A Frame-based Characterisation of the Paraconsistent Well-founded Semantics with Explicit Negation

João Alcântara, Carlos Viegas Damásio and Luís Moniz Pereira

Centro de Inteligência Artificial (CENTRIA)
Departamento de Informática, Universidade Nova de Lisboa
2829-516 Caparica, Portugal. E-mail: `jfla|cd|lmp@di.fct.unl.pt`

Abstract. In this work, we present *Paraconsistent Well-Founded Semantics with eXplicit negation* ($WFSX_p$) by introducing a frame-based semantics. As far as we know, this is the first time that a declarative approach for $WFSX_p$ is effectively characterised in terms of a model theory. Dismissing usual techniques based on immediate consequence operators, our definition depicts $WFSX_p$ by minimising models satisfying particular criteria. Aware that $WFSX_p$ captures a large deal of logic programming semantics (including the main well-founded ones), our approach subsumes indirectly a semantics for them. Then assuming an illustrative tone, by adding some conditions, we also show how it can be adjusted to define *Well-Founded Semantics with eXplicit negation* ($WFSX$), *Well-Founded Semantics* (WFS) as well as *Answer Sets*. Given its declarative aspects, our work paves the way to define a logical representation of the family of well-founded semantics.

1 Introduction

During much time, unsuccessful attempts to elaborate a model theoretical definition for *Well-Founded Semantics with eXplicit negation* ($WFSX$) [11, 2] and its paraconsistent variant ($WFSX_p$) [1, 4] have challenged researchers. One of the main difficulties is that in some cases, due to Coherence Principle[1], non-standard conditions are introduced to consider an interpretation as satisfiable. This issue is illustrated with the following program:

Example 1. Let P be the logic program containing the following rules:

$$P = \left\{ b \rightarrow a \qquad not\ b \rightarrow b \qquad \neg a \right\}$$

As it is known, coherence is one of the distinguishing characteristics of $WFSX_p$. According to this semantics, "b" is assigned truth-value *undefined* in P, whilst given that "$\neg a$" is assigned *true*, we can conclude applying coherence "*not* a" is also true, i.e. "a" is *false*. The first rule of P is satisfied by this interpretation,

[1] According to this principle, explicit negation entails default negation.

even with false head and undefined body. However, if we withdraw the rule "$\neg a$" from P, the very interpretation assigning false to head and undefined to body will not satisfy the first rule of P! The intuition is that rule "$\neg a$" overrides the undefinedness of "a" obtained on the basis of the first rule, justifying to accept this assignment as satisfiable. In other words, the same assignment of values to head and body of a rule when coherence does not intervene is considered unsatisfiable. The problem is how a model theory can characterise declaratively such a quite unusual behaviour of *WFSX* and WFSX_p.

Since Lifschitz's work [9], stable models semantics has been defined on a fully model theoretical ground. Crowning it all, D. Pearce in [10] established that stable models/answer sets are particular minimal models under Heyting's monotonic (superintuitionistic) logic of *here-and-there*. As for Well-Founded Semantics (*WFS*) [19], it is usually described operationally by an alternate fixpoint operator. However in [7], Dung redefined declaratively *WFS* as an admissibility semantics. More recently, in [3] Cabalar captured *WFS* by employing a two-dimensional extension of the logic here-and-there.

Pursuing a declarative characterisation of WFSX_p, we were motivated by G. Restall's work on Substructural Logics [14]. As defined by him, substructural logics are non-classical logics notable for the absence of structural rules present in classical logic. In addition, techniques from substructural logics are useful in the study of traditional logics such as classical and intuitionistic logic. In particular substructural logics can be elegantly interpreted via frame-based semantics.

In this paper our aim is to present (declaratively) *Paraconsistent Well-Founded Semantics with eXplicit negation* (WFSX_p) in terms of a model theory. Indeed, even extending Pearce's original motivation, we exhibit our approach by sticking to frame-based semantics as presented by Greg Restall. This shall be done by employing a three-dimensional version of the constructions presented by Pearce in [10] to capture Answer Sets. The two additional dimensions are introduced to deal with explicit negation, and incomplete information. No immediate consequences operator is required, and the definition of WFSX_p is reduced to a simple minimisation process among models satisfying certain conditions. As WFSX_p embeds many logic programming semantics, including *WFS* and *WFSX*, our approach is also suited to define them. Consequently, as well as for answer sets, we can guarantee that the family of well-founded semantics may also be characterised logically.

This paper is structured as follows: in Section 2 we present WFSX_p resorting to Przymusinski's operator [12]. Section 3 is the core of our work where we show our main contributions: the definition of WFSX_p as a frame-based semantics and how we can easily adjust it to grasp *WFSX*, *WFS* and Answer Sets. Finally, we draw conclusions, and mention future works.

2 A fixpoint definition of WFSX_p

In the first semantics for extended logic programs, explicitly negated atoms are simply interpreted as new atoms. As result explicit negation and default negation are unrelated. This is the case of Przymusinski's Extended Stable Models [13] and Sakama's Extended Well-founded Semantics [15]. By embodying the coherence principle (if $\neg L$ holds then *not* L should too), *Well-Founded Semantics with eXplicit negation WFSX* [11, 2], and its paraconsistent version ($WFSX_p$) [1, 4] fill this missing link. It is also know that $WFSX_p$ embeds a great deal of well-known logic programming semantics [5]. In this section, at once we recall the definition of extended logic programs and then we show how to interpret them via $WFSX_p$ by resorting to Przymusinski's operator alike [12].

Definition 1 (Extended Logic Programs). *An extended rule is a formula*[2]

$$
\begin{aligned}
&b_1 \wedge \ldots \wedge b_k \wedge \neg c_1 \wedge \ldots \wedge \neg c_l \wedge \\
&\qquad not\ d_1 \wedge \ldots \wedge not\ d_m \wedge not\ \neg e_1 \ldots \wedge not\ \neg e_n \rightarrow L
\end{aligned} \tag{1}
$$

in which L is a literal (it is either an atom a or its explicit negation $\neg a$), $b_r, 0 \leq r \leq k$, $c_s, 0 \leq s \leq l$, $d_t, 0 \leq t \leq m$, and $e_u, 0 \leq u \leq n$ are atoms. When $k = 0$, $l = 0$, $m = 0$ and $n = 0$, the extended rule is denoted by fact *and may be seen as $\mathbf{t} \rightarrow L$, in which $\mathbf{t}$ is a syntactical representation for the truth value "true".*

An extended logic program *is defined as a set of extended rules.*

In an extended logic program context, throughout this paper, we reserve the letters *a, b, c, d* and *e* (eventually subscripted) to refer to atoms, whilst L refers to a literal. A literal is an atom "a" (positive literal) or its explicit negation "$\neg a$" (negative literal), and $L = \neg\neg L$ holds. As usual we semantically identify an extended logic program with its *ground program*, which is the possibly infinite set of all ground rules from the program. An *objective logic program* is an extended logic program P such that for each rule (1) of P, $m = n = 0$ (there is no *not* L in P, in which L is a literal). In extended logic programs, as negative literals have the same status as positive literals, we consider the *Extended Herbrand base* of a program P contains the set of all ground literals, denoted by Lit, found in P. Additionally if $L \in Lit$, then $\neg L \in Lit$. Assuming a set of literals Lit, *formulae* are defined as usual: if ϕ and ψ are formulae, and $L \in Lit$, then the truth constant $\mathbf{t}$, L, $\neg\phi$, *not* ϕ, $\phi \wedge \psi$, $\phi \vee \psi$, and $\psi \rightarrow \phi$ are also formulae. Semantically, literals are interpreted by extending the usual notion of interpretation:

Definition 2 (Extended interpretation). *An extended interpretation S is a subset of Lit, in which a literal L is true if and only $L \in S$; otherwise it is false. The set of all extended interpretations of formulae with respect to an extended logic program P is denoted by $\mathcal{S}_P$.*

The immediate consequences operator is given by extending van Emden and Kowalski [18] to handle negative literals is crucial to the operational definition of $WFSX_p$:

[2] In order to keep uniformity with the usual notation of implication symbol in Logic, logic programming rules are represented by $\psi \rightarrow \phi$ instead of $\phi \leftarrow \psi$.

Definition 3 (Immediate Consequences Operator). *Let P be an objective logic program. Define the immediate consequences operator $T_P : \mathcal{S}_\mathcal{P} \rightarrow \mathcal{S}_\mathcal{P}$, mapping extended interpretations to extended interpretations:*

$$T_P(S) = \{L \mid \text{ if there is a rule } L_1, \ldots, L_m \rightarrow L \in P \text{ where } L_i \in S, 1 \leq i \leq m\} .$$

An additional definition is required to capture default negation:

Definition 4 (Partial Interpretations). *A partial interpretation is a pair of extended interpretations $< I^h, I^t >$*[3]*. By convention when referring to partial interpretations, we adopt the capital letters I, J or K (eventually subscripted) as a short form for respectively $< I^h, I^t >$, $< J^h, J^t >$ or $< K^h, K^t >$. The set of all partial interpretations is $\mathcal{I}^p$.*

When referring to partial or extended interpretations we usually omit the set of literals Lit, which is implicitly provided. By I^h we mean to represent what certainly holds in the partial interpretation, whilst I^t contains what may hold (i.e. the "complement" of what certainly does not hold). The annotations h and t then capture, respectively, what is "true" and what is "true or undefined" ("non-false"). It is also important to mark that $I^h \subseteq I^t$ is not imposed and, consequently, paraconsistency is allowed, i.e. something may certainly hold and not hold (by being simultaneously "true" and not "non-false"). Two orders among partial interpretations are useful:

Truth ordering: $I_1 \subseteq_T I_2$ iff $I_1^h \subseteq I_2^h$ and $I_1^t \subseteq I_2^t$.
Knowledge ordering: $I_1 \subseteq_K I_2$ iff $I_1^h \subseteq I_2^h$ and $I_2^t \subseteq I_1^t$.

in which I_1 and I_2 are partial interpretations. A minimal model with respect to $\subseteq_T$ (resp. $\subseteq_K$) is said to be T-minimal (resp. K-minimal)

The next definition is based on the usual way of interpreting default negation in well-founded and stable model semantics for logic programs. It is a Gelfond-Lifschitz like division operator [8] which transforms extended logic programs into objective ones, for which we know how to compute the corresponding least models through the immediate consequences operator.

Definition 5 (Program Division). *Consider an extended logic program P and an extended interpretation S. The division of program P by S, denoted by $\frac{P}{S}$, is the objective logic program obtained from P by removing all rules containing a default literal not L such that $L \in S$, and by then removing all the remaining default literals from P.*

The program division $\frac{P}{S}$ evaluates according to S all occurrences of default literals in P. Then the least model of the resulting program can be determined according to the following definition:

Definition 6. *Consider an extended logic program P and a partial interpretation $< I^h, I^t >$. Define*

$$C_P(< I^h, I^t >) = T_{\frac{P}{I^t}}(I^h)$$

[3] In order to unify notation we do not adopt that one used in [6], in which I^h and I^t are respectively denoted as I^t and I^{tu}.

Coherence principle, which ensures that explicit negation entails default negation ($\neg L \Rightarrow not\ L$), is imposed by Alferes and Pereira [11, 2] by resorting to the semi-normal version of an extended logic program:

Definition 7 (Semi-normal program). *Let P be an extended logic program. The semi-normal version of P, denoted P_s, is the extended logic program obtained by replacing every extended rule "$Body \rightarrow L$" belonging to P by "$Body, not\ \neg L \rightarrow L$", in which $\neg L$ is the complement of L with respect to explicit negation.*

Operator C_P maps partial interpretations to extended interpretations. We now define a new operator mapping partial interpretations to partial interpretations in the same spirit of Przymusinski's [12]:

Definition 8 (Partial Consequences Operator). *Let P be an extended logic program, and I and J be two partial interpretations. The partial consequences operator is given by the equation:*

$$\Theta_P^J(I) = \left\langle C_P(< I^h, J^t >), C_{P_s}(< I^t, J^h >)\right\rangle .$$

Given that Θ_P^J is a monotonic operator with respect to Truth ordering [6], its fixpoint is guaranteed to exist (Knaster-Tarski fixpoint theorem [17]) and is important to our objectives. We also say that a pair of partial interpretations $[K, J]$ is a *pre-model* of P whenever $\Theta_P^J(K) \subseteq_T K$, i.e. both $C_P(< K^h, J^t >) \subseteq K^h$ and $C_{P_s}(< K^t, J^h >) \subseteq K^t$.

Definition 9. *Let P be an extended logic program and J be a partial interpretation. Define $\Omega_P(J) = I$ such that I is a T-minimal partial interpretation satisfying $\Theta_P^J(I) \subseteq_T I$.*

Consequently we can say that pre-models fully characterise operator $\Omega_P(J)$. Given that $\Omega_P(J)$ is monotonic with respect to $\subseteq_K$ [6], we conclude immediately again by Knaster-Tarski fixpoint theorem that $\Omega_P(J)$ has a least fixpoint under the knowledge ordering of partial interpretations: the paraconsistent well-founded model.

Definition 10 (Paraconsistent Well-founded Semantics). *Consider an extended logic program P. The partial stable models of P are the fixpoints of the operator $\Omega_P(J)$, i.e. J is a partial interpretation of P such that $\Omega_P(J) = J$. The paraconsistent well-founded model of P, $WFM_p(P)$, is the K-minimal partial stable model of P.*

As motivated in [4], given $WFM_p(P) = < J^h, J^t >$ a literal L holds in $WFM_p(P)$ iff $L \in J^h$, whilst *not* L holds in $WFM_p(P)$ iff $L \notin J^t$.

Example 2. Let P be an extended logic program:

$$P = \left\{\ not\ b \rightarrow c \qquad a \qquad not\ d \rightarrow d \qquad a \rightarrow b \qquad \neg a\ \right\}$$

whose $WFM_p(P) = < J^h, J^t >$ in which $J^h = \{a, \neg a, b, c\}$ and $J^t = \{d\}$.

Due to coherence principle, one of the distinguishing features of $WFSX_p$ is that it does not enforce default consistency, i.e. "L" and "*not* L" may be *simultaneously* true. In Example 2 this is the case of "a", "$\neg a$", "b", and "c". As usual in well-founded semantics, a literal may also be undefined, i.e. both "L" and "*not* L" may be false. This is the case of "d" in the example above.

Considering the importance of $WFSX_p$ to the understanding of the effects of paraconsistent reasoning in logic programming, in the next section we show how to capture this semantics entirely in terms of logical concepts.

3 A frame-based semantics for $WFSX_p$

In various complex forms of reasoning, it is allowable to draw intuitive conclusions unattained in the realm of Classical Logic. Blamed by some logicians as responsible for this weakness, the plain two-valued semantics of Classical Logic has been enriched in distinct ways in order to couple with the expected results. Nonetheless, resorting to multi-valued semantics is not the only solution; indeed one can even preserve the two-valued settlement and make robust the semantics instead by offering more places at which sentences may be evaluated.

That is the core approach of semantics for Modal Logic, in which propositions are evaluated at many different possible worlds via accessibility relations. Here we shall exhibit a similar approach to introduce an entirely logical definition for $WFSX_p$. However, as stressed by Greg Restall [14], in our context it is preferable to denote these semantics by *frames* rather than *possible worlds semantics*.

In this section, by resorting to frames, we provide a complete declarative version for $WFSX_p$. We commence its presentation by showing the three requisite components of a frame: point sets, accessibility relations, and truth sets.

Definition 11 (Point Set). *[14] A point set $\mathcal{P} = \langle Q, \sqsubseteq \rangle$ is a set Q together with a partial order $\sqsubseteq$ on Q. The set $Prop(\mathcal{P})$ of* propositions *on $\mathcal{P}$ is the set of all subsets X of Q* closed upwards*, that is, if $x \in X$ and $x \sqsubseteq x'$ then $x' \in X$.*

As usual in the Substructural Logics literature, we shall employ accessibility relations to evaluate intensional connectives:

Definition 12 (Accessibility Relations). *[14]*

- *A binary relation C is a* plump negative two-place accessibility relation *on the point set $\mathcal{P}$ if and only if for any $x, y, x', y' \in \mathcal{P}$, in which xCy, $x' \sqsubseteq x$ and $y' \sqsubseteq y$ it follows that $x'Cy'$.*
- *A ternary relation R is a* plump three-place accessibility relation *on the point set $\mathcal{P}$ if and only if for any $x, y, z, x', y', z' \in \mathcal{P}$, in which $Rxyz$, $x' \sqsubseteq x$, $y' \sqsubseteq y$ and $z \sqsubseteq z'$ then $Rx'y'z'$.*

Plump negative two-place accessibility relations shall be associated with negations, whilst plump three-place accessibility relations with conditionals. Below we define truth sets to collect subsets of $Prop(\mathcal{P})$ whose interpretation makes them eligible to define the truth constant $\mathbf{t}$.

Definition 13 (Truth Sets). *[14] If R is a (plump) three-place accessibility relation on a point set $\mathcal{P}$ then for any subset $T \in Prop(\mathcal{P})$*

- *T is a* left truth set *for R if and only if for each $x, y \in \mathcal{P}$, $x \sqsubseteq y$ if and only if for some $z \in T$, $Rzxy$.*
- *T is a* right truth set *for R if and only if for each $x, y \in \mathcal{P}$, $x \sqsubseteq y$ if and only if for some $z \in T$, $Rxzy$.*

Now that we have defined a point set, accessibility relations and truth sets, the notion of frame can be introduced straightforwardly:

Definition 14 (Frame). *[14] A frame is a point set $\mathcal{P}$ together with any number of accessibility relations and truth sets on $\mathcal{P}$.*

We reserve $\mathcal{F}$ to denote a frame we shall use in the definition of $WFSX_p$:

1. The point set $\mathcal{P} = \langle Q, \sqsubseteq \rangle$, which is represented graphically in Fig 1, such that $Q = \{hhp, hhn, htp, htn, thp, thn, ttp, ttn\}$, and the partial order $\sqsubseteq$ is indicated through the sense pointed by the arrows.

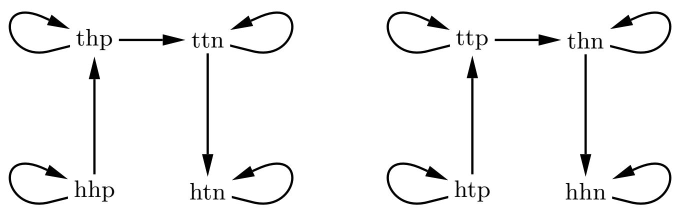

Fig. 1. Point Set for $WFSX_p$

2. The accessibility relations defined on $\mathcal{P}$: $R, R_{\neg}, R_{not}$ and $R_{\sim}$, being that R is exhibited in Table 1, and $R_{\neg}, R_{not}$ and $R_{\sim}$ are respectively shown in Figs 2, 3 and 4, in which the existence of an arrow from point x to y denotes that there is an accessibility relation from x to y. As the reader can check, R is a plump three-place accessibility relation, whilst $R_{\neg}$ and R_{not} are plump negative two-place accessibility relations. Furthermore, $R_{\sim}$ was introduced to capture the semi-normal transformation presented in the previous section.

R hhp hhp hhp	*R hhp hhp thp*	*R hhp hhp ttn*	*R hhp hhp htn*	*R hhp thp thp*	*R hhp thp ttn*
R hhp thp htn	*R htp htp htp*	*R htp htp ttp*	*R htp htp thn*	*R htp htp hhn*	*R htp ttp ttp*
R htp ttp thn	*R htp ttp hhn*	*R thp hhp thp*	*R thp hhp ttn*	*R thp hhp htn*	*R thp thp thp*
R thp thp ttn	*R thp thp htn*	*R ttp htp ttp*	*R ttp htp thn*	*R ttp htp hhn*	*R ttp ttp ttp*
R ttp ttp thn	*R ttp ttp hhn*	*R ttn hhp ttn*	*R ttn hhp htn*	*R ttn thp ttn*	*R ttn thp htn*
R thn htp thn	*R thn htp hhn*	*R thn ttp thn*	*R thn ttp hhn*	*R htn hhp htn*	*R hhn htp hhn*

Table 1. Accessibility Relation R

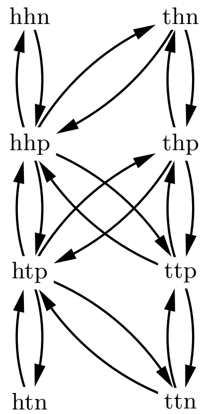

Fig. 2. Accessibility relation $R_\neg$

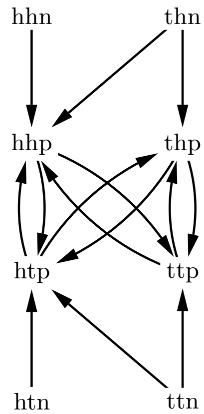

Fig. 3. Accessibility relation R_{not}

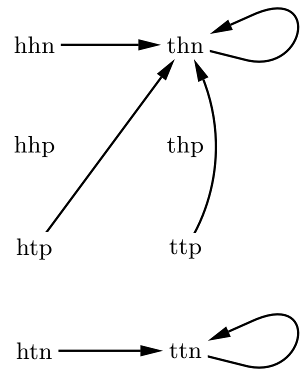

Fig. 4. Accessibility relation for $R_\sim$

3. The unique (right) truth set of R (see Definition 13) is

$$\{hhp, hhn, htp, htn, thp, thn, ttp, ttn\}$$

The frame $\mathcal{F}$ was motivated by the works presented by Pearce [10], Cabalar [3], and Restall [14], and the final goal is to guarantee that the partial interpretations K and J employed in the previous section to present pre-models are respectively captured by truth-values induced by $\{hhp, hhn, htp, htn\}$ and $\{thp, thn, ttp, ttn\}$ (see Definition 16).

In addition, in the point set $\mathcal{P} = \langle Q, \sqsubseteq \rangle$ each element $xyz \in Q$ is resultant of the three-dimensional character of our frame, being part of a complex logical engineering: the dimension x was borrowed from Pearce's work and is related to the minimisation process; the dimension y was borrowed from Cabalar's work and was introduced to deal with partial interpretations; finally, we have introduced the dimension z to handle explicit negation appropriately. Alternatively, explicit negation could be treated in terms of a partial Kripke-style semantics as Pearce did. In this case a two-dimensional frame would be enough, whilst the semantics would be four-valued at each point in order to deal with paraconsistency. However, influenced by Greg Restall's directions, we prefer to preserve the two-valued assessment applied to a three-dimensional frame. Anyway, both approaches could be interchangeably employed to capture $WFSX_p$.

The partial order $\sqsubseteq$ in $\mathcal{P}$ is motivated by the original definition of the logic of here-and-there and by the intuition behind the Coherence principle.

The accessibility relations R, $R_\neg$ and R_{not} are obtained by adapting N-valuations used in [10] to characterise respectively the operators $\rightarrow$, $\neg$ and *not* into our frame. Furthermore, extra relations should be introduced due to the requirements exhibited in Definition 12. The final result allows us to extend the above cited N-valuations, and to make them suitable to be used to capture $WFSX_p$[4].

[4] In [10] N-valuations are used to determine Answer Sets.

As for $R_\sim$, its objective is to simulate through the unary operator $\sim$ the behaviour of the semi-normal transformation (see Definition 19).

The main idea is using the frame $\mathcal{F}$ above to characterise $WFSX_p$ accordingly. In order to do that, we disclose in the sequel some definitions to handle the semantical part.

Definition 15 (Belief Sets). *By belief set, we mean a 4-tuple* $(S^{hp}, S^{hn}, S^{tp}, S^{tn})$*, in which* S^{hp}, S^{hn}, S^{tp}*, and* S^{tn} *are sets of atoms. In other words, an atom* A *is true in* S^x *if and only if* $A \in S^x$*; otherwise,* A *is false in* S^x*, with* $x \in \{hp, hn, tp, tn\}$*.*

Belief sets are a key concept in our work. In fact, $WFSX_p$ shall be defined by minimising pair of belief sets satisfying some conditions. Firstly we should notice that as $WFSX_p$ is worked out in terms of partial interpretations, in order to put on a complete declarative frame-based version for $WFSX_p$, a translation between partial interpretations and belief sets shall be used to support us to obtain our intended results:

Definition 16 (Translation). *Let* $B = (S^{hp}, S^{hn}, S^{tp}, S^{tn})$ *be a belief set and* I *a partial interpretation. A translation between them is defined as follows:*

$$\begin{array}{ll} A \in I^h \text{ iff } A \in S^{hp} & A \in I^t \text{ iff } A \in S^{tp} \\ \neg A \in I^h \text{ iff } A \notin S^{hn} & \neg A \in I^t \text{ iff } A \notin S^{tn} \end{array}$$

in which A *is an atom in* Lit*.*

We say a pair of belief sets is a *pre-model* of a program P iff their corresponding translation is also a pre-model of P. Furthermore, as side effect of Definition 16, the truth and knowledge ordering presented in Section 2 can also be mimicked in a relationship involving belief sets:

Definition 17 (Truth and knowledge ordering between belief sets). *Let* $B_1 = (S_1^{hp}, S_1^{hn}, S_1^{tp}, S_1^{tn})$ *and* $B_2 = (S_2^{hp}, S_2^{hn}, S_2^{tp}, S_2^{tn})$ *be two belief sets. The truth and knowledge orderings among belief sets are defined by*

Truth: $B_1 \subseteq_T B_2$ *iff* $S_1^{hp} \subseteq S_2^{hp}$, $S_1^{tp} \subseteq S_2^{tp}$, $S_2^{hn} \subseteq S_1^{hn}$ *and* $S_2^{tn} \subseteq S_1^{tn}$.
Knowledge: $B_1 \subseteq_K B_2$ *iff* $S_1^{hp} \subseteq S_2^{hp}$, $S_2^{tp} \subseteq S_1^{tp}$, $S_2^{hn} \subseteq S_1^{hn}$ *and* $S_1^{tn} \subseteq S_2^{tn}$.

As usual $B_1 \subset_T B_2$ (resp. $B_1 \subset_K B_2$) iff $B_1 \subseteq_T B_2$ (resp. $B_1 \subseteq_K B_2$) and $B_1 \neq B_2$. The mechanism behind truth and knowledge ordering between belief sets is crucial to guarantee the expected characterisation of $WFSX_p$. Pursuing this aim, firstly we use $\subseteq_T$ to define HT^3-Interpretations:

Definition 18 (HT^3-Interpretation). *A* HT^3-*interpretation is defined as the pair* $[B^h, B^t]$*, in which* B^h *and* B^t *are belief sets satisfying* $B^h \subseteq_T B^t$*. If* $B = B^h = B^t$*, then* $[B, B]$ *is* total.

In order to expunge any misunderstanding, we shall reserve the letters B and S for respectively denote belief sets and sets of atoms, using the notation

- $B^h = (S^{hhp}, S^{hhn}, S^{htp}, S^{htn})$
- $B^t = (S^{thp}, S^{thn}, S^{ttp}, S^{ttn})$

We are going to associate each S^x in $[B^h, B^t]$ to a x in the point set of $\mathcal{F}$. That is made clear in the next definition:

Definition 19 **(HT^3-Model).** *Let $M = [B^h, B^t]$ be a HT^3-interpretation, w be a three-dimensional point of a frame $\mathcal{F}$, A be an atom, and both ϕ and ψ be formulae. We say ϕ is satisfied by M in w, written $(M, w) \models \phi$, iff*

1. $(M, w) \models A$ *iff* $A \in S^w$
2. $(M, w) \models \mathbf{t}$ *for all* w *in* $\mathcal{F}$
3. $(M, w) \models \phi \wedge \psi$ *iff* $(M, w) \models \phi$ *and* $(M, w) \models \psi$
4. $(M, w) \models \phi \vee \psi$ *iff* $(M, w) \models \phi$ *or* $(M, w) \models \psi$
5. $(M, w) \models \neg\phi$ *iff for each* w' *in* $\mathcal{F}$ *s.t.* $wR_{\neg}w'$, $(M, w') \not\models \phi$
6. $(M, w) \models$ *not* ϕ *iff for each* w' *in* $\mathcal{F}$ *s.t.* $wR_{not}\, w'$, $(M, w') \not\models \phi$
7. $(M, w) \models \sim \phi$ *iff it exists* w' *in* $\mathcal{F}$ *s.t.* $wR_{\sim}w'$, $(M, w') \not\models \phi$
8. $(M, w) \models \psi \rightarrow \phi$ *iff for each* w', w'' *in* $\mathcal{F}$ *s.t.* $R\ w\ w'\ w''$, *if* $(M, w') \models \psi$, *then* $(M, w'') \models \phi$

In particular, M is a HT^3-model of a theory T iff for each ϕ in T, $(M, w) \models \phi$ for all w in $\mathcal{F}$.

As it shall be clear soon, the operator $\sim$ introduced in item 7 of Definition 19 captures the behaviour of the semi-normal transformation (Definition 7). In order to obtain the expected results, we interpret each rule $\phi \rightarrow L$ of an extended logic program P as the formula $\phi \rightarrow (L \vee \sim L)$. Let us refer to these programs as $P_{\sim}$. Thus the unusual behaviour addressed in Example 1 can be treated by rewriting in $P_{\sim}$ the extended logic programs rules of P. In the remaining of this section we are going to show which criteria we apply to minimise the HT^3-model of $P_{\sim}$ to obtain the corresponding extended logic program P.

An interesting point is that the satisfaction relation relies totally on the usual two truth values, i.e. given M and w, any formula ϕ is either satisfied or not. In the sequel we interrelate a HT^3-model of a program $P_{\sim}$ to a pre-model of P, as per the proposition below.

Proposition 1. *Let $M = [B^h, B^t]$ be a pair of belief sets. If M is a HT^3-model of a program $P_{\sim}$, then M is a pre-model of an extended logic program P.*

Next we define the set $SUBTOTAL(P)$, which shall be used to show that any pre-model $M \notin SUBTOTAL(P)$ is worthless for computing the partial stable models of P.

Definition 20. *Let P be an extended logic program, and both B^h and B^t be belief sets. We define $SUBTOTAL(P) = \{[B^h, B^t] \mid B^h \subseteq_T B^t$, where both $[B^h, B^t]$ and $[B^t, B^t]$ are pre-models of $P\}$.*

The following theorem assures us that $SUBTOTAL(P)$ corresponds to the set of HT^3-models of $P_\sim$.

Theorem 1. *The pair of belief sets $[B^h, B^t]$ is a HT^3-model of a program $P_\sim$ iff $[B^h, B^t] \in SUBTOTAL(P)$.*

Adducing Theorem 1 we conclude that only HT^3-models in $SUBTOTAL(P)$ should be taken into account to obtain $WFSX_p(P)$. Before proving the main result of this paper, we introduce below an ordering relation between pair of belief sets:

$$[B^h, B^t] \subseteq_h [C^h, C^t] \text{ iff } B^t = C^t \text{ and } B^h \subseteq_T C^h.$$

So in $\subseteq_h$, whilst fixing the belief set B^t, we minimise B^h according to $\subseteq_T$. A minimal pair of belief sets with respect to $\subseteq_h$ is said to be *h-minimal*. The next theorem show us how to define partial stable models of a program P via h-minimal HT^3-models of $P_\sim$.

Theorem 2. *Total h-minimal HT^3-models of $P_\sim$ correspond to partial stable models of P.*

As a corollary from this theorem, $WFSX_p(P)$ can be obtained by finding among partial stable models of P the K-minimal one. Hence just by minimising the HT^3-models of $P_\sim$ as shown in this section, we can determine $WFSX_p(P)$. We recall Example 2 to illustrate our results:

$$P = \left\{ \, not\ b \rightarrow c \qquad a \qquad not\ d \rightarrow d \qquad a \rightarrow b \qquad \neg a \, \right\}$$

Based on Definition 20, in $SUBTOTAL(P)$ there are 650 elements (HT^3-models of $P_\sim$); 75 being h-minimal; out of which only the two ones shown below are total:

$$M_1 = [(\{a,b,c\},\{b,c,d\},\{d\},\{a,b,c,d\}),(\{a,b,c\},\{b,c,d\},\{d\},\{a,b,c,d\})]$$

and

$$M_2 = [(\{a,b,c,d\},\{b,c,d\},\{\ \},\{a,b,c,d\}),(\{a,b,c,d\},\{b,c,d\},\{\ \},\{a,b,c,d\})].$$

Notice that pairs of belief sets, belief sets, and sets of atoms are respectively enclosed by using [ ], (), and {} such that M_1 and M_2 above obey the following organisation: $[(S^{hhp}, S^{hhn}, S^{htp}, S^{htn}),(S^{thp}, S^{thn}, S^{ttp}, S^{ttn})]$. Based on our results, the belief sets $B_1 = (\{a,b,c\},\{b,c,d\},\{d\},\{a,b,c,d\})$ and $B_2 = (\{a,b,c,d\},\{b,c,d\},\{\ \},\{a,b,c,d\})$ are the paraconsistent stable models of P. As K-minimal partial stable model, B_1 is $WFM_p(P)$. Translating accordingly B_1 to a partial interpretation I, we conclude that "*a*", "$\neg a$", "*b*", and "*c*" as well as "*not a*", "*not* $\neg a$", "*not b*", and "*not c*" are true in I. On the other hand, neither "*d*" nor "*not d*" are true in I (confer Example 2).

We can easily capture semantics as *WFSX* and *WFS*. Apropos, *WFSX* can be seen as the explosive version of $WFSX_p$, i.e. in order to define it, we should impose default consistency by neglecting the models belonging to $SUBTOTAL(P)$

in which both an atom and its default negation are present. Alternatively, starting from $\mathcal{F}$, we can tailor a new frame to define *WFSX* by guaranteeing that $hhp \sqsubseteq htp$, $thp \sqsubseteq ttp$, $ttn \sqsubseteq thn$, $htn \sqsubseteq hhn$. Furthermore, the frame used by Pearce in [10] to characterise Answer Sets can also be captured by collapsing *hhp* into *htp*, *thp* into *ttp*, *ttn* into *thn*, and *htn* into *hhn*. Thus for both *WFSX* and Answer Sets, contradictory interpretations are avoided. In order to couple with the requirements expressed in Definition 12, we should also add new instances to $R, R_{\neg}$ and R_{not} . Subsequent definitions hold *ipsi litteris* to define these semantics. Following a similar reasoning, *WFS* can be seen as a *WFSX* version free of explicit negation. For a program P without explicit negation this can be achieved by considering just those elements in $SUBTOTAL(P)$ such that, for each atom A, $A \in S^{hhn}$, $A \in S^{htn}$, $A \in S^{thn}$, and $A \in S^{ttn}$ hold. This process is also enough to obtain Cabalar's proposal [3] to define *WFS*.

4 Conclusion

In this work, we have defined a fully declarative approach for $WFSX_p$ grounded on a three-dimensional frame-based semantics. Although this is an elegant technique tailored to be employed within substructural logics, it can be naturally adjusted to deal with logic programs. In contrast to the usual operational approach based on program division, our proposal characterises $WFSX_p$ by resorting to a plain minimisation process among models satisfying particular conditions. Considering that $WFSX_p$ embeds many logic programming semantics, including *WFSX* and *WFS*, we are also presenting a declarative approach to them. We have shown too how one can easily embed *WFSX*, *WFS* and Answer Sets via frames.

Following other motivations, R. Schweimeier and M. Schroeder have declaratively captured $WFSX_p$ in a recent work [16] by resorting to an argumentation approach. However, as far as we know, this is the first time that a declarative proposal for $WFSX_p$ is effectively characterised in terms of a model theory.

Because of some problems resulting from coherence principle applied to disjunctions, we have deliberately postponed to a next paper to approach theories with them in the rule heads. This task accomplished, we acquire the capability of not only to capture $WFSX_p$ for extended logic programs, but also for any theory with full nesting of formulae for all programs connectives.

We can see this work as a first step in the exploration of interesting points related to a declarative/logical representation of $WFSX_p$. In particular it may provide useful information to compare one semantics with another and relate it to other formalisms. Furthermore, there are several interesting open issues connected with well-founded semantics as the exploration of links between explicit negation and other kinds of "strong negation", the presentation of a suitable treatment for disjunction, and analyses of paraconsistency in $WFSX_p$ on the ground of paraconsistent logics. By employing our frame-based structure, we expect to address these issues in future work.

References

1. J. J. Alferes, C. V. Damásio, and L. M. Pereira. A logic programming system for non-monotonic reasoning. *J. of Automated Reasoning*, 14(1):93–147, 1995.
2. J. J. Alferes and L. M. Pereira. *Reasoning with Logic Programming*. LNAI 1111. Springer-Verlag, 1996.
3. P. Cabalar. Well founded semantics as two-dimensional here-and-there. In *Answer Set Programming*, pages 15–20. AAAI Press, 2001. (AAAI Press, technical report SS-01-01), inside the 2001 AAAI Spring Symposium Series.
4. C. V. Damásio. *Paraconsistent Extended Logic Programming with Constraints*. PhD thesis, Universidade Nova de Lisboa, 1996.
5. C. V. Damásio and L. M. Pereira. A survey of paraconsistent semantics for logic programs. In D. Gabbay and P. Smets, editors, *Hand. of Defeasible Reasoning and Uncertainty Management Systems*, volume 2, pages 241–320. Kluwer Academic Publishers, 1998.
6. C. V. Damásio and L. M. Pereira. Antitonic logic programs. In T. Eiter and M. Truszczynski, editors, *Procs. of the 6th Int. Conf. on Logic Programming and Nonmonotonic Reasoning*, pages 748–759. Springer, LNCS/LNAI, 2001.
7. P. M. Dung. Negation as Hypotheses: An Abductive Foundation for Logic Programming. In Koichi Furukawa, editor, *Proceedings of the 8th Int. Conf. on Logic Programming, Paris*. MIT, June 1991.
8. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R. Kowalski and K. A. Bowen, editors, *5th International Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.
9. V. Lifschitz. Minimal belief and negation as failure. *Artificial Intelligence*, (57):53–72, 1994.
10. D. Pearce. From here to there: stable negation in logic programming. In D. M. Gabbay and H. Wansing, editors, *What is Negation?*, pages 161–181. Kluwer Academic Publishers, 1999.
11. L. M. Pereira and J. J. Alferes. Well-founded semantics for logic programs with explicit negation. In B. Neumann, editor, *European Conference on Artificial Intelligence*, pages 102–106, Wien, Austria, 1992. John Wiley & Sons.
12. H. Przymusinska and T. C. Przymusinski. Semantic issues in deductive databases and logic programs. In R. Banerji, editor, *Formal Techniques in Artificial Intelligence, a Sourcebook*, pages 321–367. North Holland, 1990.
13. T. Przymusinski. Extended stable semantics for normal and disjunctive programs. In Warren and Szeredi, editors, *ICLP'90*, pages 459–477. MIT Press, 1990.
14. Greg Restall. *An Introduction to Substructural Logics*. Routledge, 2002.
15. Chiaki Sakama. Extended well-founded semantics for paraconsistent. In Institute for New Generation Computer Technology (ICOT), editor, *Proc. of the Int. Conf. on Fifth Generation Computer Systems*, volume 2, pages 592–599. IOS Press, 1992.
16. Ralf Schweimeier and Michael Schroeder. A parameterised hierarchy of argumentation semantics for extended logic programming and its application to the well-founded semantics. *Theory and Practice of Logic Programming*, 2003. To appear.
17. A. Tarski. Lattice-theoretic fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.
18. M. van Emden and R. Kowalski. The semantics of predicate logic as a programming language. *Journal of ACM*, 4(23):733–742, 1976.
19. A. van Gelder, K. Ross, and J. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.