Reliable Semantics for Extended Logic Programs with Rule Prioritization

Anastasia Analyti & Sakti Pramanik
Department of Computer Science, Michigan State University, E. Lansing, MI 48824
E-mail: analyti@cps.msu.edu, pramanik@cpswh.msu.edu

Abstract

We present a new semantics for extended logic programs with rule prioritization (EPP). The semantics, called *reliable semantics* (RS), generalizes the well-founded semantics which is defined for normal programs [34]. It also generalizes the extended well-founded semantics which is defined for non-contradictory extended programs [24]. Because of the classical negation in the head of the rules, the well-founded model of a program can be contradictory. RS is contradiction-free and defined for all $EPPs$. To avoid contradictions, only reliable rules and reliable closed world assumptions are used in the fixpoint computation of RS. A closed world assumption (CWA) is considered reliable if it is not "suspect" for any constraint violation. A rule r is considered reliable if it is not "suspect" for any constraint violation caused by rules with priority no lower than r. We define the *stable r-models* of an EPP and we show that the RS of a program coincides with its least stable r-model. The RS of an EPP, P, represents the skeptical "meaning" of P whereas stable r-models of P represent the possible "meanings" of P.
Keywords: extended logic programs, rule prioritization, constraint violation, semantics.

1 Introduction

Extended programs are normal programs extended with classical negation. Extended programs provide negative information both implicitly (negation by default $\sim$) and explicitly (classical negation $\neg$). Classical negation is needed: (i) in case of incomplete information, since it may not be justified for a particular information to be considered false because of absence of further information (closed world reasoning), (ii) when negative information should be inferred if some conditions are satisfied, for example, $\neg light\_off \leftarrow light\_on$, and (iii) to represent default reasoning and exceptions, for example, some of the exceptions of the general rule $fly(X) \leftarrow bird(X)$ are: $\neg fly(X) \leftarrow ostrich(X)$ and $\neg fly(X) \leftarrow penguin(X)$.

Several semantics for extended programs have been proposed in the literature [27, 12, 5, 21, 23, 24, 36, 6, 37]. Yet, these semantics are not defined for all extended programs. In [27], the well-founded model [34] of an extended program P is computed as that of a normal program after replacing every literal L of P with a new atom $\neg\_L$. However, the well-founded model of an extended program can be contradictory. For example, the well-founded model of $P = \{\neg p \leftarrow \sim a.$ $p \leftarrow .$ $b \leftarrow .\}$ is $\{\sim a, \neg p, p, b\}$ and because of the contradiction, P is not given any semantics in [27]. However, intuitively, the rule b is not "suspect" for the violation of the constraint $\perp \leftarrow p, \neg p$ and thus b should be true.

The *contradiction removal semantics* (CRS), defined in [21, 23], extends the well-founded semantics [34] and avoids contradictions brought about by $CWAs$. For example, the CRS of $P = \{\neg p \leftarrow \sim a.$ $p \leftarrow .$ $b \leftarrow .\}$ is $\{p, b\}$ which is non-contradictory. Yet, the problem is not totally solved since no semantics is given to $P' = \{\neg p \leftarrow .$ $p \leftarrow .$ $b \leftarrow .\}$ even though b should be true. The same arguments hold for the *argumentation semantics* defined in [6].

Human reasoning is often based on conflicting evidence and on assumptions which are not always valid. Our goal is to derive useful conclusions from programs that may be contradictory. We consider rules to be defaults. Rule prioritization can be viewed as a tool to specify confidence information about these defaults. Some reasons for rule prioritization are:

1. Difference in the reliability of sources. It is possible that a number of sources provide information about a particular topic. If the sources contradict, we wish to use ordering to resolve conflicts.

2. The dominance of specific over general information. Object-oriented programming is an example where this principle is employed.

3. Regulation. Regulation can indicate the priority of different conflicting directives. For example, university laws require that foreign students pay out-of-state tuition and TAs (teaching assistants) pay in-state tuition. However, if a student is both foreign and TA, the directives for TAs are given higher priority than the directives for foreign students.

Prioritization of defaults is investigated in [17, 9, 10, 2, 3, 18, 19, 28, 29, 30]. Yet, negation by default is not considered in these works. In [17, 9, 10, 18, 19], alternative semantics for ordered logic programs are presented. A default in an ordered logic program is a unidirectional rule. In [2, 3], a default is a clause, that is, there is no distinction between the head and the body of a default rule. The work in [29, 30] is the most general from the point of view that defaults are general formulas and when a default instance cannot be satisfied, partial satisfaction of it, is considered. A conceptualization of both implicit and explicit preferences on data is given in [14].

An *extended program with rule prioritization* (EPP) consists of a set of partially ordered rules and a set of constraints. Every rule r has a corresponding set S_r, called the *preliminary suspect set* of r, which is a subset of the body literals of r. Intuitively, when a rule r is "suspect" for a constraint violation then the rules and $CWAs$ used in the last step of the derivation of literals in S_r are also "suspect." The *reliable semantics* (RS) extends the well-founded semantics [34] and the extended well-founded semantics [24] to $EPPs$. The RS of an EPP is always defined and does not violate any constraint. Every EPP has at least one *stable r-model* (r for reliable). The RS of a program P is the least[1] fixpoint of a monotonic operator and the least stable r-model of P. An ordered logic program can be seen as an EPP which is free of default literals, $S_r = \{\}$ for every rule r, and all constraints are of the form: $\bot \leftarrow L, \neg L$. If P is an ordered logic program then the RS of P coincides with the *skeptical c-partial model* of P [9] and is a subset of the *well-founded partial model* of P [18]. When the Herbrand base of an EPP is finite, the complexity of computing RS is polynomial w.r.t. the size of the program.

The rest of the paper is organized as follows. In Section 2, we define the r-models of an EPP. In Section 3, we define the RS and stable r-models of an EPP. We show that the RS of an EPP, P, coincides with the least stable r-model of P. In Section 4, we compare RS with other semantics. Section 5 contains the concluding remarks. The proof of all propositions is given in the Appendix.

2 r-models for Extended Programs with Rule Prioritization

Our alphabet contains a finite set of constant, predicate and variable symbols from which terms and atoms are constructed in the usual way. A *classical literal* is either an atom A or its classical negation A. The classical negation of a literal L is denoted by $\neg L$ and $\neg(\neg L) = L$. The symbol $\sim$ stands for negation by default and $\sim (\sim) L = L$. A *default literal* is denoted by $\sim L$, where L is a classical literal.

An *extended program with rule prioritization* (EPP) is a tuple $P =< R_P, IC_P, <_R>$. R_P is a finite set of rules $r : L_0 \leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots, \sim L_n$, where r is a label and L_i are classical literals. Every rule r has a corresponding set $S_r \subseteq Body_r$[2], called the *preliminary suspect set of r*. IC_P is a finite set of constraints $\bot \leftarrow L_1, \ldots, L_k$, where L_i are classical literals. The precise meaning of S_r will be given in the definitions. Intuitively, when a constraint $\bot \leftarrow L_1, \ldots, L_k$ is

[1] A set I is the *least* element of a set $\mathcal{I}$ iff $I \in \mathcal{I}$ and $I \subseteq J$, for all $J \in \mathcal{I}$.

[2] $Body_r$ denotes the set of literals in the body of rule r and $Head_r$ denotes the head of rule r.

violated, the rules used in the last step of the derivation of L_i are considered "suspect." If a rule r is "suspect" for a constraint violation then the rules and *CWAs* used in the last step of the derivation of literals in S_r are also "suspect."

The values of S_r depend on the reasons for a constraint violation. There are two basic views on the reasons for the violation of a constraint $\perp \leftarrow L_1, \ldots, L_k$: (v1) rules used in the *last* step of the derivation of L_i are incomplete[3] or (v2) *CWAs* and/or rules used in *some* step of the derivation of L_i are unreliable. According to the first view (v1), the skeptical meaning of the program $P' = \{r_1 : a \leftarrow . \quad r_2 : b \leftarrow . \quad r_3 : p \leftarrow a. \quad r_4 : \neg p \leftarrow b.\}$ is $\{a, b\}$. Rules r_1 and r_2 are not used in the last step of the derivation of $p, \neg p$ and thus according to (v1), they are reliable. On the contrary, rules r_3 and r_4 are used in the *last* step of the derivation of $p, \neg p$ and thus they are considered incomplete, i.e., r_3 should be $p \leftarrow a, \sim b$ or r_4 should be $\neg p \leftarrow b, \sim a$. Consequently, the literals a, b are evaluated as true whereas both p and $\neg p$ are undefined. View (v1) is implied by ordered logic [9, 18] and vivid logic [37] and becomes explicit in our framework when $S_r = \{\}, \forall$ rule r. For example, if $S_{r_i} = \{\}, \forall i \leq 4$ then rules r_3 and r_4 are "suspect" for the violation of $\perp \leftarrow p, \neg p$. However, rules r_1 and r_2 are not "suspect" because the literals a and b do not belong to S_{r_3} or S_{r_4}. According to the second view (v2), which is more conservative than (v1), the skeptical meaning of P' is $\{\}$. This is because all rules in P' are used in the derivation of $p, \neg p$ and thus all rules in P' are considered unreliable. View (v2) becomes explicit in our framework when $S_r = Body_r, \forall$ rule r. For example, if $S_{r_i} = Body_{r_i}$, $\forall i \leq 4$ then not only rules r_3, r_4 but also r_1, r_2 are "suspect" for the constraint violation.

Other views corresponding to $S_r \neq \{\}$ and $S_r \neq Body_r$ for a rule r are also possible. For example, consider the program $P' = \{r_1 : a \leftarrow . \quad r_2 : b \leftarrow . \quad r_3 : \neg p \leftarrow . \quad r_4 : p \leftarrow a, b.\}$. If $S_{r_4} = \{a\}$ then rule r_1 is "suspect" for the violation of $\perp \leftarrow p, \neg p$ and rule r_2 is not. Consequently, the skeptical meaning of P' is $\{b\}$. Similarly, if $S_{r_4} = \{b\}$ then the skeptical meaning of P' is $\{a\}$.

The relation $<_R \subseteq R_P \times R_P$ is a strict partial order (irreflexive, asymmetric and transitive), denoting the relative reliability of the rules. Let r and r' be two rules. The notation $r < r'$ means that r is less reliable than r', that is, $r < r'$ iff $(r, r') \in <_R$. The notation $r \not< r'$ means that r is not less reliable that r'. Note that, $r \not< r$ since $<_R$ is irreflexive. Intuitively, a rule r is considered *reliable* if it is not "suspect" for any constraint violation caused by rules with priority no lower than r. Thus, deciding if a rule r is reliable depends only on the rules $r' \not< r$.

The set of instantiated classical literals of P is called the *Herbrand Base* (HB_P) of P. The

[3] We say that a rule is *incomplete* if not all possible exceptions are enumerated in its body.

constraints in $BC_P = \{\perp \leftarrow L, \neg L |\ L \in HB_P\}$ are called *basic constraints*. We assume that $BC_P \subseteq IC_P$. An EPP, P, is called *extended* iff $IC_P = BC_P$ and $<_R = \{\}$. An extended program P is called *normal* iff rules in P are free of classically negative literals. If S is a set of literals then $\sim S =_{def} \{\sim L |\ L \in S\}$ and $\neg S =_{def} \{\neg L |\ L \in S\}$.

The instantiation of an EPP, P, is defined as follows: The instantiations of R_P and IC_P are defined the usual way. Let r_{inst} and r'_{inst} be instances of rules r and r' in P then $r_{inst} < r'_{inst}$ iff $r < r'$. In the rest of the paper, we assume that programs have been instantiated and thus all rules are propositional.

Example 2.1 (credit confusion problem) Consider the following EPP, P:

R_P= { /* If Ann is a foreign student (resp. teaching assistant) then she needs 12 (resp. 6) credits */

$r_1 : need\_credits(ann, 12) \leftarrow foreign\_stud(ann).$

$r_2 : need\_credits(ann, 6) \leftarrow TA(ann).$

$r_3 : TA(ann).$

$r_4 : foreign\_stud(ann).$ where $S_{r_i} = \{\}, \forall i \leq 4\}$,

$IC_P = \{ic : \perp \leftarrow need\_credits(ann, 6), need\_credits(ann, 12).\}$ and $r_1 < r_2$.

Every classical model of R_P violates the constraint ic. Rule r_2 is considered reliable because no constraint violation is caused by rules with priority no lower than r_2, that is, r_2, r_3 and r_4. Rule r_1 is "suspect" for the violation of ic caused by r_1, r_2, r_3 and r_4 because $Head_{r_1} \in Body_{ic}$. Consequently, rule r_1 is unreliable. Since $S_{r_1} = \{\}$ and $S_{r_2} = \{\}$, the rules r_3 and r_4 with heads $TA(ann)$ and $foreign\_stud(ann)$ are not "suspect" for the violation of ic. Consequently, r_3 and r_4 are reliable. The literals $TA(ann), foreign\_stud(ann)$ and $need\_credits(ann, 6)$ should be true in the desired semantics of P because they are derived from the reliable rules r_2, r_3 and r_4.

$S_{r_1} = \{\}$ expresses that rule r_1 is incomplete, i.e., not all possible exceptions are enumerated in the body of r_1. Consequently, though rule r_1 is "suspect," there is no reason to suspect rule r_4 which is used for the derivation $foreign\_stud(ann) \in Body_{r_1}$. In contrast, if $S_{r_1} = \{foreign\_stud(ann)\}$ then the rule r_4 is "suspect" for the constraint violation. Consequently, r_4 is unreliable and the truth value of $foreign\_stud(ann)$ is undefined. If $r_1 < r_4$ or $r_2 < r_4$ or $r_3 < r_4$ then r_4 is reliable independently of the value of S_{r_1} and thus $foreign\_stud(ann)$ is evaluated as true. Similarly, if $S_{r_2} = \{\}$ or $r_1 < r_3$ or $r_2 < r_3$ or $r_4 < r_3$ then r_3 is reliable and $TA(ann)$ is evaluated as true. Otherwise, r_3 is considered unreliable and the truth value of $TA(ann)$ is undefined.

Example 2.2 (unloading the gun [13]) Consider the following EPP, P:

$R_P = \{\ r_1 : loaded(t_1) \leftarrow loaded(t_0).\ \ldots\ r_n : loaded(t_n) \leftarrow loaded(t_{n-1}).$

$r_0 : loaded(t_0). \quad r_{n+1} : \neg loaded(t_n). \quad$ where $S_{r_i}=\{\}, \forall i \leq n+1\},$

$IC_P = BC_P, r_i < r_0$ and $r_i < r_{n+1}, \forall i \in \{1, \ldots, n\}.$

Rules $r_1, \ldots, r_n$ are instances of the default rule "if a gun is loaded at time t_i then it will still be loaded at time t_{i+1}." Rules r_0 and r_{n+1} represent the facts that the gun is loaded at time t_0 and it is found unloaded at time t_n. Note that every classical model of R_P violates the constraint $\bot \leftarrow loaded(t_n), \neg loaded(t_n)$. The rules r_0 and r_{n+1} are reliable because they have higher priority than $r_1, \ldots, r_n$ and they do not generate any constraint violation. Since $S_{r_n} = \{\}$, the rules $r_1, \ldots, r_{n-1}$ are not "suspect" for the constraint violation even though they are used in the derivation of $loaded(t_n)$. So, rules $r_1, \ldots, r_{n-1}$ are reliable and the only unreliable rule is r_n. This implies that the gun remained loaded until t_{n-1}.

If $S_{r_i} = loaded(t_i - 1), \forall 1 \leq i \leq n$, then all rules $r_j, j = 1, ..., n$, are "suspect" for the constraint violation because they are used in the derivation of $loaded(t_j), j = 1, \ldots, n$. Consequently, all rules $r_1, \ldots, r_n$ are unreliable. This implies that the gun was unloaded some time between t_1 and t_n but we do not know exactly when.

Definition 2.1 (interpretation) Let P be an EPP. A set $I = T \cup \sim F$ is an interpretation of P iff T and F are disjoint subsets of HB_P. An interpretation I is *consistent* iff there is no constraint $\bot \leftarrow L_1, \ldots, L_n$ in P s.t. $L_i \in T, \forall i \leq n$. An interpretation I is *coherent* iff it satisfies the *coherence property*: $L \in F$ if $\neg L \in T$.

In interpretation $I = T \cup \sim F$, T contains the *classically true* literals, $\neg T$ contains the *classically false* literals and F contains the literals *false by default*. When I is a consistent interpretation, there is no L such that $L \in T$ and $\neg L \in T$ because this will violate the basic constraint $\bot \leftarrow L, \neg L$. The *coherence property* first appeared in [24] and it expresses the intuition that if a literal is classically false then it is also false by default.

Definition 2.2 (truth valuation of a literal) A literal L is true (resp. false) w.r.t. an interpretation I iff $L \in I$ (resp. $\sim L \in I$). A literal that is neither true nor false w.r.t. I, it is undefined w.r.t. I.

An interpretation I can be seen equivalently as a function from the set of ground classical literals to $\{0, 1/2, 1\}$, where $I(L) = 1$ when L is true w.r.t. $I, I(L) = 0$ when L is false w.r.t.

I and $I(L) = 1/2$ when L is undefined w.r.t. I. Both views of an interpretation, as a set and as a function, will be used in the paper. Note that, $I(\sim L) = 1 - I(L)$, for any literal L. If I is a coherent interpretation then $I(L) = 1$ implies $I(\neg L) = 0$. We define $I(\emptyset) =_{def} 1$ and $I(S) =_{def} min\{I(L)|\ L \in S\}$, where S is a non-empty set of literals.

The *coherence operator* (coh) [24] transforms an interpretation to a coherent one.

Definition 2.3 (coh operator) Let $I = T \cup \sim F$ be an interpretation of an EPP. $coh(I)$ is the coherent interpretation $T \cup \sim F'$, where $F' = F \cup \{L|\ \neg L \in T\}$.

Let I be a set of literals known to be true. In Definitions 2.6 and 2.8, the concepts of *reliable default literal* and *reliable rule* w.r.t. I are defined. These concepts are used in the fixpoint computation of the RS of an EPP. In RS, a default literal $\sim L$ is true by CWA only if $\sim L$ is a reliable default literal w.r.t. RS and a rule r is used for the derivation of $Head_r$ only if r is a reliable rule w.r.t. RS.

The next definition expresses that a rule r should be blocked if $\neg Head_r$ is known to be true.

Definition 2.4 (blocked rule) Let I be a literal set. A rule r is *blocked* w.r.t. I iff $\neg Head_r \in I$.

To decide if a default literal is reliable w.r.t. I, all possible constraint violations should be considered. For this, the set of literals Pos_I is computed. Intuitively, Pos_I is the possibly inconsistent well-founded model of R_P when rules are blocked as indicated in I and coherence is enforced. Specifically, Pos_I is the least fixpoint of the monotonic operator PW_I which resembles the W operator of the well-founded semantics [34]. When P is a normal program, $PW_\emptyset \equiv W$.

Definition 2.5 (possible literal set w.r.t. I) Let P be an EPP and I, J be sets of literals. The *possible literal set* w.r.t. I, Pos_I, is defined as follows:

- $PT_{I,J}(T)$=$\{L|\ \exists$ rule $r: L \leftarrow L_1, \ldots, L_n$ in P s.t. r is not blocked w.r.t. I and $L_i \in T \cup J, \forall i \leq n\}$.
- $PT_I(J) = \cup\ \{PT_{I,J}^{\uparrow a}(\emptyset)|\ a < \omega\}$, where ω is the first infinite ordinal.
- $PF(J)$ is the greatest set S of classical literals s.t. $\forall L \in S$, if r is a rule in P s.t. $Head_r = L$ then $\exists L' \in Body_r$ s.t. $L' \in S$ or $\sim L' \in J$.
- $PW_I(J) = coh(PT_I(J) \cup \sim PF(J))$.

- Pos_I is the least fixpoint of the operator PW_I.

A default literal $\sim K$ is reliable w.r.t. I if there is no constraint violation caused by Pos_I that depends on $\sim K$. In other words, $\sim K$ is reliable if it is not "suspect" for any constraint violation. If r is a rule with $Body_r \subseteq Pos_I$ and a constraint violation depends on $Head_r$ then the constraint violation depends also on all literals in S_r. If a constraint violation depends on a default literal $\sim K$ then the constraint violation depends also on $\neg K$.

Definition 2.6 (dependency set w.r.t. I, reliable default literal) Let P be an EPP, L be a literal and I be a set of literals.

- The *dependency set* of L w.r.t. I, $Dep_I(L)$, is the least set $D(L)$ such that:
  - if L is the default literal $\sim K$ then $\{\sim K\} \subseteq D(\sim K)$ and $D(\neg K) \subseteq D(\sim K)$.
  - if $\exists r : L \leftarrow L_1, \ldots, L_n$ in P s.t. $Body_r \subseteq Pos_I$ then $\{L\} \subseteq D(L)$ and $\forall L_i \in S_r, D(L_i) \subseteq D(L)$.
- A default literal $\sim K$ is *unreliable* w.r.t. I iff $\exists \bot \leftarrow L_1, \ldots, L_n$ in P s.t. $\sim K \in Dep_I(L_i)$, for an $i \leq n$ and $L_j \in Pos_I$, $\forall j \in \{1, \ldots, n\} - \{i\}$. Otherwise, $\sim K$ is *reliable* w.r.t. I.

Note that only the dependency sets of literals in S_r are considered in the computation of $Dep_I(Head_r)$. This is because even if r is "suspect" for a constraint violation caused by Pos_I, the rules and $CWAs$ used in the derivation of $Body_r - S_r$ are not necessarily "suspect" for this constraint violation.

Example 2.3 Consider the EPP, P:

$R_P = \{r_1 : fly. \quad r_2 : \neg fly \leftarrow \sim bird.$ with $S_{r_2} = \{\sim bird\}$, $IC_P = BC_P$ and $<_R = \{\}$.

The default literal $\sim bird$ is unreliable w.r.t. $I = \emptyset$ because $\bot \leftarrow fly, \neg fly$ is a constraint, $\sim bird \in Dep_I(\neg fly)$ and $fly \in Pos_I = coh(\{fly, \neg fly, \sim bird\})$. In case that $S_{r_2} = \{\}, \sim bird$ is reliable w.r.t. I because $\sim bird \notin Dep_I(\neg fly)$ and $\sim bird \notin Dep_I(fly)$.

To decide if a rule r is reliable w.r.t. I, all possible constraint violations caused by rules with priority no lower than r, should be considered. For this, the set of literals $Pos_{r,I}$ is computed. Intuitively, $Pos_{r,I}$ is the set of literals proved by $\{\sim L \leftarrow \neg L |\ L \in HB_P\}$ and the rules $r' \not< r$ when rules are blocked as indicated in I and the truth value of $Body_{r'} - S_{r'}$ is as indicated in Pos_I.

Let r, r' be rules. We define $r \equiv_R r'$ when $r'' < r$ iff $r'' < r', \forall$ rule r''. The equivalence relation $\equiv_R$ partitions the rules in P into equivalence classes. The equivalence class of a rule r is denoted by $[r]$. When $r \equiv_R r'$, the set of rules with priority no lower than r is the same as the set of rules with priority no lower than r'. So, if $r \equiv_R r'$ then $Pos_{r,I} = Pos_{r',I}$. In other words, the literal set $Pos_{r,I}$ corresponds to the class of rules $[r]$ and the literal set I.

Definition 2.7 (possible literal set w.r.t. $[r]$ and I) Let P be an EPP, r be a rule and I be a literal set. The possible literal set w.r.t. $[r]$ and I, $Pos_{r,I}$, is defined as follows:

- $P_{r,I}(Pos) = coh(\{Head_{r'} |\ \exists$ rule r' in P s.t. (i) $r' \not< r$, (ii) r' is not blocked w.r.t. I, (iii) $S_{r'} \subseteq Pos$ and (iv) $Body_{r'} - S_{r'} \subseteq Pos_I\})$

- $Pos_{r,I} = \cup\ \{P_{r,I}^{\uparrow a}(\emptyset)|\ a < \omega\}$, where ω stands for the first infinite ordinal.

A rule r is reliable w.r.t. I if there is no constraint violation caused by $Pos_{r,I}$ that depends on r. Intuitively, r is reliable if it is not "suspect" for any constraint violation caused by rules $r' \not< r$. If a constraint violation depends on a rule r'' then the constraint violation depends also on all rules $r' \not< r$ with (i) $Head_{r'} \in S_{r''}$ or $\sim \neg Head_{r'} \in S_{r''}$, (ii) $S_{r'} \subseteq Pos_{r,I}$ and (iii) $Body_{r'} - S_{r'} \subseteq Pos_I$. In the computation of RS, the derivation of literals in $Body_{r'} - S_{r'}$ may be based on $CWAs$ and rules $r' < r$ that are in conflict with r. This is because these $CWAs$ and rules are not necessarily "suspect" for this conflict and thus not necessarily unreliable. Since only rules $r' \not< r$ are used in the computation of $Pos_{r,I}$, the truth value of $Body_{r'} - S_{r'}$ should be as indicated by Pos_I. Intuitively, the truth value of $Body_{r'} - S_{r'}$ is independent of $<_R$.

Definition 2.8 (dependency set w.r.t. $[r]$ and I, reliable rule) Let P be an EPP, r be a rule, L be a literal and I be a literal set.

- The *dependency set* of L w.r.t. $[r]$ and I, $Dep_{r,I}(L)$, is the least set $D(L)$ such that:
  - if L is the default literal $\sim K$ then $\sim K \subseteq D(\sim K)$ and $D(\neg K) \subseteq D(\sim K)$.
  - if $\exists r' : L \leftarrow L_1, ..., L_n$ in P s.t. (i) $r' \not< r$, (ii) $S_{r'} \subseteq Pos_{r,I}$ and (iii) $Body_{r'} - S_{r'} \subseteq Pos_I$ then $L \subseteq D(L)$ and $\forall L_i \in S_{r'}, D(L_i) \subseteq D(L)$.

- A rule r is *unreliable* w.r.t. I iff (i) $S_r \subseteq Pos_{r,I}$, (ii) $Body_r - S_r \subseteq Pos_I$ and (iii) $\exists \bot \leftarrow L_1, \ldots, L_n$ in P s.t. $Head_r \in Dep_{r,I}(L_i)$, for an $i \leq n$ and $L_j \in Pos_{r,I}, \forall j \in \{1, ..., n\} - \{i\}$. Otherwise, r is *reliable* w.r.t. I.

Similarly to $Pos_{r,I}$, if $r \equiv_R r'$ then $Dep_{r,I}(L) = Dep_{r',I}(L), \forall$ literal L. Note that only the dependency sets of literals in $S_{r'}$ are considered in the computation of $Dep_{r,I}(Head_{r'})$. This is because even if r' is "suspect" for a constraint violation caused by $Pos_{r,I}$, the rules and $CWAs$ used in the derivation of $Body_{r'} - S_{r'}$ are not necessarily "suspect" for this constraint violation.

Note that if $S_{r'} = Body_{r'}$ for every rule r' then $Pos_{r,I}$ does not contain literals whose derivation is based on $CWAs$. This implies that no rule r is considered unreliable merely due to constraint violations caused by $CWAs$. Intuitively, when $S_{r'} = Body_{r'}, \forall$ rule r', every rule is given higher priority than the $CWAs$. In Example 2.4, we show that this is not true when there is a literal $L \in Body_{r'} - S_{r'}$ for a rule r' and $L \in Pos_I$. It is easy to see that if a default literal or rule is reliable w.r.t. I then it is also reliable w.r.t. any literal set $I' \supseteq I$.

Example 2.4 Let P be as in Example 2.3, i.e.,

$R_P = \{r_1 : fly. \quad r_2 : \neg fly \leftarrow \sim bird. \quad$ with $S_{r_2} = \{\sim bird\}\}$,
$IC_P = BC_P$ and $<_R = \{\}$.

Let $I = \emptyset$. In Example 2.3, we showed that $\sim bird$ is unreliable w.r.t. I. Thus, we expect the literal $\sim bird$ to be evaluated as unknown. The rule r_1 is reliable w.r.t. I because $Pos_{r_1,I} = coh(\{fly\})$ and $Head_{r_1} \notin Dep_{r_1,I}(\neg fly)$. Thus, we expect the literal fly to be evaluated as true. Rule r_2 is reliable w.r.t. I because $S_{r_2} = \{\sim bird\}$ is not a subset of $Pos_{r_2,I} = coh(\{fly\})$.

If $S_{r_2} = \{\}$ then $\sim bird$ is reliable w.r.t. I and $Pos_{r_1,I} = Pos_{r_2,I} = coh(\{fly, \neg fly\})$. The rule r_1 is unreliable w.r.t. I because $Head_{r_1} \in Dep_{r_1,I}(fly)$. The rule r_2 is unreliable w.r.t. I because $Head_{r_2} \in Dep_{r_2,I}(\neg fly)$ and $Body_{r_2} - S_{r_2} = Body_{r_2} \subseteq Pos_I$. Thus, we expect the literals $fly, \neg fly$ to be evaluated as unknown and the literal $\sim bird$ to be evaluated as true.

Note that when $S_{r_2} = \{\sim bird\}$, rule r_1 is reliable and $\sim bird$ is unreliable w.r.t. I. Intuitively, if $S_{r_2} = \{\sim bird\}$ then rule r_1 is given higher priority than the $CWA \sim bird$. This is not true when $S_{r_2} = \{\}$, i.e., rule r_1 is unreliable and $\sim bird$ is reliable w.r.t. I.

Example 2.5 Let P be as in Example 2.1 and $I = \emptyset$. Then, rule r_2 is reliable w.r.t. I since $Pos_{r_2,I} = coh(\{TA(ann),\ foreign\_stud(ann),\ need\_credits(ann,6)\})$ and $Head_{r_2} \notin Dep_{r_2,I}(need\_credits(ann,12)) = \{\}$. Though $Pos_{r_4,I} = coh(\{TA(ann),\ foreign\_stud(ann),\ need\_credits(ann,6),\ need\_credits(ann,12)\})$ violates the constraint ic, rule r_4 is reliable w.r.t. I since $Head_{r_4} \notin Dep_{r_4,I}(need\_credits(ann,X)) = \{need\_credits(ann,X)\}$ for $X = 6, 12$. Similarly to r_4, rule r_3 is reliable w.r.t. I. Rule r_1 is unreliable w.r.t. I since $need\_credits(ann,6) \in Pos_{r_1,I}, Head_{r_1} \in Dep_{r_1,I}(need\_credits(ann,12))$ and $Body_{r_1} - S_{r_1} = Body_{r_1} \subseteq Pos_I$.

If $S_{r_1} = \{foreign\_stud(ann)\}$ then the rule r_4 is unreliable w.r.t. I because $Head_{r_4} \in Dep_{r_4,I}(need\_credits(ann,12))$. However, for any value of S_{r_1}, if $r_2 < r_4$ then $Pos_{r_4,I} = coh(\{TA(ann), foreign\_stud(ann)\})$ and thus rule r_4 is reliable w.r.t. I. Similarly, for any value of S_{r_1}, if $r_1 < r_4$ or $r_3 < r_4$ then rule r_4 is reliable w.r.t. I.

Definition 2.9 (truth valuation of a rule) Let P be an EPP. A rule r is r-true w.r.t. an interpretation I iff: (i) $I(Head_r) \geq I(Body_r)$ or (ii) $I(Body_r) = 1/2$ and $I(\neg Head_r) = 1$ or (iii) $I(Body_r) = 1$ and $(I(Head_r) = 1/2$ or $I(\neg Head_r) = 1)$ and r is unreliable w.r.t. I.

Definition 2.10 (r-model) Let P be an EPP. A consistent, coherent interpretation I of P is an r-model of P iff every rule in P is r-true w.r.t. I.

Example 2.6 Let P be as in Example 2.3. Then, $M = coh(\{fly\})$ is an r-model of P. We will show that $\sim bird$ is not true in any r-model of P. Let M' be an r-model of P. Then, $fly \in M'$ because r_1 is reliable w.r.t. $\emptyset$ and consequently r_1 is reliable w.r.t. $M' \supseteq \emptyset$. This implies that $\neg fly \notin M'$ because M' is a consistent interpretation of P. So, $\sim bird \notin M'$ because otherwise $\neg fly \in M'$ since r_2 is reliable w.r.t. $M' \supseteq \emptyset$. The literal $\sim \neg fly$ should also belong to M' because M' is a coherent interpretation. In case that $S_{r_2} = \{\}$, the r-models of P are $M_1 = \{\sim bird\}, M_2 = coh(\{fly, \sim bird\})$ and $M_3 = coh(\{\neg fly, \sim bird\})$.

Example 2.7 Let P be as in Example 2.1. Then, $M = coh(\{TA(ann),\ foreign\_stud(ann), need\_credits(ann,6)\})$ is an r-model of P. We will show that M is the unique r-model of P. In Example 2.5, we showed that rules r_2, r_3 and r_4 are reliable w.r.t. $I = \emptyset$. Let M' be an r-model of P. Then, r_2, r_3 and r_4 are reliable w.r.t. $M' \supseteq \emptyset$. So, the literals $TA(ann), foreign\_stud(ann), need\_credits(ann,6)$ belong to M'. The literal $need\_credits(ann,12) \notin M'$ because otherwise M' will violate the constraint ic.

Let P be a normal program and I be an interpretation as defined in [26, 27]. In [27], a rule r is true w.r.t. I iff $I(Head_r) \geq I(Body_r)$. Since P is a normal program, rules do not contain classically negative literals and the only constraints are the basic constraints. So, every rule in P is reliable w.r.t. $I' = I \cup \{\sim \neg A |\ A \text{ is an atom of } P\}$ and conditions (ii) and (iii) in Definition 2.9 are not satisfied by I', for all rules in P. This implies that a rule r in P is r-true w.r.t. I' iff r is true w.r.t. I.

Proposition 2.1 Let P be a normal program. M is a model of P iff $M \cup \{\sim \neg A|\ A$ is an atom of $P\}$ is an r-model of P.

The partial order $<'_R$ is an extension of the partial order $<_R$ iff $(r, r') \in <_R$ implies $(r, r') \in <'_R$. Let $P =< R_P, IC_P, <_R>$ and $P' =< R_P, IC_P, <_{R'}>$ be $EPPs$, where $<_{R'}$ is an extension of $<_R$. It is desirable that any r-model of P' is an r-model of P. This is because, if the reliabilities of rules r and r' cannot be compared then both $r < r'$ and $r' < r$ are possible. So, any extension of $<_R$ is possible to express the relative reliability of the rules in R_P.

Proposition 2.2 Let $P =< R_P, IC_P, <_R>$ be an EPP and $<'_R$ be an extension of $<_R$. Every r-model of $P' =< R_P, IC_P, <'_R>$ is an r-model of P.

3 Reliable Semantics

In this Section, we define the reliable model, stable r-models and reliable semantics of an EPP, P. We define the reliable model of P, RM_P, as the least fixpoint of a monotonic operator. We show that RM_P is the least stable r-model of P. In the computation of RM_P, a default literal $\sim L$ is true by CWA only if $\sim L$ is reliable w.r.t. RM_P. A rule r is used for the derivation of $Head_r$ only if r is a reliable rule w.r.t. RM_P.

The definition of an r-unfounded set for an EPP extends that of an unfounded set for a normal program [vGRS91]. If S is an r-unfounded set w.r.t. a literal set J then $\forall L \in S, \sim L$ is reliable w.r.t. J. Note that if P is a normal program then all default literals of P are reliable w.r.t. any literal set J.

Definition 3.1 (r-unfounded set) Let P be an EPP and J be a set of literals. A set S of classical literals is r-unfounded w.r.t. J iff $\forall L \in S$, (i) if r is a rule in P with $Head_r = L$ then $\exists L' \in Body_r$ s.t. $L' \in S$ or $\sim L' \in J$ and (ii) $\sim L$ is reliable w.r.t. J.

The W_P operator for EPPs extends the W_P operator for normal programs [34]. This is because if P is a normal program and J a literal set then (i) every rule is reliable w.r.t. J and (ii) a set S is an r-unfounded set w.r.t. J iff S is an unfounded set w.r.t. J.

Definition 3.2 (W_P operator) Let P be an EPP and J be a set of literals. We define:

- $T_J(T) = \{L | \exists$ rule $r : L \leftarrow L_1, \ldots, L_n$ in P s.t. (i) $L_i \in T \cup J, \forall i \leq n$ and (ii) r is reliable w.r.t. $J\}$.
- $T(J) = \cup \{T_J^{\uparrow a}(\emptyset) | \ a < \omega\}$, where ω is the first limit ordinal.

- $F(J)$ is the greatest r-unfounded set w.r.t. J.

- $W_P(J) = coh(T(J) \cup \sim F(J))$.

The sequence $T_J^{\uparrow a}(\emptyset)$ is monotonically increasing (w.r.t. $\subseteq$). So, $T(J)$ is the least fixpoint of the operator T_J. The union of two r-unfounded sets w.r.t. an interpretation J is an r-unfounded set w.r.t. J. So, $F(J)$ is the union of all r-unfounded sets w.r.t. J. We define the transfinite sequence I_a as follows: $I_0 = , I_{a+1} = W_P(I_a)$ and $I_a = \cup \{I_b | \ b < a\}$ if a is a limit ordinal.

Proposition 3.1 Let P be an EPP. I_a is a monotonically increasing (w.r.t. $\subseteq$) sequence of consistent, coherent interpretations of P.

Since I_a is monotonically increasing (w.r.t. $\subseteq$), there is a smallest countable ordinal d s.t. $I_d = I_{d+1}$ [8].

Proposition 3.2 Let P be an EPP. Then, I_d is an r-model of P.

Definition 3.3 (reliable semantics) Let P be an EPP. The *reliable model* of P, denoted as RM_P, is the interpretation I_d. The *reliable semantics* of P is the "meaning" represented by RM_P.

Example 3.1 Consider the EPP, P:

$R_P = \{r_1 : q. \quad r_2 : p \leftarrow q. \quad r_3 : \neg p. \quad r_4 : p \leftarrow \sim r. \qquad \text{with } S_{r_i} = Body_{r_i}, \forall i \leq 4\}$,
$IC_P = BC_P$ and $r_3 < r_2, r_2 < r_1, r_3 < r_1$.

Computation of $W_P(\emptyset)$: Rule r_1 is reliable w.r.t. $\emptyset$ because it has higher priority than rules r_2 and r_3 and rules r_1, r_4 do not generate a constraint violation. Similarly, rule r_2 is reliable w.r.t. $\emptyset$. In contrast, rule r_3 is unreliable w.r.t. $\emptyset$ because $p \in Pos_{r_3,\emptyset} = coh(\{q, p, \neg p\})$ and $Head_{r_3} \in Dep_{r_3,\emptyset}(\neg p)$. So, $T(\emptyset) = \{q, p\}$. The literal $\sim r$ is unreliable w.r.t. $\emptyset$ because $\neg p \in Pos_{\emptyset} = coh(\{q, p, \neg p, \sim r\})$ and $\sim r \in Dep_{\emptyset}(p)$. So, $F(\emptyset) = \{\}$ and $W_P(\emptyset) = coh(\{p, q\})$.

Computation of $W_P^{\uparrow 2}(\emptyset)$: Rule r_3 is unreliable w.r.t. $W_P(\emptyset)$. So, $T(W_P(\emptyset)) = T(\emptyset)$. However, $\sim r$ is reliable w.r.t. $W_P(\emptyset)$ because r_3 is blocked w.r.t. $W_P(\emptyset)$ and consequently, $\neg p \notin Pos_{\emptyset}$. So, $F(W_P(\emptyset)) = \{\sim r\}$ and $W_P^{\uparrow 2}(\emptyset) = coh(\{p, q, \sim r\})$.

Computation of $W_P^{\uparrow 3}(\emptyset)$: Because r_3 is unreliable w.r.t. $W_P^{\uparrow 2}(\emptyset)$, it follows that $W_P^{\uparrow 3}(\emptyset) = W_P^{\uparrow 2}(\emptyset)$. So, $RM_P = W_P^{\uparrow 2}(\emptyset)$.

Example 3.2 Let P be the program of Example 2.1. Then, the interpretation $coh(\{TA(ann), foreign\_stud(ann), need\_credits(ann, 6)\})$ is the reliable model of P. If P' is as P with $<_R = \{\}$ then the reliable model of P' is $coh(\{TA(ann), foreign\_stud(ann)\})$ which corresponds to the skeptical meaning of P'. If P' is as P with $S_r = Body_r, \forall$ rule r then rules r_3 and r_4 are unreliable w.r.t. $\emptyset$ and thus the reliable model of P' is $\{\}$.

Proposition 3.3 Let $P =< R_P, IC_P, <_R>$ be an EPP. The complexity of computing RM_P is $O(|HB_P| * |R_P| * max(|IC_P|, |HB_P| * |EC_R|)$, where EC_R is the set of equivalence classes of R_P w.r.t. $\equiv_R$.

The reliable model of an EPP corresponds to its skeptical meaning. Credulous meanings can be obtained using the transformation $P/_rI$, where I is an interpretation of P. The transformation P/I for a normal program P is defined in [11, 27]. $P/_rI$ extends P/I to $EPPs$.

Definition 3.4 (transformation $P/_rI$) Let P be an EPP and I be an interpretation of it. The program $P/_rI$ is obtained as follows:

1. Remove from P all rules that contain in their body a default literal $\sim L$ s.t. $I(L) = 1$.
2. Remove from P any rule r with $I(\neg Head_r) = 1$.
3. If r is a rule in P s.t. $I(Body_r) = 1$ and $I(Head_r) = 1/2$ then replace r with $Head_r \leftarrow u$.
4. Remove from the body of the remaining rules of P any default literal $\sim L$ s.t. $I(L) = 0$.
5. Replace all remaining default literals $\sim L$ with u.
6. If $I(L) = 1/2$ and $\sim L$ is unreliable w.r.t. I then add the rule $L \leftarrow u$.
7. Replace every classically negative literal $\neg A$ with a new atom $\neg\_A$.

The program $P/_rI$ is a non-negative program with a special proposition u. For any interpretation $J, J(u) = 1/2$. When P is a normal program and M is a model of P [27], $P/_rM \equiv P/M$ since Steps (2), (3), (6) and (7) do not have any effect on $P/_rM$.

We say that a model M of P is the $least_v$ model of P iff $M(L) \leq M'(L)$ for any model M' and classical literal L of P. The $least_v$ model of a non-negative program can be obtained as the $least_v$ fixpoint of the Ψ_P operator [27] which generalizes the immediate consequence operator of [33].

Definition 3.5 (Ψ_P operator [27]) Let P be a non-negative program, I be an interpretation and A be an atom of P. $\Psi_P(I)$ is defined as follows:

1. $\Psi_P(I)(A) = 1$ if $\exists$ rule r s.t. $Head_r = A$ and $I(Body_r) = 1$.

2. $\Psi_P(I)(A) = 1/2$ if $\Psi_P(I)(A) \neq 1$ and $\exists$ rule r s.t. $Head_r = A$ and $I(Body_r) = 1/2$.

3. $\Psi_P(I)(A) = 0$, otherwise.

Definition 3.6 (stable r-model) Let P be an EPP and M be an r-model of P. M is a stable r-model of P iff $least_v(P/_rM) = M$.

Stable r-models represent possible "meanings" of a program. For example, let P' be as the program P of Example 2.1 with $<_R = \{\}$. Then, the stable r-models of P' are:

$M_1 = coh(\{TA(ann), foreign\_stud(ann), need\_credits(ann, 6)\})$,
$M_2 = coh(\{TA(ann), foreign\_stud(ann), need\_credits(ann, 12)\})$, and
$M_3 = RM_{P'} = coh(\{TA(ann), foreign\_stud(ann)\})$.

The program P of Example 2.1 has a unique stable r-model equal to $RM_P = coh(\{TA(ann), foreign\_stud(ann), need\_credits(ann, 6)\})$.

Proposition 3.4 Let P be an EPP. The reliable model of P is a stable r-model of P.

Proposition 3.5 Let P be an EPP. The reliable model of P is the least stable r-model of P.

Proposition 3.6 Let $P = < R_P, IC_P, <_R >$ be an EPP and $<'_R$ be an extension of $<_R$. Every stable r-model of $P' = < R_P, IC_P, <'_R >$ is a stable r-model of P and $RM_P \subseteq RM_{P'}$.

We will consider an application of RS to diagnosis and we will show how prioritized defaults can be used to express the relative reliability of circuit components.

Example 3.3 The circuit of Figure 1 consists of two inverters and one AND gate. To reason about its behavior, we give a simple formulation with an EPP, P:

R_P={ /* description of I1 gate */
$r_1 : \neg c \leftarrow a, OK\_I1.$ $r_2 : c \leftarrow \neg a, OK\_I1.$

/* description of I2 gate */
$r_3 : \neg d \leftarrow b, OK\_I2.$ $r_4 : d \leftarrow \neg b, OK\_I2.$

/* description of A1 gate */
$r_5 : e \leftarrow c, d, OK\_A1.$ $r_6 : \neg e \leftarrow \neg c, OK\_A1.$ $r_7 : \neg e \leftarrow \neg d, OK\_A1.$

Figure 1: A digital circuit

$r_8 : a.$ /* a input has value 1 */ $r_9 : \neg b.$ /* b input has value 0 */
$r_{10} : e.$ /* e output has value 1 */

/* assumptions that gates are working correctly*/
$r_{11} : OK\_I1.$ $r_{12} : OK\_I2.$ $r_{13} : OK\_A1.$

$S_{r_i} = Body_{r_i}, \forall i \leq 13\}$,

$IC_P = BC_P$, and $<_R$ indicates that any rule $r_i, i = 1, \ldots, 10$ has higher priority than any rule $r_j, j = 11, 12, 13$.

Note that every classical model of R_P violates the constraint $\bot \leftarrow e, \neg e$. Apparently, the above circuit is faulty. Though there is no evidence that I2 does not work correctly, one of the gates I1 and A1 should be faulty. All rules $r_i, i \leq 10$, are reliable w.r.t. $\emptyset$ because they have higher priority than rules r_{11}, r_{12} and r_{13} and do not generate any constraint violation. Rule r_{12} also is reliable w.r.t. $\emptyset$ because OK_I2 does not belong neither to $Dep_{r_{12},\emptyset}(e)$ nor to $Dep_{r_{12},\emptyset}(\neg e)$. Rule r_{11} is unreliable w.r.t. $\emptyset$ because $e \in Pos_{r_{11},\emptyset}$ and $OK\_I1 \in Dep_{r_{11},\emptyset}(\neg e)$. Similarly, rule r_{13} is unreliable w.r.t. $\emptyset$. The reliable model of P is $RM_P = coh(\{a, \neg b, d, e, OK\_I2\})$. The truth values of OK_I1, OK_A1 are unknown because rules r_{11} and r_{13} are unreliable. The truth values of c and $\neg c$ are unknown because the truth value of OK_I1 is unknown. The other stable r-models of P are: $M_1 = coh(a, \neg b, d, e, OK\_I2, OK\_A1)$ and $M_2 = coh(a, \neg b, \neg c, d, e, OK\_I1, OK\_I2)$. If we extend $<_R$ with $r_{11} < r_{13}$, indicating that gate A1 is more reliable than gate I1 then the unique stable r-model of the new program equals M_1.

Let P' be as P with $S_r = \{\}, \forall$ rule r. All rules in P' except r_7 and r_{10} are reliable w.r.t. $\emptyset$. Consequently, $RM_{P'} = \{a, \neg b, \neg c, d, OK\_I1, OK\_I2, OK\_A1\}$ indicating that all gates are working correctly but the truth value of output e is unknown. The other stable r-models of P' are: $M'_1 = coh(\{a, \neg b, \neg c, d, e, OK\_I1, OK\_I2, OK\_A1\})$ and

$M'2 = coh(\{a, \neg b, \neg c, d, \neg e, OK\_I1, OK\_I2, OK\_A1\})$.

Model M'_1 indicates that output e has value 1 and that all gates are working correctly. This is an unintuitive result because if all gates are working correctly then output e should have value 0. Model M'_2 indicates that output e has value 0. This is also an unintuitive result because rule r_{10}, which expresses that the observed value of e is 1, has higher priority than rules r_{11}, r_{12} and r_{13}. The same reliable model and stable r-models are derived when P' is extended with $r_{11} < r_{13}$. The reason for these unintuitive results is that $S_r = \{\}, \forall$ rule r in P', even though the rules r_i are complete, for all $i \leq 10$. When a rule r_i, for $i \leq 10$, is in conflict with an observed output, the truth value of any literal in $Body_{r_i}$ may be mistaken. For this, S_{r_i} should be equal to $Body_{r_i}$, for all $i \leq 10$.

4 Related Work

The reliable semantics for $EPPs$ is a generalization of the 3-valued stable model semantics which is defined for normal programs [27].

Proposition 4.1 Let P be a normal program and M a set of classical literals. Then, M is a 3-valued stable model of P iff $M \cup \{\sim \neg A |\ A$ is an atom of $P\}$ is a stable r-model of P.

Proposition 4.1 implies that the reliable model of a normal program coincides with its well-founded model [34].

In [12], the *answer-set semantics* of an extended program is defined as the intersection of its answer-sets. However, the answer set semantics is not defined for all extended programs and can be contradictory. Moreover, the problem of finding whether an extended program has an answer set is NP-complete [7]. The following relationship between the answer-set semantics and RS can be shown.

Proposition 4.2 Let P be an extended program. If $M \neq HB_P$ is an answer-set of P then $M \cup \{\sim A | A \notin M\}$ is a stable r-model of P.

Let P be an extended program and I an interpretation of it. In [24], the operator Φ_P is defined as $\Phi_P(I) = coh(least_v(P/I))$ if $least_v(P/I)$ does not contain a pair of complementary literals. Otherwise, $\Phi_P(I)$ is not defined. The *extended well-founded model* ($XWFM_P$) of P is defined in [24] as the least fixpoint of Φ_P. An *extended stable model* of P is a fixpoint of Φ_P. Let

$I_0 = \{\}, I_{a+1} = \Phi_P(I_a)$ and $I_a = \cup \, \{I_b | \; b < a\}$ if a is a limit ordinal. Then, $XWFM_P = I_d$ where d is the smallest ordinal s.t. $I_{d+1} = I_d$. When there exists an a s.t. $\Phi_P(I_a)$ is not defined, P is called *contradictory*.

Proposition 4.3 Let P be a non-contradictory extended program. Then, M is an extended stable model of P iff M is a stable r-model of P.

Proposition 4.3 implies that if P is a non-contradictory extended program then the reliable model of P coincides with the extended well-founded model of P. Since P is a non-contradictory extended program, $XWFM_P = Pos_\emptyset$ and there is no L s.t. $L \in Pos_\emptyset$ and $\neg L \in Pos_\emptyset$. This implies that every default literal and rule in P is reliable w.r.t. $\emptyset$. So, $RM_P = XWFM_P = Pos_\emptyset$. When P is contradictory, the extended well-founded semantics of P is not defined in contrast to the reliable semantics. For example, consider the extended program $P = \{\neg p. \quad p. \quad b.\}$. P has three stable r-models $\{b, \sim \neg b\}, \{\neg p, b, \sim p, \sim \neg b\}$ and $\{p, b, \sim \neg p, \sim \neg b\}$ but no extended well-founded semantics.

In [36], a program P with constraints is called *revisable* if it has a Δ-model, that is, if there is a consistent interpretation I s.t. for every rule r in $P, M(Head_r) \geq M(Body_r)$. In [36], it is shown that every revisable program P whose well-founded model [34] is inconsistent, can be expanded into a new program P' that has consistent well-founded model and the same Δ-models as P. The semantics of P is defined as the well-founded model of P'. However, the well-founded model of P' may not be a coherent interpretation of P. Moreover, when P is not revisable, P is not given any semantics in [36]. For example, the program P of Example 2.1 with $<_R = \{\}$ is not revisable but it has three stable r-models.

Let P be an extended program. In [37], the conservative and skeptical models of P are defined, expressing different degrees of "skepticism" towards contradictory information. A set $S \subseteq HB_P$ is a conservative (resp. skeptical) stable model of P iff the unique conservative (resp. skeptical) model of P/S equals S. The *conservative (resp. skeptical) vivid logic semantics* of P is the intersection of the conservative (resp. skeptical) stable models of P. For example, the unique conservative or skeptical stable model of $P = \{p. \quad \neg p. \quad a \leftarrow \sim p.\}$ is $M = \{a\}$. According to [37], a default literal $\sim L$ is true w.r.t. a model M iff $L \notin M$. So, $\sim p$ is true w.r.t. M and $P/M = \{p. \quad \neg p. \quad a.\}$. The conservative or skeptical model of P/M is $\{a\}$ which equals M. However, intuitively $\sim p$ should not be true because of the rule $p \leftarrow$ in P. In RS, $\sim p$ is undefined. Because in vivid logic, only 2-valued models are considered, there are extended programs with no conservative or skeptical stable model.

The problem is similar to that of the stable model semantics [11] and answer set semantics [12].

Prioritization of defaults is considered in [28]. There, a default is a formula containing only the classical connectives $\neg$ and $\leftarrow$. First, the *most reliable consistent set of premisses* is defined when rules are totally ordered. Then, the semantics of a program is defined as the intersection of all classical models of the most reliable consistent set of premisses for all linear extensions of $<_R$. However, the number of linear extensions of $<_R$ can be exponentially large. For example, the number of linear extensions of $<_R = $ is $n!$, where n is the number of defaults.

Prioritization of rules is also investigated in [9, 18]. An *ordered logic program* is a partially-ordered set of rules without negation by default. Even though the *c-assumption-free semantics* [9] and *assumption-free semantics* [18] are defined for all ordered logic programs, negation by default is not supported and only the basic constraints are considered. The *skeptical c-partial model* of an ordered logic program P is defined in [9] as follows: A literal set S is *c-unfounded* w.r.t. an interpretation I iff $\forall L \in S$, if r is a rule in P with $Head_r = L$ then either (i) $\exists$ rule r' s.t. $r' \not< r, Head_{r'} = \neg Head_r$ and $Body_{r'} \cup Head_{r'} \subseteq I$ or (ii) $Body_r \cap S \neq \emptyset$. A rule r is *c-defeasible* w.r.t. I iff $\exists$ rule r' s.t. $r' \not< r, Head_{r'} = \neg Head_r$ and $(Body_{r'} \cup Head_{r'}) \cap U^c(I) = \emptyset$, where $U^c(I)$ is the greatest c-unfounded set w.r.t. I. The *sceptical c-partial model* of P is the least fixpoint of the monotonic operator $S(I) = \{L| \ \exists$ rule r in P s.t. $Head_r = L, Body_r \subseteq I$ and r is not c-defeasible w.r.t. $I\}$.

The *skeptical c-partial model* of the program P in Example 3.3 is $\{a, \neg b, \neg c, d, OK\_I1, OK\_I2, OK\_A1\}$. This is because rules r_7 and r_{10} are the only c-defeasible rules w.r.t. $\emptyset$ in P. According to this model, all gates are working correctly but the truth value of the output e is unknown. This unintuitive result is derived because in [9], the rule ordering $r' < r$ represents that rule r is an *exception* of rule r'. This corresponds in our framework with the case that $S_r = \{\}, \forall$ rule r. In Example 3.3, we showed that if $S_r = \{\}, \forall$ rule r then rules r_7 and r_{10} are the only unreliable rules w.r.t. $\emptyset$ and the reliable model equals the skeptical c-partial model of P. The next proposition shows that the reliable model of every ordered logic program, P, coincides with the skeptical c-partial model of P.

Proposition 4.4 Let $P =< R_P, IC_P, <_R>$ be an EPP which is free from default literals, $S_r = \{\}, \forall$ rule r, and $IC_P = BC_P$. Then, M is the skeptical c-partial model of P [9] iff M is the set of classical literals in RM_P.

In [18], the *well-founded partial model* of an ordered logic program P is defined. Similarly to [9], rule ordering in [18] represents *exceptions* and not *reliability.* This corresponds in our framework with the case that $S_r = \{\}, \forall$ rule r. Indeed, the reliable model of the program P in Example 3.3 with $S_r = \{\}, \forall$ rule r, is the same as the well-founded partial model of P. Another difference between the reliable model and the well-founded partial model is demonstrated by the following example: The well-founded partial model of $P = \{p. \quad \neg p \leftarrow q. \quad \neg q. \quad q.\}$ is $\{p\}$. According to this model, p is *true* even though $\neg p$ can also be derived from P. This is because rule $p \leftarrow$ is not considered defeasible. According to [18], the literal q is ambiguous and thus the derivation of q and $\neg p$ is blocked. In [32], a similar ambiguity blocking approach applied to inheritance networks was severely questioned. In our approach, ambiguities are propagated and thus rule $p \leftarrow$ is considered unreliable. Note that $RM_P = \{\}$, independently of the values of S_r.

Proposition 4.5 shows that the reliable semantics is more skeptical than the assumption-free semantics of [18]. The proposition follows immediately from Proposition 4.4 and the fact that the skeptical c-partial model of an ordered logic program P is a subset of the well-founded partial model of P [Theorem 8, [9]].

Proposition 4.5 Let $P =< R_P, IC_P, <_R>$ be an EPP which is free from default literals, $S_r = \{\}, \forall$ rule r, and $IC_P = BC_P$. Then, the set of classical literals in RM_P is a subset of the well-founded partial.model of P [18].

The rule ordering $<_R$ in RS expresses that in case of conflict, one rule is considered more reliable than another. Saying that r is more reliable than r' is different than saying that r is an exception to r'. Let $r : L \leftarrow L_1, \ldots, L_n$ and $r' : L' \leftarrow L'_1, \ldots, L'_m$ be two rules. The fact that r is an exception of r' can be expressed by replacing the old rule r' with $r' : L' \leftarrow L'_1, \ldots, L'_n, \sim name_r$ and by adding the rule: $name_r \leftarrow L_1, \ldots, L_m$ [22, 25]. For example, let $r : \neg flies(X) \leftarrow penguin(X)$ and $r' : flies(X) \leftarrow bird(X)$. The fact that r is an exception of r' is represented by replacing the old rule r' with $r' : flies(X) \leftarrow bird(X), \sim nf(X)$ and adding the rule $nf(X) \leftarrow penguin(X)$. The relation $<_R$ is extended as follows: The added rule has lower (resp. higher) priority than a rule r'' iff $r < r''$ (resp. $r'' < r$).

5 Conclusions

We have presented a new semantics for *extended programs with rule prioritization* (EPP). The semantics, called *reliable semantics* (RS), is a generalization of the well-founded semantics for normal programs [34] and extended well-founded semantics for non-contradictory extended programs [24]. RS is contradiction-free, coherent and defined for all EPPs. The *reliable model* of a program P is the least stable r-model of P and it represents the skeptical "meaning" of P. Stable r-models of P represent possible "meanings" of P. The degree of "skepticism" in RS depends on the preliminary suspect sets of its rules. If P is an ordered logic program then the RS of P coincides with the *skeptical c-partial model* of P [9] and is a subset of the *well-founded partial model* of P [18]. When the Herbrand base of an EPP is finite, the complexity of computing RS of P is polynomial w.r.t. the size of the program.

RS can apply to deductive object-oriented databases ($DOODs$). Several works combine object-oriented and logic programming, including [16, 2, 20, 15]. In $DOODs$, rule prioritization can be used (i) to express that specific rules are more reliable than general ones, (ii) to give priorities to inconsistent inherited rules in case of multiple inheritance and (iii) to give priorities to inconsistent class rules in case of multiple specializations of the same object.

Another application of RS is deriving trustworthy information from multiple sources of information that are not fully reliable. For example, when the knowledge bases (KBs) of different scientific labs are combined, conflicts are bound to occur because of measurement mistakes and imperfect techniques. Work on combining deductive databases has been done in [1]. In [1], when a constraint $\perp \leftarrow L_1, \ldots, L_n$ is violated the disjunction $L_1 \vee \ldots \vee L_n$, is added and the rules with head $L_i, i \leq n$, are removed from the KB. The maximum information is saved this way. However, it is possible that literals $L_1, \ldots, L_n$ are based on unreliable information which will continue to be true. In RS, not only the literals $L_i, i \leq n$, are considered "suspect" for the violation of the constraint but also the literals used in the derivation of $L_i, i \leq n$.

Acknowledgements: We thank the referees for their helpful comments.

APPENDIX: Proofs

Proposition 2.2 Let $P =< R_P, IC_P, <_R>$ be an EPP and $<'_R$ be an extension of $<_R$. Every r-model of $P' =< R_P, IC_P, <'_R>$ is an r-model of P.

Proof Let M be an r-model of P'. Then, M is a consistent, coherent interpretation of P. If r is a rule in P then r is r-true w.r.t. M in P'. We will show that r is r-true w.r.t. M in P. It is enough to show that if r is unreliable w.r.t. M in P' then r is unreliable w.r.t. M in P. Assume that r is unreliable w.r.t. M in P'. Since $<'_R$ is an extension of $<_R$, the set of rules with priority no lower than r in P' is a subset of that in P. So, $Pos_M, Pos_{r,M}$ and $Dep_{r,M}(L)$ in P' are subsets of the corresponding sets in P, for every literal L. This implies that r is unreliable w.r.t. M in P.

Proposition 3.1 Let P be an EPP. $\{I_a\}$ is a monotonically increasing (w.r.t. $\subseteq$) sequence of consistent, coherent interpretations of P.

Proof We will show that W_P is a monotonic operator. Let I, J be interpretations of P s.t. $I \subseteq J$. Then, $T(I) \subseteq T(J)$ because if a rule r is reliable w.r.t. I then r is reliable w.r.t. J. $F(I) \subseteq F(J)$ because if a default literal $\sim L$ is reliable w.r.t. I then $\sim L$ is reliable w.r.t. J. Since coh is a monotonic operator, W_P is a monotonic operator and I_a is a monotonically increasing sequence w.r.t. $\subseteq$.

We will prove by induction that for all a, there is no constraint ic' s.t. $Body_{ic'} \subseteq I_a$. This is true for $a = 0$. Assume that it is true for ordinals $< a$. We will prove that it is true for a.

Assume first that $a = b + 1$ is a successor ordinal. Let $a' = b' + 1$ be the first ordinal s.t. $T_{I_b}^{\uparrow a'}(\emptyset)$ is inconsistent. If $T_{I_b}^{\uparrow a'}(\emptyset)$ violates a basic constraint then let ic be one of the violated basic constraints. Otherwise, let ic be any of the violated constraints. Let $R_{b,a'} = \{r |$ rule r is fired in the computation of $T(I_c)$ for $c < b$ or in the computation of $T_{I_b}^{\uparrow c}(\emptyset)$ for $c \leq a'\}$. Let $K \in Body_{ic}$ s.t. $K \in T_{I_b}^{\uparrow a'}(\emptyset)$, $K \notin I_b$ and $K \notin T_{I_b}^{\uparrow b'}(\emptyset)$. Such a literal K exists since a' is the first ordinal s.t. $T_{I_b}^{\uparrow a'}(\emptyset)$ is inconsistent.

Case 1: The constraint ic is the basic constraint $\bot \leftarrow K, \neg K$.

Choose the smallest c' s.t. $I_{c'+1}$ contains $\neg K$. We will show that there is no L s.t. $L \in F(I_c)$, $c < c'$, and $\sim L \in Dep_{I_c}(\neg K)$. Since $\forall c < c', \neg K \notin I_c$ and there is no literal K' s.t. $K' \in I_c$ and $\neg K' \in T_{I_b}^{\uparrow b'}(\emptyset)$, it follows that $\forall c < c', K \in Pos_{I_c}$. If there are L, c s.t. $L \in F(I_c)$, $c < c'$, and $\sim L \in Dep_{I_c}(\neg K)$ then $\sim L$ is unreliable w.r.t. I_c which is a contradiction (all literals in $\sim F(I_c)$ are reliable w.r.t I_c). Similarly, there is no L s.t. $L \in F(I_c)$, $c < b$, and $\sim L \in Dep_{I_c}(K)$. This implies that there is rule $r_m \in R_{b,a'}$ which is used in the derivation of K, $\neg K \in Pos_{r_m}, I_b$ and $Head_{r_m} \in Dep_{r_m}, I_b(K)$. Moreover, $S_{r_m} \subseteq Pos_{r_m}, I_b$ and $Body_{r_m} - S_{r_m} \subseteq Pos_{I_b}$. Thus, rule r_m is unreliable w.r.t. I_b, which is a contradiction.

Case 2: The constraint ic is not a basic constraint.

Then, there is no literal K' s.t. $K' \in I_c$, $c < b$, and $\neg K' \in T_{I_b}^{\uparrow a'}(\emptyset)$. This implies that $\forall c < b, T_{I_b}^{\uparrow a'}(\emptyset) \subseteq Pos_{I_c}$ and consequently, $Body_{ic} \subseteq Pos_{I_c}$. We will show that there is no L s.t. $L \in F(I_c)$, $c < b$, and $\sim L \in Dep_{I_c}(K)$, for a $K \in Body_{ic}$. Assume that there are L, c s.t. $L \in F(I_c)$, $c < b$, and $\sim L \in Dep_{I_c}(K)$, for a $K \in Body_{ic}$. Then, $Body_{ic} \subseteq Pos_{I_c}$ and $\sim L \in Dep_{I_c}(K)$, for a $K \in Body_{ic}$. Consequently, $\sim L$ is unreliable w.r.t. I_c which is a contradiction (all literals in $\sim F(I_c)$ are reliable w.r.t. I_c). Let R_{ic} be the set of the rules $r \in R_{b,a'}$ which are used in the derivation of literals in $Body_{ic}$ and $Head_r \in Dep_{r,I_b}(K)$, for a $K \in Body_{ic}$. Let $r_m \in R_{ic}$ be s.t. there is no $r \in R_{ic}$ and $r < r_m$. Then, $Body_{ic} \subseteq Pos_{r_m,I_b}$ and $Head_{r_m} \in Dep_{r_m,I_b}(K)$, for a $K \in Body_{ic}$. Moreover, $S_{r_m} \subseteq Pos_{r_m,I_b}$, $Body_{r_m} - S_{r_m} \subseteq Pos_{I_b}$. Consequently, rule r_m is unreliable w.r.t. I_b, which is a contradiction.

So, I_a does not violate any constraint.

Let a be a limit ordinal and assume that there is constraint ic in P s.t. $Body_{ic} \subseteq I_a$. Then, there is a successor ordinal $b+1 < a$ s.t. $Body_{ic} \subseteq I_{b+1}$. This is a contradiction because of the inductive hypothesis. So, I_a is consistent for all a.

We will prove by induction that for all a, there is no literal L s.t. $L \in I_a$ and $\sim L \in I_a$. The proof is similar to that of the well-founded semantics [34]. It is true for $a = 0$. Assume that it is true for ordinals $< a$. We will prove that it is true for a.

Assume first that $a = b+1$ is a successor ordinal. We will prove by a second induction that for all a', there is no literal L s.t. $L \in T_{I_b}^{\uparrow a'}(\emptyset)$ and $\sim L \in I_a$. This is true for $a' = 0$. Assume this is true for ordinals $< a'$. Let $a' = b'+1$ is a successor ordinal. Let S be any set of classical literals that has a non-empty intersection with $T_{I_b}^{\uparrow a'}(\emptyset)$. Choose the smallest c s.t. I_{c+1} has a non-empty intersection with S and the smallest c' s.t. $T_{I_c}^{\uparrow c'+1}(\emptyset)$ has a non-empty intersection with S. Note that $c < b$ or $c = b$ and $c' < a'$. Let $L \in T_{I_c}^{\uparrow c'+1}(\emptyset) \cap S$. Then, L is derived from a rule r s.t. $Body_r \subseteq T_{I_c}^{\uparrow c'}(\emptyset) \cup I_c$. From hypothesis, there is no literal $K \in Body_r$, s.t. $\sim K \in I_b$. Moreover, from the way r is defined, there is no classical literal K in $Body_r$ s.t. $K \in S$. So, S is not r-unfounded w.r.t. I_b. This implies that $T_{I_b}^{\uparrow a'}(\emptyset) \cap F(I_b) = \emptyset$. So, $T(I_b) \cap F(I_b) = \emptyset$. Moreover, there is no classical literal L s.t. $L \in T(I_b)$ and $\neg L \in T(I_b)$, because I_a does not violate any constraint. So, there is no literal L s.t. $L \in I_a$ and $\sim L \in I_a$.

Let a be a limit ordinal and assume that there is L s.t. $L \in I_a$ and $\sim L \in I_a$. Then, there is a successor ordinal $b+1 < a$ s.t. $L \in I_{b+1}$ and $\sim L \in I_{b+1}$. This is a contradiction because of the inductive hypothesis.

I_a is a coherent interpretation, for all a, because of the *coh* operator in the definition of W_P. Proposition 3.1 follows.

Proposition 3.2 Let P be an EPP. Then, I_d is an r-model of P.

Proof From Proposition 3.1, I_d is a consistent, coherent interpretation. Let r be a rule in P. We will show that r is r-true w.r.t. I_d.

1. If $I_d(Body_r) = 1/2$ and $I_d(Head_r) = 0$ then $I_d(\neg Head_r) = 1$ because otherwise $I_d(Head_r) = 1/2$.

2. If $I_d(Body_r) = 1$ and $I_d(Head_r) = 1/2$ then r is unreliable w.r.t. I_d because otherwise, from the definition of $T(I_d), I_d(Head_r) = 1$.

3. If $I_d(Body_r) = 1$ and $I_d(Head_r) = 0$ then r is unreliable w.r.t. I_d because otherwise, from the definition of $T(I_d), I_d(Head_r) = 1$. Since r is unreliable w.r.t. I_d, it follows that $I_d(\neg Head_r) = 1$ because otherwise, from the definition of $F(I_d), I_d(Head_r) = 1/2$.

4. In all the other cases, $I_d(Head_r) \geq I_d(Body_r)$.

Proposition 3.3 Let $P =< R_P, IC_P, <_R>$ be an EPP. The complexity of computing RM_P is $O(|HB_P| * |R_P| * max(|IC_P|, |HB_P| * |EC_R|)$, where EC_R is the set of equivalence classes of R_P w.r.t. $\equiv_R$.

Proof The following algorithm, **RM**(program P), returns the reliable model of P. To compute $F(I)$, its complement set is constructed first, as in [34].

RM(program P)
{ $new\_I = \{\}$;
  repeat
    $I = new\_I$;
    compute Pos_I;   /* Step 1 */
    for each $L \in HB_P$ **do** compute $Dep_I(L)$; **endfor**   /* Step 2 */
    for each $[r]$ in P **do**   /* Step 3 */
      compute $Pos_{r,I}$;   /* Step 3.1 */
      for each $L \in HB_P$ **do** compute $Dep_{r,I}(L)$; **endfor**   /* Step 3.2 */
    endfor

    repeat   /* Step 4: Compute T(I) */
      for each rule r in P **do**
        if $Body_r \subseteq new\_I$ and r is reliable w.r.t. I **then** add $Head_r$ to new_I; **endif**
      endfor
    until no change in new_I;

    $compl\_F = \{L \in HB_P | \sim L$ is unreliable w.r.t. $I\}$;   /* Step 5 */

```
    repeat    /* Step 6: Compute HB_P − F(I) */
      for each rule r in P do
        if no literal in Body_r is false w.r.t. I and all classical literals in Body_r are in compl_F
        then add H_r to compl_F;
        endif
      endfor
    until no change in compl_F;
    for each L ∈ HB_P do    /* Step 7*/
      if L ∉ compl_F then add ∼ L to new_I; endif
    endfor

    new_I = coh(new_I);    /* Step 8: Compute coh(T(I) ∪ ∼ F(I)) */
  until I = new_I;
  return(I);
}
```

The complexity of computing Pos_I is the same as that of computing the well-founded model of P when every literal L is replaced by a new atom $\neg\_L$. So, the complexity of Step 1 is $|HB_P| * |R_P|$ [35, 31]. The complexity of Step 2 is $|HB_P| * |R_P|$ because the complexity of computing $Dep_I(L)$, for a literal L, is $|R_P|$. The complexity of Step 3.1 is $|R_P|$ and that of Step 3.2 is $|HB_P| * |R_P|$. So, the complexity of Step 3 is $|EC_R| * |HB_P| * |R_P|$. The complexity of Step 4 is $|IC_P| * |R_P|$ since $Pos_{r,I}$ and $Dep_{r,I}(L), \forall L \in HB_P$, have already been computed. The complexity of Step 5 is $|IC_P| * |HB_P| < |IC_P| * |R_P|$ and that of Step 6 is $|R_P|$ [4]. The complexity of Steps 7 and 8 is $|HB_P|$. Since $\{I_a\}$ is a monotonically increasing sequence w.r.t. $\subseteq$, the total number of iterations until $I = new\_I$, is less than $|HB_P|$. So, the complexity of the algorithm $RM(P)$ is $O(|HB_P| * |R_P| * max(|IC_P|, |HB_P| * |EC_R|)$.

Proposition 3.4 Let P be an EPP. The reliable model of P is a stable r-model of P.

Proof Let RM be the reliable model of P. From Proposition 3.2, RM is an r-model of P. So, it is enough to show that $RM = least_v(P/_rRM)$. Let $least_v(P/_rRM) = T \cup \sim F$, where T, F are sets of classical literals. Let T_a, F_a be sets of classical literals s.t. $I_a = T_a \cup \sim F_a$. Let $RM = I_d$. First, we will prove by induction that $T_b \cup \sim F_b \subseteq T \cup \sim F, \forall b \leq d$. It is true that $T_0 \subseteq T$ and $F_0 \subseteq F$. Suppose that $T_a \subseteq T$ and $F_a \subseteq F, \forall a < b$. If b is a limit ordinal then $T_b \subseteq T$ and $F_b \subseteq F$ since $I_b = \cup\{I_a | \; a < b\}$. Assume therefore that $b = a + 1$. It is true that $T_{I_a}^{\uparrow 0}(\emptyset) \subseteq T$. Assume that $T_{I_a}^{\uparrow a'}(\emptyset) \subseteq T$, we will show that $T_{I_a}^{\uparrow a'+1}(\emptyset) \subseteq T$. Let $L \in T_{I_a}^{\uparrow a'+1}(\emptyset)$. Then, $\exists r : L \leftarrow L_1, \ldots, L_n$ in P s.t. r is reliable w.r.t. I_a and $\forall i \leq n$ either (i) $L_i \in I_a$ or (ii) L_i is a classical literal and $L_i \in T_{I_a}^{\uparrow a'}(\emptyset)$. Since $I_a \subseteq T \cup \sim F$ and $L \in RM$, there is a rule $L \leftarrow L'_1, \ldots, L'_m$ in $P/_rRM$ where

$L'_1, \ldots, L'_m$ are all the classical literals in $L_1, \ldots, L_n$. From the facts $T_{I_a}^{\uparrow a'}(\emptyset) \subseteq T, I_a \subseteq T \cup \sim F$ and the definition of $least_v(P/_rRM)$, it follows that $L \in T$. This implies that $T(I_a) = T_b \subseteq T$.

Now, we will show that $F_b \subseteq F$. Since $F_b = \neg T_b \cup F(I_a)$, it is enough to show that $\neg T_b \subseteq F$ and $F(I_a) \subseteq F$. If $L \in \neg T_b$ then $\neg L \in RM$ and from Step (2) of Def. 3.4, $L \in F$. Consequently, $\neg T_b \subseteq F$. For all rules $H \leftarrow L'_1, \ldots, L'_m, \sim L_1, \ldots, \sim L_n$ in P (L_i, L'_i are classical literals) with $H \in F(I_a)$ either $\exists i \leq m, L'_i \in F(I_a) \cup F_a$ or $\exists j \leq n, L_j \in T_a$. This implies that, for each rule $H \leftarrow L'_1, \ldots, L'_m, \sim L_1, \ldots, \sim L_n$ in P with $H \in F(I_a)$ either there is a corresponding rule $H \leftarrow A_1, \ldots, A_k$ in $P/_rRM$ (from Steps (4),(5) of Def. 3.4) with $A_i \in F(I_a) \cup F$ for an $i \leq k$ or there is no corresponding rule in $P/_rRM$ (from Steps (1),(2) of Def. 3.4). Note that, no rule $H \leftarrow u$ is added to $P/_rRM$ (from Steps (3),(6) of Def. 3.4) because H is false w.r.t. RM. So, for each rule $H \leftarrow A_1, \ldots, A_k$ in $P/_rRM$ with $H \in F(I_a) \cup F, \forall i \leq k$ such that $A_i \in F(I_a) \cup F$. From the definition of $least_v(P/_rRM)$, it follows that $F(I_a) \subseteq F$. Consequently, $F_b \subseteq F$. So, we proved that $T_d \subseteq T$ and $F_d \subseteq F$.

We will show that $T \subseteq T_d$. Let a be the first ordinal s.t. there is a literal $L \notin T_d$ and $\Psi_{P'}^{\uparrow a+1}(\emptyset)(L) = 1$, where $P' \equiv P/_rRM$. Then, there is a rule $r : L \leftarrow A_1, \ldots, A_k$ in $P/_rRM$ with $\Psi_{P'}^{\uparrow a}(\emptyset)(A_i) = 1, \forall i \leq k$. This implies that there is a rule in P whose body literals are true w.r.t. RM. Since $L \notin T_d$, it follows that $\neg L \in T_d$ or L is unknown w.r.t. RM. If $\neg L \in T_d$ then from Step (2) of Def. 3.4, $L \notin T$ which is a contradiction. If L is unknown w.r.t. RM, the rule r should not exist in $P/_rRM$ because of the Step (3) of Def. 3.4 and the fact that all of the body literals of r are true w.r.t. $\Psi_{P'}^{\uparrow a}(\emptyset)$ and thus w.r.t. RM. So, $L \in T_d$ and consequently $T \subseteq T_d$.

We will show that $F \subseteq F_d$. Let $F_{coh} = \{H | \neg H \in T_d\}$. $F_{coh} \subseteq F_d$ because RM is a coherent interpretation. For all rules $H \leftarrow A_1, \ldots, A_k$ in $P/_rRM$ with $H \in F - F_{coh}$, there is $i \leq k$ such that $A_i \in F$. This implies that for each rule $H \leftarrow L'_1, \ldots, L'_m, \sim L_1, \ldots, \sim L_n$ in P (L_i, L'_i are classical literals) with $H \in F - F_{coh}$ either (i) $\exists i \leq m, L'_i \in F$ (from Steps (4),(5) of Def. 3.4) or (ii) $\exists j \leq n, L_j \in T_d$ (from Step (1) of Def. 3.4). We will show that $\forall H \in F - F_{coh}$, $\sim H$ is reliable w.r.t. RM. If $\sim H$ is unreliable w.r.t RM then $H \notin F(I_d)$ and consequently, $RM(H) \geq 1/2$. However, if $H \in F$ then $H \notin T$ and consequently $RM(H) \neq 1$. So, $RM(H) = 1/2$ and the rule $H \leftarrow u$ should be added to $P/_rRM$ (from Step (6) of Def. 3.4). This implies that $H \notin F$, which is a contradiction. So, $\forall H \in F - F_{coh}$, $\sim H$ is reliable w.r.t. RM. Since $F(I_d)$ is the maximum set that satisfies the property satisfied by $F - F_{coh}, F - F_{coh} \subseteq F(I_d)$. So, $F \subseteq F_d$. Consequently, $RM = T_d \cup \sim F_d = T \cup \sim F = least_v(P/_rRM)$.

Proposition 3.5 Let P be an EPP. The reliable model of P is the least stable r-model of P.

Proof Let RM be the reliable model of P. From Proposition 3.4, RM is a stable r-model of P. So, it is enough to show that if M is a stable r-model of P then $RM \subseteq M = least_v(P/_rM)$. Let $M = T \cup \sim F$, where T, F are sets of classical literals. Let T_a, F_a be sets of classical literals s.t. $I_a = T_a \cup \sim F_a$. Let $RM = I_d$. We will show by induction that $I_b \subseteq T \cup \sim F, \forall b \leq d$. It is true that $T_0 \subseteq T$ and $F_0 \subseteq F$. Suppose that $T_a \subseteq T$ and $F_a \subseteq F, \forall a < b$. If b is a limit ordinal then $T_b \subseteq T$ and $F_b \subseteq F$ since $I_b = \cup\{I_a | a < b\}$. Assume therefore that $b = a+1$. It is true that $T_{I_a}^{\uparrow 0}(\emptyset) \subseteq T$. Assume that $T_{I_a}^{\uparrow a'}(\emptyset) \subseteq T$, we will show that $T_{I_a}^{\uparrow a'+1}(\emptyset) \subseteq T$. Let $L \in T_{I_a}^{\uparrow a'+1}(\emptyset)$. Then, $\exists r : L \leftarrow L_1, \ldots, L_n$ in P s.t. r is reliable w.r.t. I_a and $\forall i \leq n$ either (i) $L_i \in I_a$ or (ii) L_i is a classical literal and $L_i \in T_{I_a}^{\uparrow a'}(\emptyset)$. Since $I_a \subseteq M$, it follows that r is reliable w.r.t. M. From the facts that M is an r-model of P, $I_a \subseteq M$, $T_{I_a}^{\uparrow a'}(\emptyset) \subseteq T$ and r is reliable w.r.t. M, it follows that $L \in T$. So, $T(I_a) = T_b \subseteq T$.

Now, we will show that $F_b \subseteq F$. Since $F_b = \neg T_b \cup F(I_a)$, it is enough to show that $\neg T_b \subseteq F$ and $F(I_a) \subseteq F$. If $\neg L \in T_b$ then $\neg L \in M$ and from Step (2) of Def. 3.4, $L \in F$. Consequently, $\neg T_b \subseteq F$. For all rules $H \leftarrow L'_1, \ldots, L'_m, \sim L_1, \ldots, \sim L_n$ in P (L_i, L'_i are classical literals) with $H \in F(I_a)$ either $\exists i \leq m$, $L'_i \in F(I_a) \cup F_a$ or $\exists j \leq n, L_j \in T_a$. This implies that, for each rule $r : H \leftarrow L'_1, \ldots, L'_m, \sim L_1, \ldots, \sim L_n$ in P with $H \in F(I_a)$ either there is a corresponding rule $H \leftarrow A_1, \ldots, A_k$ in $P/_rM$ (from Steps (4),(5) of Def. 3.4) with $A_i \in F(I_a) \cup F$ for an $i \leq k$ or there is no corresponding rule in $P/_rM$ (from Steps (1),(2) of Def. 3.4). Note that, r is not transformed into $H \leftarrow u$ in $P/_rM$ in Step (3) of Def. 3.4 because the facts $I_a \subseteq M$ and $least_v(P/_rM) = M$ imply that $\exists i \leq m, L'_i \notin T$ or $\exists j \leq n, L_j \notin F$. Moreover, no rule $H \leftarrow u$ with $H \in F(I_a)$ is added to $P/_rM$ in Step (6) of Def. 3.4 because the facts $\sim H$ is reliable w.r.t. I_a and $I_a \subseteq M$ imply that $\sim H$ is reliable w.r.t. M. So, for each rule $H \leftarrow A_1, \ldots, A_k$ in $P/_rM$ with $H \in F(I_a) \cup F$, $\exists i \leq k$ s.t. $A_i \in F(I_a) \cup F$. From the definition of $least_v(P/_rM)$, it follows that $L \in F$. So, $F_b \subseteq F$ and thus $T_d \subseteq T$ and $F_d \subseteq F$. Consequently, $RM = T_d \cup \sim F_d \subseteq T \cup \sim F = M$.

Proposition 3.6 Let $P =< R_P, IC_P, <_R>$ be an EPP and $<'_R$ an extension of $<_R$. Every stable r-model of $P' =< R_P, IC_P, <'_R>$ is a stable r-model of P and $RM_P \subseteq RM_{P'}$.

Proof Let M be a stable r-model of P'. We will show that every default literal which is unreliable w.r.t. M in P' is also unreliable w.r.t. M in P. Assume that $\sim L$ is unreliable w.r.t. M in P'. Since $<'_R$ is an extension of $<_R$, the set of rules with priority no lower than r in P' is a subset of that in P. So, Pos_M and $Dep_M(K)$, for a literal K, in P' are subsets of the corresponding sets in P and consequently $\sim L$ is unreliable w.r.t. M in P.

Let $S = \{L|\ M(L) = 1/2$ and $\sim L$ is unreliable w.r.t. $M\}$. For all $L \in S$, $least_v(P'/M)(L) = 1/2$ because $least_v(P'/M) = M$. This and the fact $P/_rM = P'/_rM \ \cup\ \{L \leftarrow u|\ L \in S\}$ imply that $least_v(P/M) = least_v(P'/M) = M$. From Proposition 2.2, M is an r-model of P. So, M is a stable r-model of P. Since RM_P is the least stable r-model of P, $RM_P \subseteq RM_{P'}$.

Proposition 4.1 Let P be a normal program and M a set of classical literals. Then, M is a 3-valued stable model of P iff $M \cup \{\sim \neg A|\ A$ is an atom of $P\}$ is a stable r-model of P.

Proof When P is a normal program, M is a 3-valued stable model of P iff $M \cup \{\sim \neg A|\ A$ is an atom of $P\}$ is an extended stable model of P [24]. Proposition 4.1 now follows from Proposition 4.3.

Proposition 4.2 Let P be an extended program. If $M \neq HB_P$ is an answer-set of P then $M \cup \{\sim A|\ A \notin M\}$ is a stable r-model of P.

Proof P is non-contradictory since $M \neq HB_P$ is an answer-set of P. So, if M is an answer-set of P then $M\ \cup \{\sim A|\ A \notin M\}$ is an extended stable model of P [24]. Proposition 4.2 now follows from Proposition 4.3.

Proposition 4.3 Let P be a non-contradictory extended program. Then, M is an extended stable model of P iff M is a stable r-model of P.

Proof Let M be an extended stable model of P. From the definition of extended stable model [24], M is an r-model of P and $least_v(P/_rM) = M$. So, M is a stable r-model of P.

Let M be a stable r-model of P. Since P is a non-contradictory extended program, there is no L s.t. $L \in Pos_\emptyset$ and $\neg L \in Pos_\emptyset$ and $XWFM_P = Pos_\emptyset$. So, all default literals and rules in P are reliable w.r.t. $I = \emptyset$. This implies that default literals and rules in P are reliable w.r.t. M. Consequently, Steps (3) and (6) of Definition 3.4 have no effect on $P/_rM$. So, from the definition of extended stable model, M is an extended stable model of P.

Proposition 4.4 Let $P =< R_P, IC_P, <_R>$ be an EPP which is free from default literals, $S_r = \{\}, \forall$ rule r, and $IC_P = BC_P$. Then, M is the skeptical c-partial model of P [9] iff M is the set of classical literals in RM_P.

Proof To simplify the proof, we redefine the operator $T(J)$ of Def. 3.2 as follows: $T(J) = \{L|\ \exists$ rule r in P s.t. $Head_r = L, Body_r \subseteq J$ and r is reliable w.r.t. $J\}$. Note that both definitions give equivalent semantics. Let $I_a = W_P^{\uparrow a}(\emptyset)$, for all a. We will show by induction that the set of classical literals in I_a coincides with $S^{\uparrow a}(\emptyset)$, for all a. This is true when $a = 0$. Suppose that it is true for all ordinals $\leq a$. We will show that the set of classical literals in I_{a+1} coincides with $S^{\uparrow(a+1)}(\emptyset)$. Since $S(I) = \{L | \exists$ rule r s.t. $Head_r = L, Body_r \subseteq I$ and r is not c-defeasible w.r.t. $I\}$, it is enough to show that for each rule r, $Body_r \subseteq I_a$ and r is reliable w.r.t. I_a iff $Body_r \subseteq S^{\uparrow a}(\emptyset)$ and r is not c-defeasible w.r.t. $S^{\uparrow a}(\emptyset)$.

$Body_r \subseteq I_a$ and r is reliable w.r.t. I_a
(From the inductive hypothesis and the fact that $Body_r$ is free of default literals, it follows that $Body_r \subseteq S^{\uparrow a}(\emptyset)$.)
iff $Body_r \subseteq S^{\uparrow a}(\emptyset)$ and rule r is reliable w.r.t. I_a
(From the fact $S_{r'} = \{\}, \forall$ rule r' and the definition of reliable rule, it follows that r is reliable w.r.t. I_a iff (i) there is no rule $r' \not< r$ with $Head_{r'} = \neg Head_r$ and $Body_{r'} \subseteq Pos_{I_a}$ or (ii) $Body_r$ is not a subset of Pos_{I_a}. Note that condition (ii) does not hold for r because $Body_r \subseteq S^{\uparrow a}(\emptyset) \subseteq I_a \subseteq Pos_{I_a}$.)
iff $Body_r \subseteq S^{\uparrow a}(\emptyset)$ and there is no rule $r' \not< r$ with $Head_{r'} = \neg Head_r$ and $Body_{r'} \cup Head_{r'} \subseteq Pos_{I_a}$
iff $Body_r \subseteq S^{\uparrow a}(\emptyset)$ and $\exists$ no $r' \not< r$ with $Head_{r'} = \neg Head_r$ and $(Body_{r'} \cup Head_{r'}) \cap (HB_P - Pos_{I_a}) = \emptyset$
(Let r' be a rule in P with $Body_{r'} \subseteq Pos_{I_a}$. Rule r' is blocked w.r.t. I_a iff $\neg Head_{r'} \in I_a$ iff $\exists r'' \not< r'$ with $Head_{r''} = \neg Head_{r'}$ and $Body_{r''} \cup Head_{r''} \subseteq S^{\uparrow a}(\emptyset)$. Consequently, $U^c(S^{\uparrow a}(\emptyset)) = HB_P - Pos_{I_a}$.)
iff $Body_r \subseteq S^{\uparrow a}(\emptyset)$ and $\exists$ no $r' \not< r$ with $Head_{r'} = \neg Head_r$ and $(Body_{r'} \cup Head_{r'}) \cap U^c(S^{\uparrow a}(\emptyset)) = \emptyset$
iff $Body_r \subseteq S^{\uparrow a}(\emptyset)$ and r is not c-defeasible w.r.t. $S^{\uparrow a}(\emptyset)$.

References

[1] C. Baral, S. Kraus, J. Minker, *Combining multiple knowledge bases*, IEEE Transactions on Knowledge and Data Engineering, 3(2), June 1991.

[2] S. Brass, U.W. Lipeck, *Semantics of Inheritance in Logical Object Specifications*, C. Delobel, M.Kifer, Y. Masunaga (eds.), Proc. of the 2nd Intern. Conference on Deductive and Object-Oriented Databases, 1991, pp.411-430.

[3] S. Brass, U.W. Lipeck, *Bottom-Up Query Evaluation with Partially Ordered Defaults*, Proc. of the 3rd Intern. Conference on Deductive and Object-Oriented Databases (DOOD'93), 1993.

[4] W. Dowling, J. Gallier, *Linear-Time Algorithms for Testing the Satisfiability of Propositional Horn Formulas*, Journal of Logic Programming, 3, (1984), pp. 267-284.

[5] P.M. Dung, P. Ruamviboonsuk, *Well-Founded Reasoning with Classical Negation*, A. Nerode, W. Marek, V.S. Subrahmanian (eds.), Proc. of the 1st Intern. Workshop on Logic Programming and Non-monotonic Reasoning, 1991, pp.120-132.

[6] P.M. Dung, *An Argumentation Semantics for Logic Programs with Explicit Negation*, Proc. of the 10th Intern. Conference on Logic programming (ICLP'93), 1993, pp.616-630.

[7] C. Elkan, *A Rational Reconstruction on Nonmonotonic Truth Maintenance System*, Artificial Intelligence, Elsevier Science Publishers, 43(2), 1990, pp. 219-234.

[8] M. Fitting, *A Kripke-Kleene Semantics for Logic Programs*, Journal of Logic Programming, 2(4), 1985, pp. 295-312.

[9] D. Gabbay, E. Laenens, D. Vermeir, *Credulous vs. Sceptical Semantics for Ordered Logic Programs*, J. Allen, R. Fikes, E. Sandewall (eds.), Proc. of the 2nd Intern. Conference on Knowledge Representation and Reasoning (KR'91), Morgan Kaufmann, 1991, pp. 208-217.

[10] P. Geerts, D. Vermeir, *Credulous and autoepistemic reasoning using ordered logic*, A. Nerode, W. Marek, V.S. Subrahmanian (eds.), Proc. of the 1st Intern. Workshop on Logic Programming and Non-monotonic Reasoning, 1991, pp.21-36.

[11] M. Gelfond, V. Lifschitz, *The Stable Model Semantics for Logic Programming*, R. Kowalski, K. Bowen (eds.), Proc. of the 5th Symposium on Logic Programming, MIT Press, 1988, pp. 1070-1080.

[12] M. Gelfond, V. Lifschitz, *Classical Negation in Logic Programs and Disjunctive Databases*, New Generation Computing, 9, OHMSHA, 1991, pp. 365-385.

[13] M. Goldszmidt, J. Pearl, *Rank-based systems: A simple approach to belief revision, belief update and reasoning about evidence and actions*, Proc. of the 3rd Intern. Conference on Knowledge Representation and Reasoning (KR'92), 1992, pp. 661-672.

[14] A. Hunter, *A Conceptualization of Preferences in Non-monotonic Proof Theory*, D. Pearce and G. Wagner (eds). Logics in AI, LNCS 633, 1992, pp. 174- 188.

[15] M.H. Jamil, L.V.S. Lakshmanan, *Realizing Orlog in LDL*, I.S. Mumick (ed.), Proc. of the Workshop on Combining Declarative and Object-Oriented Databases (in cooperation with ACM SIGMOD), May 1993, pp.45-59.

[16] M. Kifer, G. Lausen, J. Wu, *Logical Foundations of Object-Oriented and rame-Based Languages*, Technical Report 90/14 (2-nd revision), Dept. of Computer Science, SUNY at Stony Brook, August 1990. (To appear in the Journal of the ACM).

[17] E. Laenens, D. Vermeir, *A Fixpoint Semantics for Ordered Logic*, Journal of Logic and Computation, 1(2), Oxford University Press, 1990, pp. 159-185.

[18] E. Laenens, D. Vermeir, *Assumption-free Semantics for Ordered Logic Programs: On the Relationship Between Well-founded and Stable Partial Models*, Journal of Logic and Computation, 2(2), Oxford University Press, 1992, pp. 133-172.

[19] N. Leone, P. Rullo, *Stable Model Semantics and its Computation for Ordered Logic Programs*, B. Neumann (ed.), Proc. of the 10th European Conference on Artificial Intelligence (ECAI'92), 1992, pp. 92-96.

[20] F.G. McCabe, *Logic and Objects*, Prentice Hall, 1992.

[21] L.M. Pereira, J.J. Alferes, J.N. Aparicio, *Contradiction Removal within Well Founded Semantics*, A. Nerode, W. Marek, V.S. Subrahmanian (eds.), Proc. of the 1st Intern. Workshop on Logic programming and Non-monotonic Reasoning, 1991, pp.106-119.

[22] L.M. Pereira, J.N. Aparicio, J.J. Alferes, *Nonmonotonic Reasoning with Well Founded Semantics*, Proc. of the Intern. Conference on Logic Programming (ICLP',91), 1991, pp.475-504.

[23] L.M. Pereira, J.J. Alferes, J.N. Aparicio, *Contradiction Removal with Explicit Negation*, Proc. of the Applied Logic Conference, ILLC, 1992.

[24] L.M. Pereira, J.J. Alferes, *Well Founded Semantics for Logic Programs with Explicit Negation*, B.Neumann (ed.), Proc. of the 10th European Conference on Artificial Intelligence, John Wiley & Sons, 1992, pp. 102-106.

[25] L.M. Pereira, J.N. Aparicio, J.J. Alferes, *Nonmonotonic Reasoning with Logic Programming*, Journal of Logic Programming, Elsevier Science Publishing, 17, 1993, pp.227-263.

[26] T. Przymusinski, *Every Logic program has a Natural Stratification and an Iterated Fixed Point Model*, Proc. of the 8th Symposium on Principles of Database Systems, ACM SIGACT-SIGMOD, 1989, pp. 11-21.

[27] T. Przymusinski, *Well-Founded Semantics Coincides with the Three-Valued Stable Semantics*, Fundamenta Informaticae XIII, IOS Press, 1990, pp. 445-463.

[28] N. Roos, *A Logic for Reasoning with Inconsistent Knowledge*, Artificial Intelligence, Elsevier Science Publishers, 57, 1992, pp.69-103.

[29] M. Ryan, *Defaults and Revision in Structured Theories*, Proc. of the 6th IEEE Symposium on Logic in Computer Science (LICS), 1991, pp. 362-373.

[30] M. Ryan, *Representing Defaults as Sentences with Reduced Priority*, Proc. of the 3rd International Conference on Knowledge Representation and Reasoning (KR'92), 1992, pp. 649-660.

[31] J. S. Schlipf, *A Survey of Complexity and Undecidability Results in Logic Programming*, H. Blair, W. Marek, A. Nerode, J. Remmel (eds.), Proc. of the Workshop on Structural Complexity and Recursion-theoretic methods in Logic Programming, Nov. 1992, pp. 143-165.

[32] L. A. Stein, *Skeptical Inheritance: Computing the Intersection of Credulous Extensions*, Proc. of the 11th Intern. Joint Conference on Artificial Intelligence, 1989.

[33] M. van Emden, R. Kowalski, *The Semantics of Predicate Logic as a Programming Language*, Journal of the ACM, 23(4), 1976, pp. 733-742.

[34] A. van Gelder, K.A. Ross, J.S. Schlipf, *The Well-Founded Semantics for General Logic Programs*, Journal of the Association for Computing Machinery, 38(3), July 1991, pp.620-650.

[35] C. Witteveen, *Skeptical Reason Maintenance is Tractable*, Proc. of the 2nd Intern. Conference on Knowledge Representation and Reasoning (KR91), 1991, pp.570-581.

[36] C. Witteveen, *Expanding Logic Programs*, D. Pearce and G. Wagner (eds). Logics in AI, LNCS 633, 1992, pp. 372-390.

[37] G. Wagner, *Reasoning with Inconsistency in Extended Deductive Databases*, L.M. Pereira and A. Nerode (eds.), Proc. of the 2nd Intern. Workshop on Logic programming and Non-monotonic Reasoning, 1993, pp.300-315.