We now show how the approaches of Delgrande et al. (2000b) and Brewka/Eiter (1999; 2000) can be captured within our framework. Also, we take up a complementary characterization provided in (Delgrande et al. 2000b) in order to obtain another insightful perspective on the three approaches. For clarity, we add the letter "W" to all concepts from Section 3. Accordingly we add "D" and "B", respectively, when dealing with the two aforementioned approaches.

Characterizing D-preference. In (Delgrande et al. 2000b), the selection of preferred answer sets is characterized in terms of the underlying set of generating rules: The set $\Gamma_\Pi X$ of all *generating rules* of a(n answer) set X of literals from program Π is given by

$$\Gamma_\Pi X = \{r \in \Pi \mid \mathit{body}^+(r) \subseteq X \text{ and } \mathit{body}^-(r) \cap X = \emptyset\} \ .$$

The property distinguishing preferred answer sets from ordinary ones is referred to as *order preservation* and defined in the following way.

Definition 5
Let $(\Pi, <)$ be an ordered program and let X be an answer set of Π.

Then, X is called $<^{\mathrm{D}}$-preserving, if there exists an enumeration $\langle r_i \rangle_{i \in I}$ of $\Gamma_\Pi X$ such that for every $i, j \in I$ we have that:

1. $\mathit{body}^+(r_i) \subseteq \{head(r_j) \mid j < i\}$; and
2. if $r_i < r_j$, then $j < i$; and
3. if $r_i < r'$ and $r' \in \Pi \setminus \Gamma_\Pi X$, then
   (a) $\mathit{body}^+(r') \not\subseteq X$ or
   (b) $\mathit{body}^-(r') \cap \{head(r_j) \mid j < i\} \neq \emptyset$.

We often refer to $<^{\mathrm{D}}$-preserving answer sets as D-preferred answer sets.

Condition 1 makes the property of *groundedness*[3] explicit. Although any standard answer set enjoys this property, we will see that its interaction with preferences varies with

[3] This term is borrowed from the literature on default logic (cf. (Konolige 1988; Schwind 1990)).

the strategy. Condition 2 stipulates that $\langle r_i \rangle_{i \in I}$ is *compatible* with $<$, a property invariant to all of the considered approaches. Lastly, Condition 3 is comparable with Condition II in Definition 1; it guarantees that rules can never be blocked by lower-ranked rules.

Roughly speaking, an order preserving enumeration of the set of generating rules reflects the sequence of successive rule applications leading to some preferred answer set. For instance, the preferred answer set $X = \{p, b, \neg f, w\}$ of Example (3) can be generated by the two order preserving sequences $\langle r_5, r_4, r_1, r_2 \rangle$ and $\langle r_5, r_1, r_4, r_2 \rangle$. Intuitively, both enumerations are order preserving since they reflect the fact that r_1 is treated before r_2. [4] Although there is another grounded enumeration generating X, namely $\langle r_5, r_4, r_2, r_1 \rangle$, it is not order preserving since it violates Condition 2. The same applies to the only grounded enumeration $\langle r_5, r_4, r_2, r_3 \rangle$ that allows to generate the second standard answer set of Π_3; it violates Condition 3b. Consequently, X is the only $<^{\mathrm{D}}$-preserving answer set of $(\Pi_3, <)$.

We are now ready to provide a fixpoint definition for D-preference. For this purpose, we assume a bijective mapping $rule(\cdot)$ among rule heads and rules, that is, $rule(head(r)) = r$; accordingly, $rule(\{head(r) \mid r \in R\}) = R$. Such mappings can be defined in a bijective way by distinguishing different occurrences of literals.

Definition 6

Let $(\Pi, <)$ be an ordered logic program and let X and Y be sets of literals.

We define the set of immediate D-consequences of X with respect to $(\Pi, <)$ and Y as

$$\mathcal{T}^{\mathrm{D}}_{(\Pi,<),Y} X \quad = \quad \left\{ head(r) \;\middle|\; \begin{array}{rl} I. & r \in \Pi \text{ is active wrt } (X, Y) \text{ and} \\ II. & \text{there is no rule } r' \in \Pi \text{ with } r < r' \\ & \text{such that} \\ & (a)\ r' \text{ is active wrt } (Y, X) \text{ and} \\ & (b)\ r' \notin rule(X) \end{array} \right\}$$

if X is consistent, and $\mathcal{T}^{\mathrm{D}}_{(\Pi,<),Y} X = Lit$ otherwise.

The distinguishing feature between this definition and Definition 1 manifests itself in IIb. While D-preference requires that a higher-ranked rule has effectively applied, W-preference contents itself with the presence of the head of the rule, no matter whether this was supplied by the rule itself.

Defining iterated applications of $\mathcal{T}^{\mathrm{D}}_{(\Pi,<),Y}$ in analogy to those of $\mathcal{T}_{(\Pi,<),Y}$, we may capture D-preference by means of a fixpoint operator in the following way.

Definition 7

Let $(\Pi, <)$ be an ordered logic program and let X be a set of literals.

We define $\mathcal{C}^{\mathrm{D}}_{(\Pi,<)}(X) = \bigcup_{i \geq 0} (\mathcal{T}^{\mathrm{D}})^i_{(\Pi,<),X} \emptyset$.

A similar elaboration of $\mathcal{C}^{\mathrm{D}}_{(\Pi,<)}$ as done with $\mathcal{C}^{\mathrm{W}}_{(\Pi,<)}$ in Section 3 yields identical formal properties; in particular, $\mathcal{C}^{\mathrm{D}}_{(\Pi,<)}$ also enjoys anti-monotonicity.

The aforementioned difference is nicely illustrated by extending the programs in (5) by

[4] Note that both enumerations are compatible with the iteration through $\mathcal{T}^i_{(\Pi_3,<),X} \emptyset$ for $i = 0..4$.

rule $a \leftarrow$, yielding $(\Pi_6, <)$ and $(\Pi'_6, <')$, respectively:

$$\begin{array}{rcrcl} r_1 & = & a & \leftarrow & b \\ r_2 & = & b & \leftarrow & \\ r_3 & = & a & \leftarrow & \\ & & r_2 < r_1 & & \end{array} \qquad \begin{array}{rcrcl} r'_1 & = & a & \leftarrow & not\ b \\ r'_2 & = & b & \leftarrow & \\ r'_3 & = & a & \leftarrow & \\ & & r'_2 <' r'_1 & & \end{array} \tag{6}$$

While in both cases the single standard answer set is W-preferred, neither of them is D-preferred. Let us illustrate this in terms of the iterated applications of $\mathcal{T}^{\text{W}}_{(\Pi_6,<),X}$ and $\mathcal{T}^{\text{D}}_{(\Pi_6,<),X}$, where $X = \{a, b\}$ is the standard answer set of Π_6: At first, both operators allow for applying rule $a \leftarrow$, resulting in $\{a\}$. As with $\mathcal{T}^{\text{W}}_{(\Pi_5,<),X}$ in (5), however, operator $\mathcal{T}^{\text{D}}_{(\Pi_6,<),X}$ does not allow for applying r_2 at the next stage, unless r_1 is inactive. This requirement is now dropped by $\mathcal{T}^{\text{W}}_{(\Pi_6,<),X}$, since the head of r_1 has already been derived through r_3. In such a case, the original preference is ignored, which enables the application of r_2. In this way, we obtain the W-preferred answer set $X = \{a, b\}$. The analogous behavior is observed on $(\Pi'_6, <')$.

As W-preferred answer sets, D-preferred ones coincide with the perfect model on stratified programs.

Theorem 13
Let $X^\star$ be the perfect model of stratified logic program Π and let $<_s$ be an order induced by some stratification of Π. Then , $(\Pi, <_s)$ has the unique D-preferred answer set $X^\star$.

The subtle difference between D- and W-preference is also reflected in the resulting compilation. Given the same prerequisites as in Definition 4, the logic program $\mathbb{T}^{\text{D}}(\Pi)$ over $\mathcal{L}^+$ is defined as $\mathbb{T}^{\text{D}}(\Pi) = \mathbb{T}^{\text{W}}(\Pi) \setminus \{c_5(r, r') \mid r, r' \in \Pi\}$. Hence, in terms of this compilation technique, the distinguishing feature between D- and W-preference manifests itself in the usage of rule $c_5(r, r') : \mathsf{rdy}(n_r, n_{r'}) \leftarrow (n_r \prec n_{r'}), head(r')$. While W-preference allows for suspending a preference whenever the head of the preferred rule was derived, D-preference stipulates the application of the preferred rule itself. This is reflected by the fact that the translation $\mathbb{T}^{\text{D}}$ merely uses rule $c_3(r, r') : \mathsf{rdy}(n_r, n_{r'}) \leftarrow (n_r \prec n_{r'}), \mathsf{ap}(n_{r'})$ to enforce that the preferred rule itself has been applied. This demonstrates once more how closely the compilation technique follows the specification given in the fixpoint operation.

As shown in (Delgrande et al. 2000b), a set of literals X is a $<^{\text{D}}$-preserving answer set of a program Π iff $X = Y \cap \mathcal{L}$ for some answer set Y of $\mathbb{T}^{\text{D}}(\Pi, <)$. This result naturally extends to the fixpoint operator $\mathcal{C}^{\text{D}}_{(\Pi,<)}$, as shown in the following result.

Theorem 14
Let $(\Pi, <)$ be an ordered logic program over $\mathcal{L}$ and let X be a consistent set of literals. Then, the following propositions are equivalent.

1. $\mathcal{C}^{\text{D}}_{(\Pi,<)}(X) = X$;
2. $X = Y \cap \mathcal{L}$ for some answer set Y of $\mathbb{T}^{\text{D}}(\Pi, <)$;
3. X is a $<^{\text{D}}$-preserving answer set of Π.

While the last result dealt with effective answer sets, the next one shows that applying $\mathcal{C}^{\text{D}}_{(\Pi,<)}$ is equivalent to the application of $C_{\Pi'}$ to the translated program $\Pi' = \mathbb{T}^{\text{D}}(\Pi, <)$.

Theorem 15
Let $(\Pi, <)$ be an ordered logic program over $\mathcal{L}$ and let $X \subseteq \{head(r) \mid r \in \Pi\}$ be a consistent set of literals. Then, there is some set of literals Y over $\mathcal{L}^+$ where $X = Y \cap \mathcal{L}$ such that $\mathcal{C}^{\mathrm{D}}_{(\Pi,<)}(X) = C_{\mathbb{T}^{\mathrm{D}}(\Pi,<)}(Y) \cap \mathcal{L}$.

Characterizing W*-preference (alternatively).* We now briefly elaborate upon a characterization of W-preference in terms of order preservation. This is interesting because order preservation provides an alternative perspective on the formation of answer sets. In contrast to the previous fixpoint characterizations, order preservation furnishes an account of preferred answer sets in terms of the underlying generating rules. While an immediate consequence operator provides a rather rule-centered and thus local characterization, order preservation gives a more global and less procedural view on an entire construction. In particular, the underlying sequence nicely reflects the interaction of its properties. In fact, we see below that different approaches distinguish themselves by a different degree of interaction between groundedness and preferences.

Definition 8
Let $(\Pi, <)$ be an ordered program and let X be an answer set of Π.

Then, X is called $<^{\mathrm{W}}$-preserving, if there exists an enumeration $\langle r_i \rangle_{i \in I}$ of $\Gamma_\Pi X$ such that for every $i, j \in I$ we have that:

1. (a) $body^+(r_i) \subseteq \{head(r_j) \mid j < i\}$ or
   (b) $head(r_i) \in \{head(r_j) \mid j < i\}$; and
2. if $r_i < r_j$, then $j < i$; and
3. if $r_i < r'$ and $r' \in \Pi \setminus \Gamma_\Pi X$, then
   (a) $body^+(r') \not\subseteq X$ or
   (b) $body^-(r') \cap \{head(r_j) \mid j < i\} \neq \emptyset$ or
   (c) $head(r') \in \{head(r_j) \mid j < i\}$.

The primary difference between this concept of order preservation and the one for D-preference is clearly the weaker notion of groundedness. While D-preference makes no compromise when enforcing rule dependencies induced by preference, W-preference “smoothes” their integration with those induced by groundedness and defeat relationships: First, regarding rules in $\Gamma_\Pi X$ (via Condition 1b) and second concerning rules in $\Pi \setminus \Gamma_\Pi X$ (via Condition 3c). The rest of the definition is identical to Definition 5.

This “smoothed” integration of preferences with groundedness and defeat dependencies is nicely illustrated by programs $(\Pi_6, <)$ and $(\Pi'_6, <)$. Regarding Π_6, we observe that there is no enumeration of $\Gamma_\Pi X$ satisfying both Condition 1a and 2. Rather it is Condition 1b that weakens the interaction between both conditions by tolerating enumeration $\langle r_3, r_2, r_1 \rangle$. A similar observation can be made regarding Π'_6, where, in contrast to Π_6, the preferred rule r'_1 does not belong to $\Gamma_\Pi X$. We observe that there is no enumeration of $\Gamma_\Pi X$ satisfying both Condition 2 and 3a/b. Now, it is Condition 3c that weakens the interaction between both conditions by tolerating enumeration $\langle r'_3, r'_2 \rangle$. In fact, the two examples show that both Condition 1b as well as 3c function as exceptions to conditions 1a and 3a/b, respectively. In this way, W-preference imposes the same requirements as D-preference, *unless* the head of the rule in focus has already been derived by other means.

Finally, we have the following summarizing result.

Theorem 16
Let $(\Pi, <)$ be an ordered logic program over $\mathcal{L}$ and let X be a consistent set of literals. Then, the following propositions are equivalent.

1. $\mathcal{C}^{\mathrm{W}}_{(\Pi,<)}(X) = X$;
2. $X = Y \cap \mathcal{L}$ for some answer set Y of $\mathbb{T}^{\mathrm{W}}(\Pi, <)$;
3. X is a $<^{\mathrm{W}}$-preserving answer set of Π.

Characterizing B*-preference.* Another approach to preference is proposed in (Brewka and Eiter 1999). This approach differs in two ways from the previous ones. First, the construction of answer sets is separated from verifying preferences. Interestingly, this verification is done on the basis of the prerequisite-free program obtained from the original one by "evaluating" $body^+(r)$ for each rule r wrt the separately constructed (standard) answer set. Second, rules that may lead to counter-intuitive results are explicitly removed. This is spelled out in (Brewka and Eiter 2000), where the following filter is defined:

$$\mathcal{E}_X(\Pi) = \Pi \setminus \{r \in \Pi \mid head(r) \in X, body^-(r) \cap X \neq \emptyset\} \tag{7}$$

Accordingly, we define $\mathcal{E}_X(\Pi, <) = (\mathcal{E}_X(\Pi), < \cap\, (\mathcal{E}_X(\Pi) \times \mathcal{E}_X(\Pi))\,)$.

We begin with a formal account of B-preferred answer sets. In this approach, partially ordered programs are reduced to totally ordered ones: A *fully ordered logic program* is an ordered logic program $(\Pi, \ll)$ where $\ll$ is a total ordering. The case of arbitrarily ordered programs is reduced to this restricted case: Let $(\Pi, <)$ be an ordered logic program and let X be a set of literals. Then, X is a B-preferred answer set of $(\Pi, <)$ iff X is a B-preferred answer set of some fully ordered logic program $(\Pi, \ll)$ such that $< \subseteq \ll$.

The construction of B-preferred answer sets relies on an operator, defined for prerequisite-free programs, comprising only rules r with $body^+(r) = \emptyset$.

Definition 9
Let $(\Pi, \ll)$ be a fully ordered prerequisite-free logic program, let $\langle r_i \rangle_{i \in I}$ be an enumeration of Π according to $\ll$, and let X be a set of literals. Then, $\mathcal{B}_{(\Pi,\ll)}(X)$ is the smallest logically closed set of literals containing $\bigcup_{i \in I} X_i$, where $X_j = \emptyset$ for $j \notin I$ and

$$X_i = \begin{cases} X_{i-1} & \text{if } body^-(r_i) \cap X_{i-1} \neq \emptyset \\ X_{i-1} \cup \{head(r_i)\} & \text{otherwise.} \end{cases}$$

This construction is unique insofar that for any such program $(\Pi, \ll)$, there is at most one standard answer set X of Π such that $\mathcal{B}_{\mathcal{E}_X(\Pi,\ll)}(X) = X$. Accordingly, this set is used for defining the B*-preferred answer set* of a prerequisite-free logic program:

Definition 10
Let $(\Pi, \ll)$ be a fully ordered prerequisite-free logic program and let X be a set of literals. Then, X is the B-preferred answer set of $(\Pi, \ll)$ iff $\mathcal{B}_{\mathcal{E}_X(\Pi,\ll)}(X) = X$.

The reduction of $(\Pi, \ll)$ to $\mathcal{E}_X(\Pi, \ll)$ removes from the above construction all rules whose heads are in X but which are defeated by X. This is illustrated in (Brewka and Eiter 2000) through the following example:

$$\begin{array}{llllll} r_1 = & a & \leftarrow & not\ b, & r_3 = a \leftarrow not\ \neg a, & \{r_j < r_i \mid i < j\}\,. \\ r_2 = & \neg a & \leftarrow & not\ a, & r_4 = b \leftarrow not\ \neg b, & \end{array} \tag{8}$$

Program $\Pi_8 = \{r_1, \ldots, r_4\}$ has two answer sets, $\{a, b\}$ and $\{\neg a, b\}$. The application of operator $\mathcal{B}$ relies on sequence $\langle r_1, r_2, r_3, r_4 \rangle$. Now, consider the processes induced by $\mathcal{B}_{\mathcal{E}_X(\Pi_8,<)}(X)$ and $\mathcal{B}_{(\Pi_8,<)}(X)$ for $X = \{a, b\}$, respectively:

$$\begin{array}{rllll} \mathcal{B}_{\mathcal{E}_X(\Pi_8,<)}(X): & X_1 = \{\} & X_2 = \{\neg a\} & X_3 = \{\neg a\} & X_4 = \{\neg a, b\} \\ \mathcal{B}_{(\Pi_8,<)}(X): & X'_1 = \{a\} & X'_2 = \{a\} & X'_3 = \{a\} & X'_4 = \{a, b\} \end{array}$$

Thus, without filtering by $\mathcal{E}_X$, we get $\{a, b\}$ as a B-preferred answer set. As argued in (Brewka and Eiter 2000), such an answer set does not preserve priorities because r_2 is defeated in $\{a, b\}$ by applying a rule which is less preferred than r_2, namely r_3. The above program has therefore no B-preferred answer set.

The next definition accounts for the general case by reducing it to the prerequisite-free one. For checking whether an answer set X is B-preferred, the prerequisites of the rules are evaluated wrt X. For this purpose, we define $r^- = \mathit{head}(r) \leftarrow \mathit{body}^-(r)$ for a rule r.

Definition 11
Let $(\Pi, \ll)$ be a fully ordered logic program and X a set of literals.

The logic program $(\Pi_X, \ll_X)$ is obtained from $(\Pi, \ll)$ as follows:

1. $\Pi_X = \{r^- \mid r \in \Pi \text{ and } \mathit{body}^+(r) \subseteq X\}$;
2. for any $r'_1, r'_2 \in \Pi_X$, $r'_1 \ll_X r'_2$ iff $r_1 \ll r_2$ where $r_i = \max_{\ll}\{r \in \Pi \mid r^- = r'_i\}$.

In other words, Π_X is obtained from Π by first eliminating every rule $r \in \Pi$ such that $\mathit{body}^+(r) \not\subseteq X$, and then substituting all remaining rules r by r^-.

In general, B-preferred answer sets are then defined as follows.

Definition 12
Let $(\Pi, \ll)$ be a fully ordered logic program and X a set of literals.

Then, X is a B-preferred answer set of $(\Pi, \ll)$, if

1. X is a (standard) answer set of Π, and
2. X is a B-preferred answer set of $(\Pi_X, \ll_X)$.

The distinguishing example of this approach is given by program $(\Pi_9, <)$:

$$\begin{array}{rclcl} r_1 & = & b & \leftarrow & a, \mathit{not}\ \neg b \\ r_2 & = & \neg b & \leftarrow & \mathit{not}\ b \\ r_3 & = & a & \leftarrow & \mathit{not}\ \neg a \end{array} \quad \text{with} \quad \{r_j < r_i \mid i < j\}\ . \tag{9}$$

Program $\Pi_9 = \{r_1, r_2, r_3\}$ has two standard answer sets: $X_1 = \{a, b\}$ and $X_2 = \{a, \neg b\}$. Both $(\Pi_9)_{X_1}$ as well as $(\Pi_9)_{X_2}$ turn r_1 into $b \leftarrow \mathit{not}\ \neg b$ while leaving r_2 and r_3 unaffected. Clearly, $\mathcal{E}_{X_i}(\Pi_9, <) = (\Pi_9, <)$ for $i = 1, 2$. Also, we obtain that $\mathcal{B}_{(\Pi_9,<)}(X_1) = X_1$, that is, X_1 is a B-preferred answer set. In contrast to this, X_2 is not B-preferred. To to see this, observe that $\mathcal{B}_{(\Pi_9,<)}(X_2) = X_1 \neq X_2$. That is, $\mathcal{B}_{(\Pi_9,<)}(X_2)$ reproduces X_1 rather than X_2. In fact, while X_1 is the only B-preferred set, neither X_1 nor X_2 is W- or D-preferred (see below).

We note that B-preference disagrees with W- and D-preference on Example (3). In fact, both answer sets of program $(\Pi_3, <)$ are B-preferred, while only $\{p, b, \neg f, w\}$ is W- and D-preferred. In order to shed some light on these differences, we start by providing a fixpoint characterization of B-preference:

Definition 13

Let $(\Pi, <)$ be an ordered logic program and let X and Y be sets of literals.

We define the set of immediate consequences of X with respect to $(\Pi, <)$ and Y as

$$\mathcal{T}^{\text{B}}_{(\Pi,<),Y} X \quad = \quad \left\{ head(r) \;\middle|\; \begin{array}{ll} I. & r \in \Pi \text{ is active wrt } (Y,Y) \text{ and} \\ II. & \text{there is no rule } r' \in \Pi \text{ with } r < r' \\ & \text{such that} \\ & (a)\ r' \text{ is active wrt } (Y,X) \text{ and} \\ & (b)\ head(r') \notin X \end{array} \right\}$$

if X is consistent, and $\mathcal{T}^{\text{B}}_{(\Pi,<),Y} X = Lit$ otherwise.

The difference between this operator[5] and its predecessors manifests itself in Condition I, where activeness is tested wrt (Y, Y) instead of (X, Y), as in Definition 1 and 4. In fact, in Example (9) it is the (unprovability of the) prerequisite a of the highest-ranked rule r_1 that makes the construction of W- or D-preferred answer sets break down (cf. Definition 1 and 4). This is avoided with B-preference because once answer set $\{a, b\}$ is provided, preferences are enforced wrt the program obtained by replacing r_1 with $b \leftarrow not\ \neg b$.

With an analogous definition of iterated applications of $\mathcal{T}^{\text{B}}_{(\Pi,<),Y}$X as above, we obtain the following characterization of B-preference:

Definition 14

Let $(\Pi, <)$ be an ordered logic program and let X be a set of literals.

We define $\mathcal{C}^{\text{B}}_{(\Pi,<)}(X) = \bigcup_{i \geq 0} (\mathcal{T}^{\text{B}})^i_{(\Pi,<),X} \emptyset$.

Unlike above, $\mathcal{C}^{\text{B}}_{(\Pi,<)}$ is not anti-monotonic. This is related to the fact that the "answer set property" of a set is verified separately (cf. Definition 12). We have the following result.

Theorem 17

Let $(\Pi, <)$ be an ordered logic program over $\mathcal{L}$ and let X be an answer set of Π.

Then, we have that X is B-preferred iff $\mathcal{C}^{\text{B}}_{\mathcal{E}_X(\Pi,<)}(X) = X$.

As with D- and W-preference, B-preference gives the perfect model on stratified programs.

Theorem 18

Let $X^\star$ be the perfect model of stratified logic program Π and let $<_s$ be an order induced by some stratification of Π. Then , $(\Pi, <_s)$ has the unique B-preferred answer set $X^\star$.

Alternatively, B-preference can also be captured by appeal to order preservation:

Definition 15

Let $(\Pi, <)$ be an ordered program and let X be an answer set of Π.

Then, X is called $<^{\text{B}}$-preserving, if there exists an enumeration $\langle r_i \rangle_{i \in I}$ of $\Gamma_\Pi X$ such that, for every $i, j \in I$, we have that:

1. if $r_i < r_j$, then $j < i$; and
2. if $r_i < r'$ and $r' \in \Pi \setminus \Gamma_\Pi X$, then

[5] We have refrained from integrating (7) in order to keep the fixpoint operator comparable to its predecessors. This is taken care of in Theorem 19. We note however that an integration of (7) would only affect Condition II.

(a) $body^+(r') \not\subseteq X$ or
(b) $body^-(r') \cap \{head(r_j) \mid j < i\} \neq \emptyset$ or
(c) $head(r') \in X$.

This definition differs in two ways from its predecessors. First, it drops any requirement on groundedness. This corresponds to using (Y, Y) instead of (X, Y) in Definition 13. Hence, groundedness is fully disconnected from order preservation. For example, the B-preferred answer set $\{a, b\}$ of $(\Pi_9, <)$ is associated with the $<^{\text{B}}$-preserving sequence $\langle r_1, r_2\rangle$, while the standard answer set is generated by the grounded sequence $\langle r_2, r_1\rangle$. Second, Condition 2c is more relaxed than in Definition 8. That is, any rule r' whose head is in X (as opposed to $\{head(r_j) \mid j < i\}$) is taken as "applied". Also, Condition 2c integrates the filter in (7).[6] For illustration, consider Example (6) extended by $r_3 < r_2$:

$$\begin{array}{rcll} r_1 & = & a \leftarrow & not\ b \qquad\qquad r_3 < r_2 < r_1 \\ r_2 & = & b \leftarrow & \\ r_3 & = & a \leftarrow & \end{array} \tag{10}$$

While this program has no D- or W-preferred answer set, it has a B-preferred one: $\{a, b\}$ generated by $\langle r_2, r_3\rangle$. The critical rule r_1 is handled by 2c. As a net result, Condition 2 is weaker than its counterpart in Definition 8. We have the following summarizing result.

Theorem 19
Let $(\Pi, <)$ be an ordered logic program over $\mathcal{L}$ and let X be a consistent answer set of Π. Then, the following propositions are equivalent.

1. X is B-preferred;
2. $\mathcal{C}^{\text{B}}_{\mathcal{E}_X(\Pi,<)}(X) = X$;
3. X is a $<^{\text{B}}$-preserving answer set of Π;
4. $X = Y \cap \mathcal{L}$ for some answer set Y of $\mathbb{T}^{\text{B}}(\Pi, <)$ (where $\mathbb{T}^{\text{B}}$ is defined in (Delgrande et al. 2000a)).

Unlike theorems 14 and 16, the last result stipulates that X must be an answer set of Π. This requirement can only be dropped in case 4, while all other cases rely on this property.

Relationships. First of all, we observe that all three approaches treat the blockage of (higher-ranked) rules in the same way. That is, a rule r' is found to be blocked if either its prerequisites in $body^+(r')$ are *never* derivable or if some member of $body^-(r')$ has been derived by higher-ranked or unrelated rules. This is reflected by the identity of conditions IIa and 2a/b in all three approaches, respectively. Although this is arguably a sensible strategy, it leads to the loss of preferred answer sets on programs like $(\Pi'_5, <')$.

The difference between D- and W-preference can be directly read off Definition 1 and 4; it manifests itself in Condition IIb and leads to the following relationships.

Theorem 20
Let $(\Pi, <)$ be an ordered logic program such that for $r, r' \in \Pi$ we have that $r \neq r'$ implies $head(r) \neq head(r')$. Let X be a set of literals. Then, X is a D-preferred answer set of $(\Pi, <)$ iff X is a W-preferred answer set of $(\Pi, <)$.

[6] Condition $body^-(r') \cap X \neq \emptyset$ in (7) is obsolete because $r' \notin \Gamma_\Pi X$.

The considered programs deny the suspension of preferences under W-preference, because all rule heads are derivable in a unique way. We have the following general result.

Theorem 21
Every D-preferred answer set is W-preferred.

Example (6) shows that the converse does not hold.

Interestingly, a similar relationship is obtained between W- and B-preference. In fact, Definition 15 can be interpreted as a weakening of Definition 8 by dropping the condition on groundedness and weakening Condition 2 (via 2c). We thus obtain the following result.

Theorem 22
Every W-preferred answer set is B-preferred.

Example (9) shows that the converse does not hold.

Let $\mathcal{AS}(\Pi) = \{X \mid C_\Pi(X) = X\}$ and $\mathcal{AS}_P(\Pi,<) = \{X \in \mathcal{AS}(\Pi) \mid X \text{ is } P\text{-preferred}\}$ for $P = \textsc{w}, \textsc{d}, \textsc{b}$. Then, we obtain the following summarizing result.

Theorem 23
Let $(\Pi,<)$ be an ordered logic program. Then, we have

$$\mathcal{AS}_{\textsc{d}}(\Pi,<) \subseteq \mathcal{AS}_{\textsc{w}}(\Pi,<) \subseteq \mathcal{AS}_{\textsc{b}}(\Pi,<) \subseteq \mathcal{AS}(\Pi)$$

This hierarchy is primarily induced by a decreasing interaction between groundedness and preference. While D-preference requires the full compatibility of both concepts, this interaction is already weakened in W-preference, before it is fully abandoned in B-preference. This is nicely reflected by the evolution of the condition on groundedness in definitions 5, 8, and 15. Notably, groundedness as such is not the ultimate distinguishing factor, as demonstrated by the fact that prerequisite-free programs do not necessarily lead to the same preferred answer sets, as witnessed in (6) and (10). Rather it is the degree of integration of preferences within the standard reasoning process that makes the difference.

Taking together theorems 9, 13, and 18, we obtain the following result.

Theorem 24
Let $X^\star$ be the perfect model of stratified logic program Π and let $<_s$ be an order induced by some stratification of Π. Let $(\Pi,<)$ be an ordered logic program such that $< \subseteq <_s$.

Then, we have $\mathcal{AS}_{\textsc{d}}(\Pi,<) = \mathcal{AS}_{\textsc{w}}(\Pi,<) = \mathcal{AS}_{\textsc{b}}(\Pi,<) = \mathcal{AS}(\Pi) = \{X^\star\}$.

5 Discussion and related work

Up to now, we have been dealing with static preferences only. In fact, all fixpoint characterizations are also amenable to dynamically ordered programs, as introduced in Section 4. To see this, consider Definition 1 along with a dynamically ordered program Π and sets of literals X, Y over a language extended by preference atoms $At_\prec$. Then, the corresponding preferred answer sets are definable by substituting "$r < r'$" by "$(r \prec r') \in Y$" in definitions 1, 6, and 13, respectively. That is, instead of drawing preference information from the external order $<$, we simply consult the initial context, expressed by Y. In this way, the preferred answer sets of Π can be given by the fixpoints of an operator $\mathcal{C}_\Pi$.

Also, we have concentrated so far on preferred answer sets semantics that amount to selection functions on the standard answer sets of the underlying program. Another strategy is advocated in (Gelfond and Son 1997), where the preference $d_1 < d_2$ *"stops the application of default d_2 if defaults d_1 and d_2 are in conflict with each other and the default d_1 is applicable"* (Gelfond and Son 1997). In contrast to B-, D-, and W-preference this allows for exclusively concluding $\neg p$ from program $(\{r_1, r_2\}, <)$:

$$r_1 \;=\; p \;\leftarrow \qquad\qquad r_2 \;=\; \neg p \;\leftarrow \qquad\qquad r_1 < r_2$$

This approach amounts to B-preference on certain "hierarchically" structured programs (Gelfond and Son 1997). A modification of the previous compilation techniques for this strategy is discussed in (Delgrande and Schaub 2000). Although conceptually different, one finds similar strategies when dealing with inheritance, update and/or dynamic logic programs (Buccafurri et al. 1999; Eiter et al. 2000; Alferes et al. 1998), respectively.

While all of the aformentioned approaches remain within the same complexity class, other approaches step up in the polynomial hierarchy (Rintanen 1995; Sakama and Inoue 1996; Zhang and Foo 1997). Among them, preferences on literals are investigated in (Sakama and Inoue 1996). In contrast to these approaches, so-called courteous logic programs (Grosof 1997) step down the polynomial hierarchy into P. Due to the restriction to acyclic positive logic programs a courteous answer set can be computed in $O(n^2)$ time. Other preference-based approaches that exclude negation as failure include (Dimopoulos and Kakas 1995; Pradhan and Minker 1996; You et al. 2001) as well as the framework of defeasible logics (Nute 1987; Nute 1994). A comparision of the latter with preferred well-founded semantics (as defined in (Brewka 1996)) can be found in (Brewka 2001).

In a companion paper, we exploit our fixpoint operators for defining regular and well-founded semantics for ordered logic programs within an alternating fixpoint theory.[7] This yields a surprising yet negative result insofar as these operators turn out to be too weak in the setting of well-founded semantics. We address this by defining a parameterizable framework for preferred well-founded semantics, summarized in (Schaub and Wang 2002).

6 Conclusion

The notion of preference seems to be pervasive in logic programming when it comes to knowledge representation. This is reflected by numerous approaches that aim at enhancing logic programming with preferences in order to improve knowledge representation capacities. Despite the large variety of approaches, however, only very little attention has been paid to their structural differences and sameness, finally leading to solid semantical underpinnings. In particular, there were up to now only few attempts to characterize one approach in terms of another one. The lack of this kind of investigation is clearly due to the high diversity of existing approaches.

[7] This material was removed from this paper due to space restrictions.

This work is a first step towards a systematic account to logic programming with preferences. To this end, we employ fixpoint operators following the tradition of logic programming. We elaborated upon three different approaches that were originally defined in rather heterogenous ways. We obtained three alternative yet uniform ways of characterizing preferred answer sets (in terms of fixpoints, order preservation, and an axiomatic account). The underlying uniformity provided us with a deeper understanding of how and which answer sets are preferred in each approach. This has led to a clarification of their relationships and subtle differences. On the one hand, we revealed that the investigated approaches yield an increasing number of answer sets depending on how tight they connect preference to groundedness. On the other hand, we demonstrated how closely the compilation technique developed in (Delgrande et al. 2000b) follows the specification given in the fixpoint operation. Also, we have shown that all considered answer sets semantics correspond to the perfect models semantics whenever the underlying ordering stratifies the program.

We started by formally developing a specific approach to preferred answer sets semantics that is situated "between" the approaches of Delgrande et al. (2000b) and that of Brewka and Eiter (1999). This approach can be seen as a refinement of the former approach in that it allows to suspend preferences whenever the result of applying a preferred rule has already been derived. This feature avoids the overly strict prescriptive approach to preferences pursued in (Delgrande et al. 2000b), which may lead to the loss of answer sets.

Acknowledgements. This work was supported by the German Science Foundation (DFG) under grant FOR 375/1-1, TP C. We are grateful to the anonymous referees, although we were unable to follow all suggestions due to severe space restrictions.

7 Proofs

Proof 1 It can be directly verified from the definition of $\mathcal{T}_{(\Pi,<),Y}$. □

Proof 2

1. $\mathcal{C}_{(\Pi,<)}(X) \subseteq C_P(X)$: Since $\mathcal{C}_{(\Pi,<)}(X) = \bigcup_{i\geq 0} \mathcal{T}^i_{(\Pi,<),X}\emptyset$ and $C_\Pi(X) = T^i_{\Pi^X}\emptyset$, we need only to prove that $\mathcal{T}^i_{(\Pi,<),X}\emptyset \subseteq T^i_{\Pi,X}\emptyset$ for $i \geq 0$ by using induction on i.

   Base For $i = 0$, it is obvious that $\mathcal{T}^0_{(\Pi,<),X}\emptyset = \emptyset \subseteq T^0_{\Pi,X}\emptyset$.

   Step Assume that $\mathcal{T}^i_{(\Pi,<),X}\emptyset \subseteq T^i_{\Pi,X}\emptyset$, we want to show that $\mathcal{T}^{i+1}_{(\Pi,<),X}\emptyset \subseteq T^{i+1}_{\Pi,X}\emptyset$. In fact, if $L \in \mathcal{T}^{i+1}_{(\Pi,<),X}\emptyset$, then, by Definition 1, there is a rule r in Π such that $L = \mathit{head}(r)$, $\mathit{body}^+(r) \subseteq \mathcal{T}^i_{(\Pi,<),X}\emptyset$ and $\mathit{body}^-(r) \cap X = \emptyset$. By induction assumption, $\mathit{body}^+(r) \subseteq T^i_{\Pi,X}\emptyset$. Since the rule $L \leftarrow \mathit{body}^+(r)$ is in the reduct program P^X, $L \in T^{i+1}_{\Pi,X}\emptyset$.

2. $C_P(X) \subseteq \mathcal{C}_{(\Pi,<)}(X)$ if $X \subseteq \mathcal{C}_{(\Pi,<)}(X)$: For simplicity, we denote $T_i = T^i_{\Pi,X}\emptyset$ and $X_i = \mathcal{T}^{\Pi}_{(<,X),\emptyset}$ for $i \geq 0$. It suffices to prove $T^i_{\Pi,X}\emptyset \subseteq \mathcal{C}_{(\Pi,<)}(X)$ for $k \geq 0$ by using induction on k. That is, for each $i \geq 0$, there is $n_i \geq 0$ such that $T_i \subseteq X_{n_i}$

   Base If $k = 1$, it is obvious that $T^0_{\Pi,X}\emptyset = \emptyset \subseteq X_0$.

   Step Assume that $T_i \subseteq X_{n_i}$. We want to show $T_{i+1} \subseteq X_{n_{i+1}}$. Let $a \in T_{i+1}$, then there is a rule $r \in \Gamma$ with $\mathit{head}(r) = a$, $\mathit{body}^+(r) \subseteq T_i$ and $\mathit{body}^-(r) \cap X = \emptyset$.

By the induction assumption, r is active wrt (X_{n_i}, X). We claim that there will be no rule r' such that both of Condition I and II hold wrt (X_{n_i}, X). Otherwise, suppose that there is a rule r' such that $head(r') \notin X_{n_i}$, $r < r'$ and r' is active wrt (X, X_{n_i}). Without loss of generality, there is no rule r'' such that $head(r'') \notin X_{n_i}$, $r < r'' < r'$ and r'' is active wrt (X, X_{n_i}). Since $X \subseteq \mathcal{C}_{(\Pi,<)}(X)$, there be a number $n \geq n_i$ such that r' is active wrt (X_n, X). By the assumption of r'', it should be that $head(r'') \in X_n$. A contradiction. Therefore, $head(r) \in X_{n_i+1}$.

3. If $<$ is empty, then the condition II in Definition 1 is automatically satisfied because, for any rule $r \in \Pi$, there is no rule r' that is preferred to r. This implies that $\mathcal{T}^i_{(\Pi,<),X}\emptyset = T^i_{\Pi,X}\emptyset$ for any $i \geq 0$. Therefore, $\mathcal{C}_{(\Pi,<)}(X) = C_P(X)$. □

Proof 5 If $X \subseteq X'$, it is a direct induction on i to show that $\mathcal{T}^i_{(\Pi,<),X'}\emptyset \subseteq \mathcal{T}^i_{(\Pi,<),X}\emptyset$. □

Proof 6

If Π has no consistent answer set, the conclusion is obvious. Thus, we assume that X is consistent. First, we can easily generalize the notion of generating rules as follows: For any two sets Y_1 and Y_2 of literals, set $\Gamma(Y_1, Y_2) = \{head(r) \leftarrow body^+(r) \mid body^+(r) \subseteq Y_1, body^-(r) \cap Y_2 = \emptyset\}$.

Since X is an answer set of Π, we have $X = C_\Pi(X) = \bigcup_{i\geq 0} T^i_{\Pi,X}\emptyset$. Let $\Gamma_0 = \Gamma(T_{\Pi^X}\emptyset, X)$ and $\Gamma_{k+1} = \Gamma(T^k_{\Pi^X}\emptyset, X) - \Gamma_k$ for $k \geq 1$. Define a total order $\ll_X$ on Π such that the following requirements are satisfied:

1. $r' \ll_X r$ for any $r \in \Gamma_k$ and $r' \in \Gamma_{k+1}$, $k = 0, 1, \ldots$.
2. If $r \in \cup_{n\geq 0}\Gamma_n$ and $r' \notin \cup_{n\geq 0}\Gamma_n$, then $r' \ll_X r$.

Since $\Gamma_k \cap \Gamma_{k'} \neq \emptyset$ for $n \neq n'$, such an ordering exists. Denote $X_i = \mathcal{T}^i_{(\Pi,\ll_X),X}\emptyset$. We need only to prove the following two propositions P1 and P2:

P1 X is a prioritized answer set of $(\Pi, \ll_X)$: Since $C_P(X) = X$, it suffices to prove that $\mathcal{C}_{(\Pi,<)}(X) = C_P(X)$. Firstly, by Theorem 2, $\mathcal{C}_{(\Pi,<)}(X) \subseteq C_P(X)$. For the opposite inclusion, we note that $C_P(X) = head(\cup_{k\geq 0}\Gamma_k))$, where $head(\cup_{k\geq 0}\Gamma_k)) = \{head(r) \mid r \in \cup_{k\geq 0}\Gamma_k\}$. Hence, we need only to prove that $head(\Gamma_k) \subseteq \mathcal{C}_{(\Pi,<)}(X)$ for any $k \geq 0$ by using induction on k.

Base For $k = 0$, without loss of generality, suppose that $\Gamma_0 = \{r_1, \ldots, r_t\}$ and $r_t \ll_X \cdots \ll_X r_1$. We use second induction to show that $head(r_i) \in C_P(X)$ for $1 \leq i \leq t$.

Base For $i = 1$, since there is no rule r' with $r_1 \ll_X r'$, $head(r_1) \in X_1$.

Step Assume that $head(r_i) \in X_i$, then $head(r_{i+1}) \in X_{i+1}$. Thus $head(\Gamma_0) \subseteq X_t$.

Step Assume that $head(\Gamma_k) \subseteq \mathcal{C}_{(\Pi,<)}(X)$. Then $head(\Gamma_k) \in X_{m_k}$ for some $m_k > 0$. Let $\Gamma_{k+1} = \{r_1, \ldots, r_u\}$ and $r_u \ll_X \cdots \ll_X r_1$. Then, similar to the case of $k = 0$, we have that $head(r_i) \in X_{m_k+i}$ for $i = 1, \ldots, u$.
Thus, $head(\Gamma_k) \subseteq \mathcal{C}_{(\Pi,<)}(X)$ for any $k \geq 0$.

This implies that $C_P(X) \subseteq \mathcal{C}_{(\Pi,<)}(X)$. Therefore, $\mathcal{C}_{(\Pi,<)}(X) = X$.

P2 If X' is an answer set of Π such that $X' \neq X$, then X' is not a prioritized answer set of $(\Pi, \ll_X)$: First note that $X \setminus X' \neq \emptyset$ and $X' \setminus X \neq \emptyset$. We assert that there is literal $l \in X \setminus X'$ such that $l \notin \mathcal{C}_{(\Pi,<)}(X')$: otherwise, $X \setminus X' \subseteq \mathcal{C}_{(\Pi,<)}(X')$. We can choose $t \geq 0$ and a literal $l_0 \in X \setminus X'$ such that $X'_t \subseteq X \cap X'$ and $l_0 \in X'_{t+1}$. Then there is a rule r such that $head(r) = l_0$, $body^+(r) \subseteq X'_t$ and $body^-(r) \cap X' = \emptyset$. This will implies that $l \in C_{\Pi^{X'}}(X')$, i. e. $l \in X'$, contradiction. Therefore, we have shown that there is a rule r in Π such that $head(r) \in X$ and $head(r) \notin \mathcal{C}_{(\Pi,<)}(X')$. For each $l' \in X' \setminus X$ and each rule r' such that $head(r') = l'$, we have $r' \ll_X r$. Thus, we know that $l' \notin \mathcal{C}_{(\Pi,<)}(X')$. This means that $X' \neq \mathcal{C}_{(\Pi,<)}(X')$ and thus, X' is not a prioritized answer set of $(\Pi, \ll_X)$. $\Box$

Proof 7

On the contrary, suppose that $(\Pi, <)$ has two distinct prioritized answer sets X and X'. Since $X \setminus X' \neq \emptyset$ and $X' \setminus X \neq \emptyset$, there are literals l and l' such that $l \in X \setminus X'$ and $l' \in X' \setminus X$. Without loss of generality, assume that $\mathcal{T}^i_{(\Pi,<),X}\emptyset = \mathcal{T}^i_{(\Pi,<),X'}\emptyset$ for $i \leq n$ but $l \in \mathcal{T}^{n+1}_{(\Pi,<),X}\emptyset$ and $l' \in \mathcal{T}^{n+1}_{(\Pi,<),X'}\emptyset$. This means that there are two rules r and r' such that $head(r) = l$, $head(r') = l'$, r and r' satisfy the two conditions I and II in Definition 1 at stage n with respect to X and X', respectively. We observe two obvious facts: F1. r' is active wrt $(X, \mathcal{T}^n_{(\Pi,<),X}\emptyset)$; and F2. r is active wrt $(X', \mathcal{T}^n_{(\Pi,<),X'}\emptyset)$. By F1, we have $r' \ll r$. Similarly, by F2, it should be $r \ll r'$, contradiction. Therefore, $(\Pi, \ll)$ has the unique prioritized answer sets. $\Box$

Proof 8

1. $X^\star = M_t$ is a prioritized answer set of $(\Pi, <_s)$: $X^\star = \mathcal{C}_{(\Pi,<_s)}(X^\star)$.
   (a) $\mathcal{C}_{(\Pi,<_s)}(X^\star) \subseteq X^\star$: we show that $\mathcal{T}^i_{(\Pi,<_s),X^\star}\emptyset \subseteq X^\star$ by using induction on i.

   Base For $i = 0$, $\mathcal{T}^0_{(\Pi,<_s),X^\star}\emptyset = \emptyset \subseteq X^\star$ is obvious.

   Step Assume that $\mathcal{T}^i_{(\Pi,<_s),X^\star}\emptyset \subseteq X^\star$. If $p \in \mathcal{T}^{i+1}_{(\Pi,<_s),X^\star}\emptyset$, then there is a rule r in Π such that $p = head(r)$, $body^+(r) \subseteq \mathcal{T}^i_{(\Pi,<_s),X^\star}\emptyset$ and $body^-(r) \cap X^\star = \emptyset$. By induction assumption, $body^+(r) \subseteq X^\star$. If $r \in \Pi_j$, then $body^+(r) \subseteq M_j$ and $body^-(r) \cap M_{j-1} = \emptyset$. Therefore, $p \in X^\star$. That is, $\mathcal{T}^{i+1}_{(\Pi,<_s),X^\star}\emptyset \subseteq X^\star$.

   (b) $X^\star \subseteq \mathcal{C}_{(\Pi,<_s)}(X^\star)$: we show that $M_i \subseteq \mathcal{C}_{(\Pi,<_s)}(X^\star)$ for $0 \leq i \leq t$.

   Base For $i = 1$, it is obvious since $M_0 = \emptyset$.

   Step If we have shown $M_i \subseteq \mathcal{C}_{(\Pi,<_s)}(X^\star)$, we want to show that $M_{i+1} \subseteq \mathcal{C}_{(\Pi,<_s)}(X^\star)$. We again use second induction on k to prove that if $p \in T^k_{\Pi_{i+1},M_i}M_i$, then $p \in \mathcal{C}_{(\Pi,<_s)}(X^\star)$:

   Base For $k = 1$, i. e. $p \in T^1_{\Pi_{i+1},M_i}M_i$, if $p \notin M_i$, then there is a rule r in Π_{i+1} such that $p = head(r)$, $body^+(r) = \emptyset$ and $body^-(r) \cap M_i = \emptyset$. Then $body^-(r) \cap X^\star = \emptyset$.

   By the first induction assumption, $M_i \subseteq \mathcal{T}^{j_0}_{(\Pi,<_s),X^\star}\emptyset$ for some j_0. If there are $j > 0$ and a rule r' such that $r <_s r'$ and r' is active with respect to $(X^\star, \mathcal{T}^j_{(\Pi,<_s),X^\star}\emptyset)$ and $head(r') \notin \mathcal{T}^j_{(\Pi,<_s),X^\star}\emptyset$. Then, $body^+(r') \subseteq X^\star$ and $body^-(r') \cap \mathcal{T}^j_{(\Pi,<_s),X^\star}\emptyset = \emptyset$. We assert that $j \leq j_0$. Otherwise, if $j > j_0$, $body^-(r') \cap \mathcal{T}^j_{(\Pi,<_s),X^\star}\emptyset = \emptyset \Rightarrow body^-(r') \cap \mathcal{T}^{j_0}_{(\Pi,<_s),X^\star}\emptyset = \emptyset \Rightarrow body^-(r') \cap M_i = \emptyset \Rightarrow body^-(r') \cap X^\star = \emptyset$. Therefore, $head(r') \in M_i \subseteq \mathcal{T}^j_{(\Pi,<_s),X^\star}\emptyset$, a contradiction. Thus, when $j > j_0$, there will be no rule in Π that prevents r to be included in $\mathcal{T}^j_{(\Pi,<_s),X^\star}\emptyset$. Thus, $p \in \mathcal{C}_{(\Pi,<_s)}(X^\star)$.

Step Assume that $p \in \mathcal{C}_{(\Pi,<_s)}(X^\star)$ if $p \in T^k_{\Pi_{i+1},M_i} M_i$. Suppose that $p \in T^{k+1}_{\Pi_{i+1},M_i} M_i$ but $p \notin M_i$, then there is a rule r in Π_{i+1} such that $p = head(r)$, $body^+(r) \subseteq T^k_{\Pi_{i+1},M_i}\emptyset$ and $body^-(r) \cap M_i = \emptyset$. Then $body^+(r) \subseteq M_i \subseteq \mathcal{T}^{j_0}_{(\Pi,<_s),X^\star}\emptyset$ for some j_0 and $body^-(r) \cap X^\star = \emptyset$. Similar to the proof of the case $k = 1$, we can also prove that $p \in \mathcal{C}_{(\Pi,<_s)}(X^\star)$.

2. If $X = \mathcal{C}_{(\Pi,<_s)}(X)$, then X is a preferred answer set of $(\Pi,<_s)$. By Corollary 3, X is also an answer set of Π. However, Π has the unique answer set $X^\star$ and thus $X = X^\star$. $\square$

Proof 9
By Theorem 8 (1), the perfect model $X^\star$ is a preferred answer set. On the other hand, since each preferred answer set X is also a standard answer set. In particular, for the stratified program Π, it has the unique answer set $X^\star$. Therefore, $X = X^\star$. $\square$

Proof 11
Let $(\Pi,<)$ be an ordered logic program over $\mathcal{L}$ and X a consistent set of literals over $\mathcal{L}$.

"⊆"-part Define[8]

$$\begin{aligned} Y \quad = \quad & \{head(r) \mid r \in rule(C_{\mathbb{T}(\Pi,<)}(Y))\} \\ & \cup \{\mathsf{ap}(n_r) \mid r \in rule(C_{\mathbb{T}(\Pi,<)}(Y))\} \cup \{\mathsf{bl}(n_r) \mid r \notin rule(C_{\mathbb{T}(\Pi,<)}(Y))\} \\ & \cup \{\mathsf{ok}(n_r) \mid r \in \Pi\} \cup \{\mathsf{rdy}(n_r, n_{r'}) \mid r, r' \in \Pi\} \end{aligned}$$

Clearly, we have $X = Y \cap \mathcal{L}$. By definition, we have $\mathcal{C}_{(\Pi,<)}(X) = \bigcup_{i\geq 0} \mathcal{T}^i_{(\Pi,<),X}\emptyset$ and $C_{\mathbb{T}(\Pi,<)}(Y) = Cn(\mathbb{T}(\Pi,<)^Y)$.

In view of this, we show by induction that $\mathcal{T}^i_{(\Pi,<),X}\emptyset \subseteq Cn(\mathbb{T}(\Pi,<)^Y)$ for $i \geq 0$. To be precise, we show for every $r \in \Pi$ by nested induction that $head(r) \in \mathcal{T}^i_{(\Pi,<),X}\emptyset$ implies $head(r) \in Cn(\mathbb{T}(\Pi,<)^Y)$ and moreover, for every $r' \in \Pi$, that if $r < r'$ then $\mathsf{bl}(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$ or $\mathsf{ap}(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$ or $head(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$.

$i = 0$ By definition, $\mathcal{T}^0_{(\Pi,<),X}\emptyset = \emptyset \subseteq Cn(\mathbb{T}(\Pi,<)^Y)$.

$i > 0$ Consider $r \in \Pi$ such that $head(r) \in \mathcal{T}^{i+1}_{(\Pi,<),X}\emptyset$. By definition, we have that r is active wrt $(\mathcal{T}^i_{(\Pi,<),X}\emptyset, X)$. That is,

1. $body^+(r) \subseteq \mathcal{T}^i_{(\Pi,<),X}\emptyset$. By the induction hypothesis, we get $body^+(r) \subseteq Cn(\mathbb{T}(\Pi,<)^Y)$.
2. $body^-(r) \cap X = \emptyset$. By definition of Y, this implies $body^-(r) \cap Y = \emptyset$.
   Furthermore, this implies that $a_2(r)^+ = \mathsf{ap}(n_r) \leftarrow \mathsf{ok}(n_r), body^+(r) \in \mathbb{T}(\Pi,<)^Y$.

We proceed by induction on $<$.

Base Suppose r is maximal with respect to $<$. We can show the following lemma.

[8] As defined in Section 4, $rule(\cdot)$ is a bijective mapping between rule heads and rules.

Lemma 7.1
If $r \in \Pi$ is maximal with respect to $<$, then $\mathsf{ok}(n_r) \in Cn(\mathbb{T}(\Pi,<)^Y)$.

Given that we have just shown in 1 and 2 that $body^+(r) \subseteq Cn(\mathbb{T}(\Pi,<)^Y)$ and $a_2(r)^+ \in \mathbb{T}(\Pi,<)^Y$, Lemma 7.1 and the fact that $Cn(\mathbb{T}(\Pi,<)^Y)$ is closed under $\mathbb{T}(\Pi,<)^Y$ imply that $\mathsf{ap}(n_r) \in Cn(\mathbb{T}(\Pi,<)^Y)$. Analogously, we get $head(r) \in Cn(\mathbb{T}(\Pi,<)^Y)$ due to $a_1(r)^+ \in \mathbb{T}(\Pi,<)^Y$. We have thus shown that $\{head(r), \mathsf{ap}(n_r)\} \subseteq Cn(\mathbb{T}(\Pi,<)^Y)$.

Step We start by showing the following auxiliary result.

Lemma 7.2
Given the induction hypothesis, we have $\mathsf{ok}(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$.

Proof 7.2
Consider $r'' \in \Pi$ such that $r' < r''$. By the induction hypothesis, we have either $\mathsf{bl}(n_{r''}) \in Cn(\mathbb{T}(\Pi,<)^Y)$ or $\mathsf{ap}(n_{r''}) \in Cn(\mathbb{T}(\Pi,<)^Y)$ or $head(n_{r''}) \in Cn(\mathbb{T}(\Pi,<)^Y)$. Clearly, we have $(n_{r'} \prec n_{r''}) \in Cn(\mathbb{T}(\Pi,<)^Y)$ iff $r' < r''$. Hence, whenever $r' < r''$, we obtain $\mathsf{rdy}(n_{r'}, n_{r''}) \in Cn(\mathbb{T}(\Pi,<)^Y)$ by means of $c_3(r', r'')^+$, $c_4(r', r'')^+$, or $c_5(r', r'')^+$ (all of which belong to $\mathbb{T}(\Pi,<)^Y$). Similarly, we get $\mathsf{rdy}(n_{r'}, n_{r''}) \in Cn(\mathbb{T}(\Pi,<)^Y)$, whenever $r' \not< r''$ from $c_2(r', r'')^+$. Lastly, we obtain $\mathsf{ok}(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$ via $c_1(r')^+ \in \mathbb{T}(\Pi,<)^Y$. $\square$

For all rules r' with $r < r'$, we have that either

1. r' is not active wrt $(X, \mathcal{T}^i_{(\Pi,<),X}\emptyset)$. That is, we have that either
   (a) $body^+(r) \not\subseteq X$. By definition of Y, this implies $body^+(r) \not\subseteq Y$.
   By definition, $b_1(r', L^+)^+ = \mathsf{bl}(n_{r'}) \leftarrow \mathsf{ok}(n_{r'}) \in \mathbb{T}(\Pi,<)^Y$ for some $L^+ \in body^+(r)$ such that $L^+ \notin Y$. By Lemma 7.2, we have $\mathsf{ok}(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$. Given that $Cn(\mathbb{T}(\Pi,<)^Y)$ is closed under $\mathbb{T}(\Pi,<)^Y$, we get that $\mathsf{bl}(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$.
   (b) $body^-(r) \cap \mathcal{T}^i_{(\Pi,<),X}\emptyset \neq \emptyset$. By the induction hypothesis, this implies that $body^-(r) \cap Cn(\mathbb{T}(\Pi,<)^Y) \neq \emptyset$.
   Therefore, $b_2(r, L^-)^+ = \mathsf{bl}(n_r) \leftarrow \mathsf{ok}(n_r), L^- \in \mathbb{T}(\Pi,<)^Y$ for some $L^- \in body^-(r) \cap Cn(\mathbb{T}(\Pi,<)^Y)$. In analogy to 1a, this allows us to conclude that $\mathsf{bl}(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$.

   In both cases, we conclude $\mathsf{bl}(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$. By the induction assumption, $head(r') \in Cn(\mathbb{T}(\Pi,<)^Y)$.

We have thus shown that either $\mathsf{bl}(n_{r'}) \in Cn(\mathbb{T}(\Pi,<)^Y)$ or $head(r') \in Cn(\mathbb{T}(\Pi,<)^Y)$ for all r' such that $r < r'$.
In analogy to what we have shown in the proof of Lemma 7.2, we can now show that $\mathsf{ok}(n_r) \in Cn(\mathbb{T}(\Pi,<)^Y)$.
In analogy to the base case, we may then conclude $\{head(r), \mathsf{ap}(n_r)\} \subseteq Cn(\mathbb{T}(\Pi,<)^Y)$.

"$\supseteq$"-part We have $X = Y \cap \mathcal{L}$. By definition, we have $C_{\mathbb{T}(\Pi,<)}(Y) = Cn(\mathbb{T}(\Pi,<)^Y)$ and moreover that $Cn(\mathbb{T}(\Pi,<)^Y) = \bigcup_{i \geq 0} T^i_{\mathbb{T}(\Pi,<)^Y}\emptyset$. Given this, we show by induction that $(T^i_{\mathbb{T}(\Pi,<)^Y}\emptyset \cap \mathcal{L}) \subseteq \mathcal{C}_{(\Pi,<)}(X)$ for $i \geq 0$.

$i = 0$ By definition, $T^0_{\mathbb{T}(\Pi,<)^Y}\emptyset = \emptyset \subseteq \mathcal{C}_{(\Pi,<)}(X)$.

$i > 0$ Consider $r \in \Pi$ such that $head(r) \in (T^{i+1}_{\mathbb{T}(\Pi,<)^Y}\emptyset \cap \mathcal{L})$. In view of $\mathbb{T}(\Pi,<)^Y$, this implies that $\mathsf{ap}(n_r) \in (T^{i}_{\mathbb{T}(\Pi,<)^Y}\emptyset \cap \mathcal{L})$ and thus $a_2(r)^+ \in \mathbb{T}(\Pi,<)^Y$. The latter implies that $body^-(r) \cap Y = \emptyset$, whence $body^-(r) \cap X = \emptyset$ because of $X = Y \cap \mathcal{L}$. The former implies that $body^+(r) \cup \{\mathsf{ok}(n_r)\} \subseteq T^{i-1}_{\mathbb{T}(\Pi,<)^Y}\emptyset$. By the induction hypothesis, we obtain that $body^+(r) \subseteq \mathcal{C}_{(\Pi,<)}(X)$. Consequently, r is active wrt $(\mathcal{C}_{(\Pi,<)}(X), X)$.

Suppose there is some $r' \in \Pi$ with $r < r'$ such that

1. r' is active wrt $(X, \mathcal{C}_{(\Pi,<)}(X))$. That is,
   (a) $body^+(r) \subseteq X$ and
   (b) $body^-(r) \cap \mathcal{C}_{(\Pi,<)}(X) = \emptyset$.
2. $head(r') \notin \mathcal{C}_{(\Pi,<)}(X)$.

By the induction hypothesis, we obtain from 2 that $head(r') \notin T^{j}_{\mathbb{T}(\Pi,<)^Y}\emptyset$ for $j \leq i$.

Clearly, we have $(n_{r'} \prec n_{r''}) \in T^{i}_{\mathbb{T}(\Pi,<)^Y}\emptyset$ for $i \geq 1$ iff $r' < r''$. Moreover, $\mathsf{ok}(n_r) \in T^{i-1}_{\mathbb{T}(\Pi,<)^Y}\emptyset$ implies (see above) $\mathsf{rdy}(n_r, n_{r''}) \in T^{i-2}_{\mathbb{T}(\Pi,<)^Y}\emptyset$ for all $r'' \in \Pi$. This and the fact that $head(r') \notin T^{j}_{\mathbb{T}(\Pi,<)^Y}\emptyset$ for $j \leq i$ implies that $\mathsf{bl}(n_{r'}) \in T^{i-3}_{\mathbb{T}(\Pi,<)^Y}\emptyset$.

This makes us distinguish the following two cases.

1. If $\mathsf{bl}(n_{r'})$ is provided by $b_1(r', L^+)$, then there is some $L^+ \in body^+(r')$ such that $L^+ \notin Y$. Given that $X = Y \cap \mathcal{L}$, this contradicts 1a.
2. If $\mathsf{bl}(n_{r'})$ is provided by $b_2(r', L^-)$, then there is some $L^- \in body^-(r')$ such that $L^- \in T^{i-4}_{\mathbb{T}(\Pi,<)^Y}\emptyset$. By the induction hypothesis, we obtain that $L^- \in \mathcal{C}_{(\Pi,<)}(X)$. A contradiction to 1b.

So, given that r is active wrt $(\mathcal{C}_{(\Pi,<)}(X), X)$ and that there is no $r' \in \Pi$ such that $r < r'$ satisfying 1a, 1b, and 2, we have that $head(r) \in \mathcal{T}_{(\Pi,<),X}(\mathcal{C}_{(\Pi,<)}(X))$. That is, $head(r) \in \mathcal{C}_{(\Pi,<)}(X)$. □

Proof 12 It follows from Lemma 7.7 and Lemma 7.8. □

Proof 13 Similar to the proof of Theorem 8. □

By Theorem 4.8 in (Delgrande et al. 2003), it suffices to show the following Lemma 7.4. Before doing this, we first present a definition. Given a statically ordered logic program $(\Pi, <)$ and a set X of literals, set $X_i = (\mathcal{T}^{\text{D}})^i_{(\Pi,<),X}\emptyset$ for $i \geq 0$.

Definition 16
Let $(\Pi, <)$ be a statically ordered logic program and r be a rule in Π. X and $X_i (i \geq 1)$ are as above. We say another rule r' is a D-*preventer* of r in the context (X, X_i) if (1) $r < r'$ and (2) r' is active wrt (X, X_i) and $r' \notin rule(X_i)$.

Lemma 7.3
Let $(\Pi, <)$ be a statically ordered logic program and X a set of literals with $\mathcal{C}^{\text{D}}_{(\Pi,<)}(X) = X$. Then, for any $r \in \Gamma_\Pi X$, there exists a number i such that $r \in rule(X_i)$.

The intuition behind this lemma is that each D-preventer of a rule in $\Gamma_\Pi X$ is a "temporary" one if $\mathcal{C}^{\text{D}}_{(\Pi,<)}(X) = X$.

Proof 7.3

On the contrary, suppose that there is a rule $r \in \Gamma_\Pi X$ such that $r \notin rule(X_i)$ for any i. Without loss of generality, assume that there is no such rule that is preferred than r.

Since $r \in \Gamma_\Pi X$ and $X = \cup_{i=1}^{\infty} X_i$, r will become active wrt (X_t, X) at some stage $t \geq 0$. Therefore, it must be the case that there is a D-preventer r' satisfying $r' \in \Gamma_\Pi X$. This implies that $r < r'$ and $r' \in \Gamma_\Pi X$ but $r' \notin rule(X_i)$ for any i, contradiction to our assumption on r. Thus, the lemma is proven. □

Lemma 7.4

Let $(\Pi, <)$ be a statically ordered logic program and X a set of literals. Then X is a $<^{\mathrm{D}}$-preserving answer set ofΠ if and only if X is a set of literals with $\mathcal{C}^{\mathrm{D}}_{(\Pi,<)}(X) = X$.

Proof 7.4

Without loss of generality, assume that $rule(X_i) = \{r_{i1}, \ldots, r_{in_i}\}$ for $i \geq 1$.

if part Let $\mathcal{C}^{\mathrm{D}}_{(\Pi,<)}(X) = X$. By Lemma 7.3, $\Gamma_\Pi X = \cup_{i=1}^{\infty} rule(X_i)$. This means that the sequence Δ: $\langle r_{11}, \ldots, r_{1n_1}, r_{21}, \ldots, r_{2n_2}, \ldots \rangle$ is an enumeration of $\Gamma_\Pi X$.

It suffices to prove that this sequence of rules in Δ is $<^{\mathrm{D}}$-preserving with respect to X.

We need to justify the two conditions of $<^{\mathrm{D}}$-sequence are satisfied by Δ:

C1 For each $r_i \in rule(X_t)$ where $t > 0$, then r_i is active wrt (X_{t-1}, X). This implies that $body^+(r_i) \subseteq \{head(r_j) \mid j < i\}$.

C2 if $r < r'$, then r' is prior to r in Δ: notice that, since $X = \cup_{i=1}^{\infty} X_i$, if a rule is active wrt (X_i, X) then it is also active wrt (X, X_i). Thus, by Definition 1, r and r' can not be in the same section $rule(X_s)$. If C2 is not satisfied by Δ, then there are two rules, say r and r', such that $r < r'$ but r is prior to r' in Δ. Without loss of generality, assume that $r \in rule(X_i)$ and $r' \in rule(X_j)$ but $i < j$. Then r' should prevent r to be included in $rule(X_i)$, which means $r \notin rule(X_i)$, contradiction. Therefore, C2 holds.

C3 if $r_i < r'$ and $r' \in \Pi \setminus \Gamma_\Pi X$, then $body^+(r') \not\subseteq X$ or r' is defeated by the set $\{head(r_j) \mid j < i\}$: Assume that $r_i \in rule(X_t)$, then $r' \notin rule(X_t)$. On the contrary, assume that $body^+(r') \subseteq X$ and r' is not defeated by the set $\{head(r_j) \mid j < i\}$, then r' is not defeated by X_{t-1} because $X_{t-1} \subseteq \{head(r_j) \mid j < i\}$. Thus, r' is active wrt (X, X_{t-1}) and $r' \notin rule(X_{t-1})$. This means that r' is a D-preventer of r_i in the context (X, X_{t-1}) and thus, $r_i \notin rule(X_t)$, contradiction. That is, $body^+(r') \not\subseteq X$ or r' is defeated by the set $\{head(r_j) \mid j < i\}$.

only-if part Assume that X is a $<^{\mathrm{D}}$-preserving answer set of Π, then there is a grounded enumeration $\langle r_i \rangle_{i \in I}$ of $\Gamma_\Pi X$ such that, for every $i, j \in I$, we have that:

1. if $r_i < r_j$, then $j < i$; and
2. if $r_i < r'$ and $r' \in \Pi \setminus \Gamma_\Pi X$, then either (a) $body^+(r') \not\subseteq X$ or (b) $body^-(r') \cap \{head(r_j) \mid j < i\} \neq \emptyset$.

A set $\bar{\Delta}$ of rules is *discrete* if there is no pair of rules r and r' in $\bar{\Delta}$ s. t. $r < r'$.

We define recursively a sequence of sets of rules as follows.

Define $\bar{\Delta}_1$ as the largest section of $\langle r_i \rangle_{i \in I}$ satisfying the following conditions:

1. $\bar{\Delta}_1$ is discrete;
2. $r_1 \in \bar{\Delta}_1$;
3. $body(r) = \emptyset$ for any $r \in \bar{\Delta}_1$.

Suppose that $\bar{\Delta}_i$ is well-defined and r_{m_i} is the last rule of $\bar{\Delta}_i$, we define $\bar{\Delta}_{i+1}$ as the largest section of $\langle r_i \rangle_{i \in I}$ satisfying the following conditions:

1. $\bar{\Delta}_{i+1}$ is discrete;
2. $r_{m_i+1} \in \bar{\Delta}_{i+1}$;
3. $body(r) \subseteq \{head(r') \mid r' \in \cup_{j=0}^{i} \bar{\Delta}_j\}$ for any $r \in \bar{\Delta}_{i+1}$.
4. disjoint with $\cup_{j=0}^{i} \bar{\Delta}_j$.

Denote $\bar{X}_i = \{head(r) \mid r \in \cup_{j=0}^{i} \bar{\Delta}_j\}$. Then we have the following fact:

if $r \in \bar{\Delta}_{i+1}$ such that $\bar{X}_{i-1} \models body^+(r)$ and no rule $r' \in \bar{\Delta}_i$ with $r < r'$, then we can move r from $\bar{\Delta}_{i+1}$ to $\bar{\Delta}_i$, the resulting sequence of rules still is $<^{\mathrm{D}}$-preserving.

Without loss of generality, assume that our sequence $\langle \bar{\Delta}_i \rangle$ is fully transformed by the above transformation. Since $\cup_{i=0}^{\infty} \bar{X}_i = X$, we can prove $\cup_{i=0}^{\infty} X_i = X$ by proving $\bar{X}_i = X_i$ for every $i \in I$. Thus, it suffices to prove $\bar{\Delta}_i = rule(X_i)$ for every $i \in I$. We use induction on i:

Base $\bar{\Delta}_0 = rule(X_0) = \emptyset$.

Step Assume that $\bar{\Delta}_i = rule(X_i)$, we want to prove $\bar{\Delta}_{i+1} = rule(X_{i+1})$.

$\bar{\Delta}_{i+1} \subseteq rule(X_{i+1})$: For any $r_t \in \bar{\Delta}_{i+1}$, by the condition 3 in the construction of $\bar{\Delta}_{i+1}$, r_t is active wrt (X_i, X). And for any r' such that $r_t < r'$ and r' is active wrt (X, X_i) then $body^+(r') \subseteq X$ and r' is not defeated by X_i. By induction, $\cup_{k<t} head(r_k) \subseteq X_i = \bar{X}_i$, thus r' is not defeated by $\cup_{k<t} head(r_k)$. By Definition 5, it should be the case that $r' \in \Gamma_\Pi X$, which implies that $r' \in \bar{\Delta}_i = rule(X_i)$. Therefore, r' is not a D-preventer of r_t. That is, $r_t \in rule(X_{i+1})$.

$rule(X_{i+1}) \subseteq \bar{\Delta}_{i+1}$: For $r \in rule(X_{i+1})$, we claim that $r \in \bar{\Delta}_{i+1}$. Otherwise, there would exist $t > i+1$ such that $r \in \bar{\Delta}_t$. Notice that, by induction assumption, $body^+(r) \subseteq X_i$. Thus, it must be the case that there is at least one rule $r' \in \cup_{j=i+1}^{t-1} rule(X_j)$ such that $r' < r$. But r' is active wrt (X, X_{i+1}), which contradicts to $r \in rule(X_{i+1})$. Therefore, $rule(X_{i+1}) \subseteq \bar{\Delta}_{i+1}$. □

Proof 15 Similar to the proof of Theorem 11. □

Proof 16 It follows from the following Lemma 7.7 and Theorem 12. □

Lemma 7.5

Let $(\Pi, <)$ be an ordered logic program over $\mathcal{L}$ and let Y be a consistent answer set of $\mathcal{T}^{\mathrm{W}}_{((,\Pi),,}<)$. Denote $X = Y \cap \mathcal{L}$. Then, we have for any $r \in \Pi$:

1. $\mathsf{ok}(n_r) \in Y$; and
2. $\mathsf{ap}(n_r) \in Y$ iff $\mathsf{bl}(n_r) \notin Y$.

Proof 7.5

We prove the two propositions by parallel induction on ordering $<$.

Base Let r be a maximal element of $<$.

1. By assumption, $r \not< r'$ for any $r' \in \Pi$. This implies that $\mathsf{rdy}(n_r, n_{r'}) \in Y$ for any $r' \in \Pi$. Thus, $\mathsf{ok}(n_r) \in Y$.
2. There are two possible cases:
   - $body(r)$ is satisfied by X: Since $a_2(r) \in \mathbb{T}(\Pi,<)^Y$, we have $\mathsf{ap}(n_r) \in Y$.
   - $body(r)$ is not satisfied by X: The body of at least one of $b_1(r, L^+)$ and $b_2(r, L^-)$ is satisfied by Y, thus $\mathsf{bl}(n_r) \in \mathbb{T}(\Pi,<)^Y$.

Step

1. Consider $r \in \Pi$. Assume that $\mathsf{ok}(n_{r'}) \in Y$ and either $\mathsf{ap}(n_{r'}) \in Y$ or $\mathsf{bl}(n_{r'}) \in Y$ for all r' with $r < r'$. In analogy to the base case, we have $\mathsf{rdy}(n_r, n_{r'}) \in Y$ for all $r' \in \Pi$ with $r \not< r'$.
   For r' with $r < r'$, by the induction assumption, we have either $\mathsf{ap}(n_{r'}) \in Y$ or $\mathsf{bl}(n_{r'}) \in Y$. Hence the body of at least one of $c_3(r, r')$ and $c_4(r, r')$ is satisfied by Y. This implies $\mathsf{rdy}(n_r, n_{r'}) \in Y$.
   So, we have proved that $\mathsf{rdy}(n_r, n_{r'}) \in Y$ for any $r' \in \Pi$. Thus, $\mathsf{ok}(n_r) \in Y$.
2. Analogous to the base case. $\square$

== Given a statically ordered logic program $(\Pi,<)$ and a set X of literals, set $X_i = (\mathcal{T}^{\mathsf{W}})^i_{(\Pi,<),X}\emptyset$ for $i \geq 0$ and $ugr(X_i) = \{r \in \Gamma_\Pi X \setminus ugr(X_{i-1}) \mid$ either r applied in producing $X_i \setminus X_{i-1}$ or $head(r) \in X_{i-1}\}$ for $i > 0$. Intuitively, $ugr(X_i)$ is the set of the generating rules that are used at stage i.

Definition 17
Let $(\Pi,<)$ be a statically ordered logic program and r be a rule in Π. X and $X_i (i \geq 0)$ are as above. We say another rule r' is a W-*preventer* of r in the context (X, X_i) if the following conditions are satisfied:

1. $r < r'$ and
2. r' is active wrt (X, X_i) and $head(r') \notin X_i$.

Lemma 7.6
Let $(\Pi,<)$ be a statically ordered logic program and X a set of literals with $\mathcal{C}^{\mathsf{W}}_{(\Pi,<)}(X) = X$. Then, for any $r \in \Gamma_\Pi X$, there exists a number i such that $r \in ugr(X_i)$.

The intuition behind this lemma is that each W-preventer of a rule in $\Gamma_\Pi X$ is a "temporary" one if $\mathcal{C}^{\mathsf{W}}_{(\Pi,<)}(X) = X$.

Proof 7.6
On the contrary, suppose that there is a rule $r \in \Gamma_\Pi X$ such that $r \notin ugr(X_i)$ for any i. Without loss of generality, assume that there is no such rule that is preferred than r. Since $r \in \Gamma_\Pi X$ and $X = \cup_{i=1}^{\infty} X_i$, r will become active wrt (X_t, X) at some stage $t \geq 0$. Therefore, it must be the case that there is a W-preventer r' satisfying $r' \in \Gamma_\Pi X$. This implies that $r < r'$ and $r' \in \Gamma_\Pi X$ but $r' \notin ugr(X_i)$ for any i, contradiction to our assumption on r. Thus, the lemma is proven. $\square$

Lemma 7.7
Let $(\Pi, <)$ be a statically ordered logic program and X a set of literals. Then X is a $<^{\text{W}}$-preserving answer set of Π if and only if X is a set of literals with $\mathcal{C}^{\text{W}}_{(\Pi,<)}(X) = X$.

Proof 7.7
Without loss of generality, assume that $ugr(X_i) = \{r_{i1}, \ldots, r_{in_i}\}$ for $i \geq 1$.

if part Let $\mathcal{C}^{\text{W}}_{(\Pi,<)}(X) = X$. By Lemma 7.6, $\Gamma_\Pi X = \cup_{i=1}^{\infty} ugr(X_i)$. This means that the sequence Δ: $\langle r_{11}, \ldots, r_{1n_1}, r_{21}, \ldots, r_{2n_2}, \ldots \rangle$ is an enumeration of $\Gamma_\Pi X$. It suffices to prove that this sequence is $<^{\text{W}}$-preserving with respect to X.

We need to justify that the three conditions of $<^{\text{W}}$-sequence are satisfied by Δ:

C1 For each $r_i \in \Delta$, either r_i is active wrt (X_t, X) or $head(r_i) \in X_t$ for some $t > 0$. Thus, Condition 1 in Definition 8 is satisfied.

C2 If $r < r'$, then r' is prior to r in Δ: Notice that $X = \cup_{i=1}^{\infty} X_i$, if a rule is active wrt (X_i, X) then it is also active wrt (X, X_i). Thus, by Definition 1, r and r' can not be in the same section $ugr(X_s)$.

If C2 is not satisfied by Δ, then there are two rules, say r and r', such that $r < r'$ but r is prior to r' in Δ. Without loss of generality, assume that $r \in ugr(X_i)$ and $r' \in ugr(X_j)$ but $i < j$. Then r' should prevent r to be included in $ugr(X_i)$, which means $r \notin ugr(X_i)$, contradiction. Therefore, C2 holds.

C3 On the contrary, suppose that Condition 3 in Definition 8 is not satisfied. That is, there are two rules r_i and r' such that $r_i < r'$, $r' \in \Pi \setminus \Gamma_\Pi X$ and the following items hold:

1. $body^+(r') \subseteq X$,
2. r' is not defeated by the set $\{head(r_j) \mid j < i\}$,
3. $head(r') \notin \{head(r_j) \mid j < i\}$.

Without loss of generality, assume that $r_i \in ugr(X_t)$, then r' is not defeated by X_{t-1} because $X_{t-1} \subseteq \{head(r_j) \mid j < i\}$. Thus, r' is active wrt (X, X_{t-1}) and $r' \notin ugr(X_{t-1})$. This means that r' is a W-preventer of r_i in the context (X, X_{t-1}) and thus, $r_i \notin ugr(X_t)$, contradiction.

only-if part Assume that X is a $<^{\text{W}}$-preserving answer set of Π, then there is a grounded enumeration $\langle r_i \rangle_{i \in I}$ of $\Gamma_\Pi X$ such that the three conditions in Definition 8 are all satisfied.

A set $\bar{\Delta}$ of rules is *discrete* if there is no pair of rules r and r' in $\bar{\Delta}$ such that $r < r'$. We define recursively a sequence of sets of rules as follows.

Define $\bar{\Delta}_1$ as the largest section of $\langle r_i \rangle_{i \in I}$ satisfying the following conditions:

1. $\bar{\Delta}_1$ is discrete;
2. $r_1 \in \bar{\Delta}_1$;
3. $body(r) = \emptyset$ for any $r \in \bar{\Delta}_1$.

Suppose that $\bar{\Delta}_i$ is well-defined and r_{m_i} is the last rule of $\bar{\Delta}_i$, we define $\bar{\Delta}_{i+1}$ as the largest section of $\langle r_i \rangle_{i \in I}$ satisfying the following conditions:

1. $\bar{\Delta}_{i+1}$ is discrete;
2. $r_{m_i+1} \in \bar{\Delta}_{i+1}$;

3. Either $body(r) \subseteq \{head(r') \mid r' \in \cup_{j=0}^{i}\bar{\Delta}_j\}$ or $head(r) \in \{head(r') \mid r' \in \cup_{j=0}^{i}\bar{\Delta}_j\}$ for any $r \in \bar{\Delta}_{i+1}$.
4. Disjoint with $\cup_{j=0}^{i}\bar{\Delta}_j$.

Denote $\bar{X}_i = \{head(r) \mid r \in \cup_{j=0}^{i}\bar{\Delta}_j\}$. Then we observe the following fact:

If $r \in \bar{\Delta}_{i+1}$ such that $body^+(r)$ is satisfied by $\bar{X}_{i-1}$ and no rule $r' \in \bar{\Delta}_i$ with $r < r'$, then we can move r from $\bar{\Delta}_{i+1}$ to $\bar{\Delta}_i$, the resulting sequence of rules is still $<^{\mathrm{W}}$-preserving.

Without loss of generality, assume that our sequence $\langle\bar{\Delta}_i\rangle$ is fully transformed by the above transformation. Since $\cup_{i=0}^{\infty}\bar{X}_i = X$, we can prove $\cup_{i=0}^{\infty}X_i = X$ by proving $\bar{X}_i = X_i$ for every $i \in I$. Thus, it suffices to prove $\bar{\Delta}_i = ugr(X_i)$ for every $i \in I$. We use induction on i:

Base $\bar{\Delta}_0 = ugr(X_0) = \emptyset$.

Step Assume that $\bar{\Delta}_i = ugr(X_i)$, we want to prove $\bar{\Delta}_{i+1} = ugr(X_{i+1})$.

1. $\bar{\Delta}_{i+1} \subseteq ugr(X_{i+1})$: For any $r_t \in \bar{\Delta}_{i+1}$, by the condition 3 in the construction of $\bar{\Delta}_{i+1}$, either r_t is active wrt (X_i, X) or $head(r_t) \in X_i$. If $head(r_t) \in X_i$, it is obvious that $r \in ugr(X_{i+1})$. Thus, we assume that r_t is active wrt (X_i, X). For any r' such that $r_t < r'$ and r' is active wrt (X, X_i) then $body^+(r') \subseteq X$ and r' is not defeated by X_i. By induction, $\cup_{k<t}head(r_k) \subseteq X_i = \bar{X}_i$, thus r' is not defeated by $\cup_{k<t}head(r_k)$. By Definition 8, it should be the case that $r' \in \Gamma_\Pi X$, which implies that $r' \in \bar{\Delta}_i = ugr(X_i)$. Therefore, r' is not a W-preventer of r_t in the context of (X_i, X). That is, $r_t \in ugr(X_{i+1})$.
2. $ugr(X_{i+1}) \subseteq \bar{\Delta}_{i+1}$: For $r \in ugr(X_{i+1})$, we claim that $r \in \bar{\Delta}_{i+1}$. Otherwise, there would exist $t > i+1$ such that $r \in \bar{\Delta}_t$. Notice that, by induction assumption, $body^+(r) \subseteq X_i$ (Note that $head(r) \in X_i$ is impossible because we assume that $r \in \bar{\Delta}_t$ and $t > i+1$). Thus, it must be the case that there is at least one rule $r' \in \cup_{j=i+1}^{t-1}ugr(X_j)$ such that $r' < r$. But r' is active wrt (X, X_{i+1}), which contradicts to $r \in ugr(X_{i+1})$. Therefore, $ugr(X_{i+1}) \subseteq \bar{\Delta}_{i+1}$. $\square$

Lemma 7.8
Let $(\Pi, <)$ be an ordered logic program over $\mathcal{L}$ and let X and Y be consistent sets of literals. Then, we have that

1. if X is a $<^{\mathrm{W}}$-preserving answer set of Π, then there is some answer set Y of $\mathbb{T}^{\mathrm{W}}(\Pi, <)$ such that $X = Y \cap \mathcal{L}$;
2. if Y is an answer set of $\mathbb{T}^{\mathrm{W}}(\Pi, <)$, then X is a $<^{\mathrm{W}}$-preserving.

Proof 7.8

1 Let X be a $<^{\mathrm{W}}$-preserving answer set of Π. Define

$$\begin{aligned} Y \quad = \quad & \{head(r) \mid r \in \Gamma_\Pi X\} \\ & \cup \{\mathsf{ap}(n_r) \mid r \in \Gamma_\Pi X\} \cup \{\mathsf{bl}(n_r) \mid r \notin \Gamma_\Pi X\} \\ & \cup \{\mathsf{ok}(n_r) \mid r \in \Pi\} \cup \{\mathsf{rdy}(n_r, n_{r'}) \mid r, r' \in \Pi\} \\ & \cup \{n_r \prec n_{r'} \mid r < r'\} \cup \{\neg(n_r \prec n_{r'}) \mid r \not< r'\} \end{aligned}$$

Notice that $L \in X$ iff $L \in Y$. We want to show that $Y = Cn(\mathbb{T}(\Pi, <)^Y)$ by two steps:

"$\supseteq$"-part For any $s \in \mathsf{T}^{\mathsf{w}}(\Pi, <)$, if $s^+ \in \mathbb{T}(\Pi, <)^Y$ and $body^+(s) \subseteq Y$, we need to prove $head(s) \in Y$ by cases.

Case 1 $a_1(r) : head(r) \leftarrow \mathsf{ap}(n_r)$. Since $a_1(r) = a_1(r)^+$, $a_1(r) \in \mathbb{T}(\Pi, <)^Y$. If $\mathsf{ap}(n_r) \in Y$, then $r \in \Gamma_\Pi X$. This implies $head(r) \in Y$.

Case 2 $a_2(r) : \mathsf{ap}(n_r) \leftarrow \mathsf{ok}(n_r), body(r)$. If $\mathsf{ok}(n_r) \in Y$, $body^+(r) \subseteq Y$ and $body^-(r) \cap Y = \emptyset$, then $body^+(r) \subseteq X$ and $body^-(r) \cap X = \emptyset$. This implies that $r \in \Gamma_\Pi X$ and thus $\mathsf{ap}(n_r) \in Y$.

Case 3 $b_1(r, L^+) : \mathsf{bl}(n_r) \leftarrow \mathsf{ok}(n_r), not\ L^+$. If $\mathsf{ok}(n_r) \in Y$ and $L^+ \notin Y$, then $L^+ \notin X$. That is, $r \notin \Gamma_\Pi X$ and thus $\mathsf{bl}(r) \in Y$.

Case 4 $b_2(r, L^-) : \mathsf{bl}(n_r) \leftarrow \mathsf{ok}(n_r), L^-$. If $\mathsf{ok}(n_r) \in Y$ and $L^- \in Y$, then $L^- \in X$. That is, $r \notin \Gamma_\Pi X$ and thus $\mathsf{bl}(r) \in Y$.

Case 5 For the rest of rules in $\mathsf{T}^{\mathsf{w}}(\Pi, <)$, we trivially have that $head(s) \in Y$ whenever $s^+ \in \mathbb{T}(\Pi, <)^Y$ and $body^+(s) \subseteq Y$.

"$\subseteq$"-part Since X is a $<^{\mathsf{w}}$-preserving answer set of Π, there is an enumeration $\langle r_i \rangle_{i \in I}$ of $\Gamma_\Pi X$ satisfying all conditions in Definition 8. This enumeration can be extended to an enumeration of Π as follows:

For any $r \notin \Gamma_\Pi X$, let r_i be the first rule that blocks r and r_j be the last rule s. t. $r < r_j$. Then we insert r immediately after $r_{max\{i,j\}}$. For simplicity, the extended enumeration is still denoted $\langle r_i \rangle_{i \in I}$. Obviously, this enumeration has the following property by Definition 8.

Lemma 7.9
Let $\langle r_i \rangle_{i \in I}$ be the enumeration for Π defined as above. If $r_i < r_j$, then $j < i$.

For each $r_i \in \Pi$, we define Y_i as follows:

$$\begin{array}{llll} & \{head(r_i), \mathsf{ap}(n_{r_i}) \mid r_i \in \Gamma_\Pi X, i \in I\} & \cup & \{\mathsf{bl}(n_{r_i}) \mid r_i \notin \Gamma_\Pi X, i \in I\} \\ \cup & \{\mathsf{ok}(n_{r_i}) \mid i \in I\} & \cup & \{\mathsf{rdy}(n_{r_i}, n_{r_j}) \mid i, j \in I\} \\ \cup & \{n_r \prec n_{r'} \mid r < r'\} & \cup & \{\neg(n_r \prec n_{r'}) \mid r \not< r'\}. \end{array}$$

We prove $Y_i \subseteq Cn(\mathcal{T}(\Pi)^Y)$ by using induction on i.

Base Consider $r_0 \in \Pi$. Given that X is consistent, we have $r_0 \not< r$ for all $r \in \Pi$ by Definition 8(3). Thus, $\neg(n_{r_0} \prec n_r) \in Y$ for all $r \in \Pi$. Consequently,

$$c_2(r_0, r)^+ : \mathsf{rdy}(n_{r_0}, n_r) \leftarrow \in \mathbb{T}(\Pi, <)^Y \text{ for all } r \in \Pi.$$

This implies $\mathsf{rdy}(n_{r_0}, n_r) \in Cn(\mathbb{T}(\Pi, <)^Y)$ for all $r \in \Pi$.
Let $\Pi = \{r_0, r_1, \ldots, r_k\}$. Since

$$c_1(r_0) = c_1(r_0)^+ : \mathsf{ok}(n_{r_0}) \leftarrow \ \mathsf{rdy}(n_{r_0}, n_{r_1}), \ldots, \mathsf{rdy}(n_{r_0}, n_{r_k}) \in \mathbb{T}(\Pi, <)^Y, \quad (11)$$

thus $\mathsf{ok}(n_{r_0}) \in Cn(\mathbb{T}(\Pi, <)^Y)$. We distinguish two cases.

Case 1 If $r_0 \in \Gamma_\Pi X$, we have $body^+(r_0) = \emptyset$ by Definition 8(1), and $body^-(r_0) \cap X = \emptyset$ which also implies $body^-(r_0) \cap Y = \emptyset$. Thus

$$a_2(r_0) = a_2(r_0)^+ : \mathsf{ap}(n_{r_0}) \leftarrow \mathsf{ok}(n_{r_0}) \in \mathbb{T}(\Pi, <)^Y . \quad (12)$$

Accordingly, we obtain $\mathsf{ap}(n_{r_0}) \in Cn(\mathbb{T}(\Pi, <)^Y)$ by $\mathsf{ok}(n_{r_0}) \in Cn(\mathbb{T}(\Pi, <)^Y)$.

Furthermore, from

$$a_1(r_0) = a_1(r_0)^+ : head(n_{r_0}) \leftarrow \mathsf{ap}(n_{r_0}) \in \mathbb{T}(\Pi, <)^Y \;, \tag{13}$$

we obtain $head(r_0) \in Cn(\mathbb{T}(\Pi, <)^Y)$.

Case 2 If $r_0 \in \Pi \setminus \Gamma_\Pi X$, we must have $body^+(r_0) \not\subseteq X$ by Definition 8. That is, $body^+(r_0) \not\subseteq Y$. Then, there is some $L^+ \in body^+(r_0)$ with $L^+ \notin X$. We also have $L^+ \notin Y$. Therefore,

$$b_1(r_0, L^+) = b_1(r_0, L^+)^+ : \mathsf{bl}(n_{r_0}) \leftarrow \mathsf{ok}(n_{r_0}) \in \mathbb{T}(\Pi, <)^Y \;. \tag{14}$$

Since we have shown above that $\mathsf{ok}(n_{r_0}) \in Cn(\mathbb{T}(\Pi, <)^Y)$, we obtain

$$\mathsf{bl}(n_{r_0}) \in Cn(\mathbb{T}(\Pi, <)^Y).$$

Step Assume that $Y_j \subseteq \mathbb{T}(\Pi, <)^Y$ for all $j < i$, we show $Y_i \subseteq \mathbb{T}(\Pi, <)^Y$ by cases.

- $\mathsf{rdy}(n_{r_i}, n_{r_j}) \in Cn(\mathbb{T}(\Pi, <)^Y)$:
  If $r_i < r_j$, then $n_{r_i} \prec n_{r_j} \in Y$ and $j < i$ by Lemma 7.9.
  By the induction assumption, either $\mathsf{ap}(n_{r_j}) \in Cn(\mathbb{T}(\Pi, <)^Y)$ or $\mathsf{bl}(n_{r_j}) \in Cn(\mathbb{T}(\Pi, <)^Y)$. Since $c_3(r_i, r_j)$, $c_4(r_i, r_j)$ are in $\mathbb{T}(\Pi, <)^Y$, we have

  $$\mathsf{rdy}(n_{r_i}, n_{r_j}) \in Cn(\mathbb{T}(\Pi, <)^Y) \qquad \text{whenever } r_i < r_j \;.$$

  If $r_i \not< r_j$, then $\neg(n_{r_i} \prec n_{r_j}) \in Y$ and thus

  $$c_2(r_i, r_j)^+ : \mathsf{rdy}(n_{r_i}, n_{r_j}) \leftarrow \; \in \mathbb{T}(\Pi, <)^Y.$$

  Consequently, for all $j \in I$, $\mathsf{rdy}(n_{r_i}, n_{r_j}) \in Cn(\mathbb{T}(\Pi, <)^Y)$.
- $\mathsf{ok}(n_{r_i}) \in Cn(\mathbb{T}(\Pi, <)^Y)$: It is obtained directly by $c_1(r_i)^+ = c_1(r_i) \in Cn(\mathbb{T}(\Pi, <)^Y)$.
- If $r_i \in \Gamma_\Pi X$, then $\{\mathsf{ap}(r_i), head(r_i)\} \subseteq Cn(\mathbb{T}(\Pi, <)^Y)$.
  By Definition 8, $body^+(r_i) \subseteq \{head(r_j) \mid r_j \in \Gamma_\Pi X, j < i\}$ or $head(r_i) \in \{head(r_j) \mid r_j \in \Gamma_\Pi X, j < i\}$. By the induction assumption, $body^+(r_i) \subseteq Cn(\mathbb{T}(\Pi, <)^Y)$. Also, $r_i \in \Gamma_\Pi X$ implies $body^-(r_i) \cap X = \emptyset$. Thus $body^-(r_i) \cap Y = \emptyset$.
  This means that

  $$a_2(r_i) = a_2(r_i)^+ : \mathsf{ap}(n_{r_i}) \leftarrow \mathsf{ok}(n_{r_i}), body^+(r_i) \in \mathbb{T}(\Pi, <)^Y. \tag{15}$$

  As shown above, $\mathsf{ok}(n_{r_i}) \in Cn(\mathbb{T}(\Pi, <)^Y)$. Therefore, $\mathsf{ap}(n_{r_i}) \in Cn(\mathbb{T}(\Pi, <)^Y)$. Accordingly, we obtain $head(r_i) \in Cn(\mathbb{T}(\Pi, <)^Y)$ due to $a_1(r_i)^+ \in \mathbb{T}(\Pi, <)^Y$.
- If $r_i \in \Pi \setminus \Gamma_\Pi X$, $\mathsf{bl}(n_{r_i}) \in Cn(\mathbb{T}(\Pi, <)^Y)$: We consider three possibilities.
  1. $body^+(r_i) \not\subseteq X$: then there is some $L^+ \in body^+(r_i)$ with $L^+ \notin X$.
     Also, $L^+ \notin Y$. Thus,

     $$b_1(r_i, L^+) = b_1(r_i, L^+)^+ : \mathsf{bl}(n_{r_i}) \leftarrow \mathsf{ok}(n_{r_i}) \in \mathbb{T}(\Pi, <)^Y \;. \tag{16}$$

     By $\mathsf{ok}(n_{r_i}) \in Cn(\mathbb{T}(\Pi, <)^Y)$, we have $\mathsf{bl}(n_{r_i}) \in Cn(\mathbb{T}(\Pi, <)^Y)$.
  2. $body^-(r) \cap \{head(r_j) \mid r_j \in \Gamma_\Pi X, j < i\} \neq \emptyset$: then there is some $L^- \in body^-(r_i)$ with $L^- \in \{head(r_j) \mid r_j \in \Gamma_\Pi X, j < i\}$. That is, $L^- = head(r_j)$ for some $r_j \in \Gamma_\Pi X$ with $j < i$. With the induction hypothesis, we then obtain

$L^- \in Cn(\mathbb{T}(\Pi,<)^Y)$. Since $\mathsf{ok}(n_{r_i}) \in Cn(\mathbb{T}(\Pi,<)^Y)$, we obtain $\mathsf{bl}(n_{r_i}) \in Cn(\mathbb{T}(\Pi,<)^Y)$.

3. $head(r_i) \in \{head(r_j) \mid r_j \in \Gamma_\Pi X, j < i\}$: this is obtained directly by the induction assumption.

2 Let Y be a consistent answer set of $\mathsf{T}^{\mathrm{w}}(\Pi,<)$ and $X = Y \cap \mathcal{L}$. To prove that X is a $<^{\mathrm{w}}$-preserving answer set of Π, it suffices to prove that the following two propositions $P1$ and $P2$:

P1 X is an answer set of Π: that is, $Cn(\Pi^X) = X$.

1. $Cn(\Pi^X) \subseteq X$: Let $r \in \Pi$ s. t. $body^+(r) \subseteq X$ and $body^-(r) \cap X = \emptyset$.
Then $body^+(r) \subseteq Y$ and $body^-(r) \cap Y = \emptyset$. By Lemma 7.5, $\mathsf{ok}(n_r) \in Y$ and thus $a_2(r)^+ \in \mathbb{T}(\Pi,<)^Y$. Since Y is closed under $\mathbb{T}(\Pi,<)^Y$, $\mathsf{ap}(n_r) \in Y$ and thus $head(r) \in Y$. This means $head(r) \in X$.
2. $X \subseteq Cn(\Pi^X)$: Since $X = Y \cap \mathcal{L} = (\cup_{i\geq 0} T^i_{\mathbb{T}(\Pi,<)^Y}\emptyset \cap \mathcal{L}$, we need only to show by induction on i that, for $i \geq 0$,

$$(T^i_{\mathbb{T}(\Pi,<)^Y}\emptyset \cap \mathcal{L}) \subseteq Cn(\Pi^X). \tag{17}$$

Base It is obvious that $T^0_{\mathbb{T}(\Pi,<)^Y}\emptyset = \emptyset$.
Step Assume that (17) holds for i, we want to prove (17) holds for $i+1$.
If $L \in T^{i+1}_{\mathbb{T}(\Pi,<)^Y}\emptyset$, then there is a rule $r \in \Pi$ s. t. $head(r) = L$, $a_1(r)^+, a_2(r)^+ \in \mathbb{T}(\Pi,<)^Y$ and $\{\mathsf{ap}(n_r), \mathsf{ok}(n_r)\} \cup body^+(r) \subseteq T^i_{\mathbb{T}(\Pi,<)^Y}\emptyset$. This also means $body^-(r) \cap Y = \emptyset$. By the induction assumption, $body^+(r) \in Cn(\Pi^X)$. Together with $body^-(r) \cap X = \emptyset$, we have $r \in \Pi^X$ and thus $head(r) \in Cn(\Pi^X)$. Therefore, $X = Cn(\Pi^X)$.

P2 X is $<^{\mathrm{w}}$-preserving: Since Y is a standard answer set of $\mathsf{T}^{\mathrm{w}}(\Pi,<)$, there is a grounded enumeration $\langle s_k\rangle_{k\in K}$Induction of $\Gamma_{\mathsf{T}^{\mathrm{w}}(\Pi)}Y$. Define $\langle r_i\rangle_{i\in I}$ as the enumeration obtained from $\langle s_k\rangle_{k\in K}$ by

- deleting all rules apart from those of form $a_2(r)$, $b_1(r, L^+)$, $b_2(r, L^-)$;
- replacing each rule of form $a_2(r)$, $b_1(r, L^+)$, $b_2(r, L^-)$ by r;
- removing duplicates[9] by increasing i.

for $r \in \Pi$ and $L^+ \in body^+(r)$, $L^- \in body^-(r)$.
We justify that the sequence $\langle r_i\rangle_{i\in I}$ satisfies the conditions in Definition 8:

1. Since $\langle s_k\rangle_{k\in K}$ is grounded, Condition 1 is satisfied.
2. If $r_i < r_j$, we want to show $j < i$. Since $\mathsf{rdy}(n_i, n_j) \in Y$, at least one of $a_2(r_j)$, $b_1(r_j, L^+)$, $b_2(r_j, L^-)$ appears before any of $a_2(r_i)$, $b_1(r_i, L^+)$, $b_2(r_i, L^-)$. Thus, $j < i$.
3. Let $r_i < r'$ and $r' \in \Pi \setminus \Gamma_\Pi X$. Suppose that $body^+(r) \subseteq X$ and $head(r) \notin \{head(r_k) \mid k < i\}$. Since $body^-(r) \cap X \neq \emptyset$, there is some $L^- \in body^-(r)$ s. t. $L^- \in X$. Then $L^- \in Y$. Without loss of generality, let L^- is included in Y through rule s_{k_0}. Furthermore, we can assume that there is no $k' < k_0$ such that $s_{k'}$

[9] Duplicates can only occur if a rule is blocked in multiple ways.

is before s_{k_0}, $head(s_{k'}) \in body^-(r)$ and $head(s_{k'}) \in X$. Since $\mathsf{ok}(r_i) \in Y$, we have $\mathsf{rdy}(n_i, n_{r'})$. This implies, $\mathsf{bl}(n_{r'}) \in Y$ and $b_2(r_i, L^-)$ appears before $a_2(r)$ in $\langle s_k \rangle_{k \in K}$. Thus, $L^- \in \{head(r_k) \mid k < i\}$. $\square$

Proof 17 See the proof of Theorem 19. $\square$

Proof 18 Similar to Proof 8. $\square$

Proof 19
Throughout the proofs for Theorem 19, the set X_i for any $i \geq 0$ is defined as in Definition 9. By the definition of $\mathcal{E}_X(\Pi, <)$, we observe the following facts:

F1 X is a standard answer set of $(\Pi, <)$ iff X is a standard answer set of $\mathcal{E}_X(\Pi, <)$.
F2 X is a $<^{\text{B}}$-preserving answer set of $(\Pi, <)$ iff X is a $<^{\text{B}}$-preserving answer set of $\mathcal{E}_X(\Pi, <)$.
F3 X is a standard answer set of $\mathbb{T}^{\text{B}}(\Pi, <)$ iff X is a standard answer set of $\mathbb{T}(\mathcal{E}_X(\Pi, <))$.

Having the above facts, we can assume that $(\Pi, <) = \mathcal{E}_X(\Pi, <)$. Thus, we need only to prove the following Lemma 7.10 and Lemma 7.14. $\square$

Given a statically ordered logic program $(\Pi, <)$ and a set X of literals, set $X_i = (\mathcal{T}^{\text{B}})^i_{(\Pi,<),X}\emptyset$ for $i \geq 0$.

Lemma 7.10
Let $(\Pi, <)$ be a statically ordered logic program over $\mathcal{L}$ and let X be an answer set of Π. Then, the following propositions are equivalent.

1. X is a B-preferred answer set of $(\Pi, <)$;
2. $\mathcal{C}''_{(\Pi_X, <_X)}(X) = X$.

To prove Lemma 7.10, some preparations are in order.

Definition 18
Let $(\Pi, <) = \langle r_1, r_2, \ldots, r_n \rangle$ be a totally ordered logic program, where $r_{i+1} < r_i$ for each i, and let X be a set of literals.

We define

$$
\begin{array}{rcl}
\bar{X}_0 & = & \emptyset \qquad \text{and for } i \geq 0 \\
\bar{X}_{i+1} & = & \bar{X}_i \cup \left\{ head(r_{i+1}) \;\middle|\; \begin{array}{ll} (1) & r_{i+1} \text{ is active wrt } (X, X) \text{ and} \\ (2) & \text{there is no rule } r' \in \Pi \text{ with } r_{i+1} < r' \\ & \text{such that} \\ & (a)\ r' \text{ is active wrt } (X, \bar{X}_i) \text{ and} \\ & (b)\ head(r') \notin \bar{X} \end{array} \right\}
\end{array}
$$

Then, $\mathcal{D}_{(\Pi,<)}(X) = \bigcup_{i \geq 0} \bar{X}_i$ if $\bigcup_{i \geq 0} \bar{X}_i$ is consistent. Otherwise, $\mathcal{D}_{(\Pi,<)}(X) = Lit$.

If we want to stress that $\bar{X}_i$ is for ordering $<$, we will also write it as $\bar{X}_i^<$. We assume the same notation for X_i.

Lemma 7.11
Let $(\Pi, <)$ be an ordered logic program. $\bar{X}_i$ for $i \geq 0$ is given as above and Π is prerequisite-free. Then $X_i = \bar{X}_{k_i}$ for some non-decreasing sequence $\{k_i\}_{i \geq 0}$ with $0 \leq k_1 \leq \cdots \leq k_i \leq \cdots$.

Proof 7.11
Without loss of generality, assume that $\bar{X}_0 = \cdots = \bar{X}_{k_1}, \bar{X}_{k_1+1} = \cdots = \bar{X}_{k_2}, \ldots$. Then by a simple induction on i, we can directly prove that

$$X_0 = \bar{X}_0, X_1 = \bar{X}_{k_1+1}, \ldots, X_i = \bar{X}_{k_i+1}, \ldots. \qquad \square$$

Lemma 7.12
The conclusion of Lemma 7.10 is correct for ordered logic program $(\Pi, <)$ if Π is prerequisite-free and $<$ is total.

Proof 7.12
Since Π is prerequisite-free, we have that $\Pi_X = \Pi$. By Lemma 7.11, it is enough to prove that $X = \cup \bar{X}_i$ iff $X = \cup X_i$ (see Definition 9). For simplicity, we say a rule r is applicable wrt $(X, \bar{X}_i)$ (only in this proof) if r satisfies the conditions in the definition of $\bar{X}_{i+1}$.

if part If $X = \cup \bar{X}_i$, we want to prove that $X = \cup X_i$. It suffices to show that $\bar{X}_i = X_i$ hold for all $i \geq 0$. We use induction on $i \geq 0$:

Base $\bar{X}_0 = X_0 = \emptyset$.
Step Assume that $\bar{X}_{i-1} = X_{i-1}$, we need to show that $\bar{X}_i = X_i$.

1. $\bar{X}_i \subseteq X_i$:
   If $\bar{X}_i = \bar{X}_{i-1}$, the inclusion follows from the induction assumption;
   If $\bar{X}_i \neq \bar{X}_{i-1}$, then r_i is applicable wrt $(X, \bar{X}_{i-1})$.
   Thus, r_i is not defeated by X by Definition 18.
2. $X_i \subseteq \bar{X}_i$: If $X_i = X_{i-1}$, the inclusion follows from the induction assumption; Let $X_i \neq X_{i-1}$, that is, $head(r_i) \in X_i$. Then we can assert that $head(r_i) \in \bar{X}_i$.
   Otherwise, if $head(r_i) \notin \bar{X}_i$, there will be two possible cases because Π is prerequisite-free:
   - r_i is not active wrt (X, X): then there exists a literal $l \in body^-(r_i)$ such that $l \in X$. On the other hand, since $head(r_i) \in X_i$, r_i is not defeated by $X_{i-1} = \bar{X}_{i-1}$, so we have $l \notin \bar{X}_{i-1}$. This implies that there exists $t \leq i$ such that $l \in \bar{X}_t \setminus \bar{X}_{i-1}$. Thus, $l = head(r_t)$ and $r_t < r_i$. Notice that r_i is active wrt $(X, X_{i-1}) = (X, \bar{X}_{i-1})$ and $head(r_i) \notin X$, thus r_i is active wrt $(X, \bar{X}_{t-1})$ and $head(r_i) \notin \bar{X}_{t-1}$. This implies that r_i is a preventer of r_t. Therefore, $head(r_t) \notin \bar{X}_t$ and so by $X = \cup \bar{X}_i$, $head(r_t) \notin \bar{X}$, contradiction.
   - There is a rule $r' \in \Pi$ with $r_i < r'$ such that r' is active wrt $(X, \bar{X}_{i-1})$ and $head(r') \notin X$. Since there are only a finite number of rules in Π which are preferred over r_i, so this case is impossible.

   Combining the two cases, we have $X_{i+1} \subseteq \bar{X}_{i+1}$. Thus, $X_i = \bar{X}_i$ for all $i \geq 0$.

only-if part Suppose that $X = \cup X_i$ and X is an answer set of Π, we want to prove that

$$X = \cup \bar{X}_i :$$

1. We prove $\bar{X}_i \subseteq X$ by using induction on i.
   Base $\bar{X}_0 = \emptyset \subseteq X$.

Step Assume that $\bar{X}_i \subseteq X$. If $head(r_{i+1}) \in \bar{X}_{i+1}$, then r_{i+1} is not defeated by X and thus not defeated by X_i. Thus $head(r_{i+1}) \in X_{i+1}$.

2. $X \subseteq \cup \bar{X}_i$: it is sufficient to show that $X_i \subseteq \bar{X}_i$ by using induction on i.

   Base $X_0 = \emptyset = \bar{X}_0$.

   Step Assume that $X_k \subseteq \bar{X}_k$ for $k \leq i$, then we claim that $\bar{X}_i = X_i$. On the contrary, assume that $head(r_{i+1}) \in X_{i+1} \setminus \bar{X}_{i+1}$. From $X = \cup X_i$, we have $head(r_{i+1}) \in X$. Notice that X is an answer set of Π, so we can further assume that r_{i+1} is active wrt (X, X). Therefore, $head(r_{i+1}) \not\in \bar{X}_{i+1}$ implies that there is a number $t \leq i$ such that r_t is active wrt $(X, \bar{X}_i)$ but $head(r_t) \not\in \bar{X}_i$. Thus, r_t is active wrt (X, X_{t-1}) by induction. This forces $head(r_t) \in X$ and r_t is not active wrt (X, X), contradiction. $\square$

Lemma 7.13

The conclusion of Lemma 7.10 is correct for ordered logic program $(\Pi, <)$ if Π is prerequisite-free and $<$ is a partial ordering.

Proof 7.13

if part Suppose that X is an answer set of Π and $X = \cup \bar{X}_i^{<}$.

Let $<_t$ be any total ordering on Π satisfying the following three conditions:

1. If $r < r'$ then $r <_t r'$; and
2. If r and r' are unrelated wrt $<$ two rules and they are applied in producing $\bar{X}_i$ and $\bar{X}_j$ respectively ($i < j$), then $r' <_t r$.
3. If
   - r is active wrt (X, X) and
   - r' is active wrt $(X, \bar{X}_i)$ with $head(r') \not\in \bar{X}_i$ for some i and
   - r and r' are unrelated wrt $<$,

   then $r' <_t r$.

Notice that the above total ordering $<_t$ exists. We want to prove that $X = \cup \bar{X}_i^{<_t}$. By the condition (3) above, there will be no new preventer in $(\Pi, <_t)$ for any rule r though there may be more rules that are preferred than r. Thus, $\cup \bar{X}_i^{<_t} = \cup \bar{X}_i^{<}$. That is, $X = \cup \bar{X}_i^{<_t}$. Since $<_t$ is a total ordering, $X = \cup X_i^{<_t}$. Therefore, X is a BE-preferred answer set of$(\Pi, <)$.

only-if part Suppose that X is a BE-preferred answer set of $(\Pi, <)$, then there is a total ordering $<_t$ such that $X = \cup X_i^{<_t}$. By Lemma 15, $X = \cup \bar{X}_i^{<_t}$. We want to prove that $\cup \bar{X}_i^{<_t} = \cup \bar{X}_i^{<}$: On the contrary, assume that this is not true. Then $\cup \bar{X}_i^{<_t} \subset \cup \bar{X}_i^{<}$. That is, there is a rule $r \in \Pi$ such that $head(r) \not\in \cup \bar{X}_i^{<_t} = X$ but r is active wrt (X, X). On the other hand, since X is an answer set of Π, $head(r) \in X$, contradiction. Therefore, $X = \cup \bar{X}_i^{<}$. $\square$

Proof 7.10
If Π is transformed into $\mathcal{E}_X(\Pi)$, then Π may be performed two kinds of transformations:

1. Deleting every rule having prerequisite l such that $l \in X$: this kind of rule can be neither active wrt (X, X) nor a preventer of another rule because it is not active wrt $(X, \bar{X}_i)$ for any $i \geq 0$.
2. Removing from each remaining rule r all prerequisites.

Suppose that r is changed into r' by this transformation. Then

- r is active wrt (X, X) iff r' is active wrt (X, X);
- r is a preventer in $(\Pi, <)$ iff r' is a preventer in $\mathcal{E}_X(\Pi), <)$.

By Lemma 7.13, Lemma 7.10 is proven. □

Lemma 7.14
Let $(\Pi, <)$ be a statically ordered logic program over $\mathcal{L}$ and let X be an answer set of Π. Then X satisfies the Brewka/Eiter criterion for Π (or equivalently for $\mathcal{E}_X(\Pi)$) according to (Brewka and Eiter 1999) if and only if X is a $<_{\text{B}}$-preserving answer set of Π.

To prove this theorem, the following result given in (Brewka and Eiter 1999) is required.

Lemma 7.15
Let $(\Pi, <)$ be a statically ordered logic program over $\mathcal{L}$ and let X be an answer set of Π. Then X is a B-preferred answer set if and only if, for each rule $r \in \Pi$ with $body^+(r) \subseteq X$ and $head(r) \notin X$, there is a rule $r' \in \Gamma_\Pi X$ such that $r < r'$ and $head(r') \in body^-(r)$.

Proof 7.14

if part Let X be a $<_{\text{B}}$-preserving answer set of Π.

Assume that X is not a B-preferred answer set, by Lemma 7.15, then there is a rule $r \in \Pi$ such that the followings hold:

1. $body^+(r) \subseteq X$;
2. $head(r) \notin X$ and
3. For any rule $r' \in \Gamma_\Pi X$ with $r < r'$, $head(r')$ does not defeat r.

Then, $head(r') \notin body^-(r)$. Thus $r' \in \Pi \setminus \Gamma_\Pi X$. This contradict to the Condition 2 in Definition 15. Therefore, X is a B-preferred answer set of Π.

only-if part Suppose that X is a B-preferred answer set of Π. Then X is also a B-preferred answer set of $(\Pi, <')$ where $<'$ is a total ordering and compatible with $<$. Notice that the ordering $<'$ actually determines an enumeration $\langle r_i \rangle_{i \in I}$ of $\Gamma_\Pi X$ such that $r_i <' r_j$ if $j < i$. Thus, this enumeration of $\Gamma_\Pi X$ obviously satisfies the condition 1 in Definition 15.

We prove the Condition 2 is also be satisfied. Let $r_i < r'$ and $r' \in \Pi \setminus \Gamma_\Pi X$. Suppose that $body^+(r') \subseteq X$ and $head(r') \notin X$. By Lemma 7.15, there is a rule $r_j \in \Gamma_\Pi X$ such that $r' < r_j$ and $head(r_j) \in body^-(r')$. Thus, the Condition 2 is satisfied. □

Proof 20
Under the assumption of the theorem, we can see that $\mathcal{T}^{\text{D}}_{(\Pi,<),Y}X = \mathcal{T}^{\text{W}}_{(\Pi,<),Y}X$ for any sets X and Y of literals, which implies $\mathcal{C}^{\text{D}}_{(\Pi,<)}(X) = \mathcal{C}^{\text{W}}_{(\Pi,<)}(X)$ for any set X of literals. Thus, the conclusion is obtained by Theorem 14. □

Proof 21
By comparing Condition II(b) in Definition 1 and 6, we get

$$\mathcal{T}^{\text{D}}_{(\Pi,<),Y}X \subseteq (\mathcal{T}^{\text{B}})_{(\Pi,<),Y}X.$$

This means $\mathcal{C}^{\text{D}}_{(\Pi,<)}(X) \subseteq \mathcal{C}_{(\Pi,<)}(X)$. If X is a D-preferred answer set of $(\Pi,<)$, it follows from Theorem 14 that $\mathcal{C}^{\text{D}}_{(\Pi,<)}(X) = X$. Thus, $X \subseteq \mathcal{C}_{(\Pi,<)}(X)$. On the other hand, since a D-preferred answer set is also a standard answer set, we have $\mathcal{C}_{(\Pi,<)}(X) \subseteq C_\Pi X = X$. Therefore, $X = \mathcal{C}_{(\Pi,<)}(X)$. □

Proof 22
By comparing Condition I in Definition 1 and 13, we get

$$(\mathcal{T}^{\text{B}})_{(\Pi,<),Y}X \subseteq \mathcal{T}^{\text{B}}_{(\Pi,<),Y}X.$$

This means $\mathcal{C}_{(\Pi,<)}(X) \subseteq \mathcal{C}^{\text{B}}_{(\Pi,<)}(X)$. If X is a W-preferred answer set of $(\Pi,<)$, then $X = \mathcal{C}_{(\Pi,<)}(X)$. Thus, $X \subseteq \mathcal{C}^{\text{B}}_{(\Pi,<)}(X)$. On the other hand, since X is also a standard answer set, we have $\mathcal{C}^{\text{B}}_{(\Pi,<)}(X) \subseteq C_\Pi X = X$. Therefore, $X = \mathcal{C}_{(\Pi,<)}(X)$. □

Proof 23 It follows directly from Theorem 21 and 22. □

Proof 24
By Theorem 13, the ordered program $(\Pi,<_s)$ has the unique D-preferred answer set $X^\star$. Since $< \subseteq <_s$, $X^\star$ is also a D-preferred answer set of $(\Pi,<)$. On the other hand, each stratified logic program has the unique answer set (the perfect model), ie. $\mathcal{AS}(\Pi) = \{X^\star\}$. By Theorem 23, we arrive at the conclusion of Theorem 24. □

References

ALFERES, J., LEITE, J., PEREIRA, L., PRZYMUSINSKA, H., AND PRZYMUSINSKI, T. 1998. Dynamic logic programming. In *Proc. Int'l Conference on the Principles of Knowledge Representation and Reasoning*, A. Cohn et al., Eds. Morgan Kaufmann Publishers, 98–109.

APT, K., BLAIR, H., AND WALKER, A. 1987. Towards a theory of declarative knowledge. In *Foundations of Deductive Databases and Logic Programming*, J. Minker, Ed. Morgan Kaufmann Publishers, Chapter 2, 89–148.

BAADER, F. AND HOLLUNDER, B. 1993. How to prefer more specific defaults in terminological default logic. In *Proc. Int'l Joint Conference on Artificial Intelligence*, R. Bajcsy, Ed. Morgan Kaufmann Publishers, 669–674.

BREWKA, G. 1994. Adding priorities and specificity to default logic. In *European Workshop on Logics in Artificial Intelligence*, L. Pereira and D. Pearce, Eds. Springer-Verlag, 247–260.

BREWKA, G. 1996. Well-founded semantics for extended logic programs with dynamic preferences. *Journal of Artificial Intelligence Research 4*, 19–36.

BREWKA, G. 2001. On the relation between defeasible logic and well-founded semantics. In *Proc. Int'l Conference on Logic Programming and Nonmonotonic Reasoning*, T. Eiter, W. Faber, and M. Truszczyński, Eds. Springer-Verlag, 121–132.

BREWKA, G. AND EITER, T. 1999. Preferred answer sets for extended logic programs. *Artificial Intelligence 109,* 1-2, 297–356.

BREWKA, G. AND EITER, T. 2000. Prioritizing default logic. In *Intellectics and Computational Logic: Papers in Honour of Wolfgang Bibel*, S. Hölldobler, Ed. Kluwer Acad. Publishers, 27–45.

BUCCAFURRI, F., FABER, W., AND LEONE, N. 1999. Disjunctive logic programs with inheritance. *Theory and Practice of Logic Programming 3,* 2, 293-321.

CUI, B. AND SWIFT, T. 2002. Preference logic grammars: Fixed-point semantics and application to data standardization. *Artificial Intelligence 138,* 1-2, 117–147.

DELGRANDE, J. AND SCHAUB, T. 2000. The role of default logic in knowledge representation. In *Logic-Based Artificial Intelligence*, J. Minker, Ed. Kluwer Acad. Publishers, Dordrecht, 107–126.

DELGRANDE, J., SCHAUB, T., AND TOMPITS, H. 2000a. A compilation of Brewka and Eiter's approach to prioritization. In *Proc. European Workshop on Logics in Artificial Intelligence*, M. Ojeda-Aciego, I. Guzmán, G. Brewka, and L. Pereira, Eds. Springer-Verlag, 376–390.

DELGRANDE, J., SCHAUB, T., AND TOMPITS, H. 2000b. Logic programs with compiled preferences. In *Proc. European Conference on Artificial Intelligence*, W. Horn, Ed. IOS Press, 392–398.

DELGRANDE, J., SCHAUB, T., AND TOMPITS, H. 2003. A framework for compiling preferences in logic programs. *Theory and Practice of Logic Programming 2,* 3. To appear in March 2003.

DIMOPOULOS, Y. AND KAKAS, C. 1995. Logic programming without negation as failure. In *Proc. Int'l Symposium of Logic Programming*, J. Lloyd, Ed. The MIT Press, 369–383.

EITER, T., FINK, M., SABBATINI, G., AND TOMPITS, H. 2000. Considerations on updates of logic programs. In *Proc. Seventh European Workshop on Logics in Artificial Intelligence*. Springer-Verlag, 2–20.

GELFOND, M. AND SON, T. 1997. Reasoning with prioritized defaults. In *Int'l Workshop on Logic Programming and Knowledge Representation*, J. Dix, L. Pereira, and T. Przymusinski, Eds. Springer-Verlag, 164–223.

GORDON, T. 1993. Dissertation. TU Darmstadt, Germany.

GROSOF, B. 1997. Prioritized conflict handling for logic programs. In *Logic Programming: Proc. 1997 Int'l Symposium*, J. Maluszynsk, Ed. The MIT Press, 197–211.

GROSOF, B. 1999. Business rules for electronic commerce. IBM Research. `http://www.research.ibm.com/rules/`.

KONOLIGE, K. 1988. On the relation between default and autoepistemic logic. *Artificial Intelligence 35,* 2, 343–382.

LIFSCHITZ, V. 1996. Foundations of logic programming. In *Principles of Knowledge Representation*, G. Brewka, Ed. CSLI Publications, 69–127.

LLOYD, J. 1987. *Foundations of Logic Programming*, 2nd ed. Springer-Verlag.

NUTE, D. 1987. Defeasible reasoning. In *Proc. Hawaii Int'l Conference on Systems Science*. IEEE Press, 470–477.

NUTE, D. 1994. Defeasible logic. In *Handbook of Logics in Artificial Intelligence and Logic Programming, Vol. 3*. Oxford University Press, 353–395.

PRADHAN, S. AND MINKER, J. 1996. Using priorities to combine knowledge bases. *Int'l Journal of Cooperative Information Systems 5,* 2-3, 333–364.

PRAKKEN, H. 1997. *Logical Tools for Modelling Legal Argument*. Kluwer Acad. Publishers.

PRZYMUSINSKI, T. 1988. On the declarative semantics of deductive databases and logic programs. In *Foundations of Deductive Databases and Logic Programming*, J. Minker, Ed. Morgan Kaufmann Publishers, 193–216.

RINTANEN, J. 1995. On specificity in default logic. In *Proc. Int'l Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers, 1474–1479.

SAKAMA, C. AND INOUE, K. 1996. Representing priorities in logic programs. In *Proc. Joint Int'l Conference and Symposium on Logic Programming*, M. Maher, Ed. The MIT Press, 82–96.

SCHAUB, T. AND WANG, K. 2002. Preferred well-founded semantics for logic programming by alternating fixpoints: Prel. report. In *Proc. Workshop on Nonmonotonic Reasoning*. 238–246.

SCHWIND, C. 1990. A tableaux-based theorem prover for a decidable subset of default logic. In *Proc. Conference on Automated Deduction*, M. Stickel, Ed. Springer-Verlag, 528–542.

VAN GELDER, A. 1993. The alternating fixpoint of logic programs with negation. *Journal of Computer and System Science 47*, 185–120.

WANG, K., ZHOU, L., AND LIN, F. 2000. Alternating fixpoint theory for logic programs with priority. In *Proc. Int'l Conference on Computational Logic*. Springer-Verlag, 164–178.

YOU, J., WANG, X., AND YUAN, L. 2001. Nonmonotonic reasoning as prioritized argumentation. *IEEE Transactions on Knowledge and Data Engineering 6*, 13, 968–979.

ZHANG, Y. AND FOO, N. 1997. Answer sets for prioritized logic programs. In *Proc. Int'l Symposium on Logic Programming*, J. Maluszynski, Ed. The MIT Press, 69–84.