Back and Forth Semantics for Normal, Disjunctive and Extended Logic Programs

David Pearce
DFKI, PF 15 11 50, D-66041 Saarbrücken
e-mail: pearce@dfki.de

Abstract

We define a logical semantics called *back-and-forth*, applicable to normal and disjunctive datalog programs as well as to programs possessing a second, explicit or 'strong' negation operator. We show that on normal programs it is equivalent to the well-founded semantics (WFS), and that on disjunctive programs it is equivalent to the P-stable semantics of Eiter, Leone and Saccà, hence to Przymusinski's 3-valued stable semantics. The main advantage is that it is characterised by simple conditions on models in a well-known nonclassical logic and therefore provides a better insight into the nature of partial stable models from a logical standpoint. It also suggests why the P-stable models are a natural generalisation of WFS to the disjunctive case.

On extended programs with strong negation, the back-and-forth semantics is apparently new, differing from answer sets, from WSFX and from the static semantics.

Keywords: stable models, P-stable models, disjunctive programs, intermediate logics, strong negation.

1 Introduction

Of the major semantics proposed for logic programs with negation-as-failure, the well-founded semantics (WFS) of [9] and the stable model semantics of [10] have proved to have appealing and enduring features. Each has its advantages and drawbacks. Inference based on stable model semantics is stronger and in many instances more 'intuitive' than that based on well-founded semantics. On the other hand, WFS is defined for a larger class of programs and admits a more efficient computation. Then again, the stable model semantics for normal programs has a very simple and natural extension to disjunctive programs, and even to extended programs having an additional, strong negation operator. So it is almost immediately applicable to these more expressive kinds of formalism. It is much harder to identify analogously simple and natural extensions of WFS. Indeed, in the case of disjunctive programs, several different semantics have been proposed as extensions of WFS.

Another difference between the two approaches concerns their logical characterisation and their relation to other systems of nonmonotonic reasoning. There is a large body of results comparing stable model inference with default logic, circumscription, nonmonotonic modal logic, and other systems. Recently, in [22], inference base on stable models and answer sets has been characterised as a simple form of minimal model reasoning in a nonclassical logic. In the case of WFS, fewer interesting connections with nonmonotonic logics are known, and comparable characterisations in nonclassical logic have yet to be provided.

The present paper is devoted to one such characterisation. More precisely, we describe here a logical semantics for logic programs which we call the *back-and-forth* semantics. Like stable semantics, it has an obvious extension to disjunctive programs and to programs possessing a strong negation operator. The underlying logic involved is a well-known nonclassical logic, sometimes called *here-and-there*, because it is complete for linear Kripke-frames having precisely two 'worlds', "here" and"there". Our semantics is so named because its intended models can be described by a certain condition that takes one from "here" to "there" and back again. The logic of here-and-there can also be described as a 3-valued logic and as the greatest extension of intuitionistic logic that is properly contained in classical logic. It also has a natural interpretation already proposed by Heyting [16] in 1930: namely as a logic whose "third" truth value corresponds to the property that the proposition in question can never be false, yet cannot be proven true. To handle strong negation, we simply take the least strong negation extension of here-and-there (in the sense of Nelson's [20] strong negation) and re-apply the same back-and-forth conditions. Here-and-there, and its least strong negation extension, are precisely the logics suitable for characterising inference based on stable models and answer sets: the latter are in fact just back-and-forth models satisfying one extra condition.

It turns out that on normal programs back-and-forth models are equivalent to stationary models, so the resulting semantics is equivalent to WFS. On disjunctive programs our semantics coincides with the P-stable, or partial stable semantics, defined by Eiter, Leone and Saccà [6], and shown by them to be equivalent in turn to the 3-valued stable models of Przymusinski [27]. On extended programs with strong negation, back and forth semantics appears to be new. It is intermediate in strength between answer sets semantics and the extension of WFS with 'explicit' negation, WFSX, developed in [23]. It is also different from the extended semantics proposed in [26] and recently in [28].

In summary, the present paper aims to help clarify the logical nature of WFS and stationary semantics as a form of preferential model reasoning in a nonclassical logic. We also identify"natural" extensions of WFS to disjunctive and extended logic programs and study the relation of these to other semantics proposed in the literature.

2 Preliminaries

This section reviews the main definitions and results needed for the remainder of the paper.

2.1 Logic Programs

First let us recall the syntax of logic programs. We use standard logical notation, rather than the special notation often employed in logic programming. In the setting of disjunctive logic programs, program formulas are built-up from atomic formulas using the logical constants: $\wedge, \vee, \rightarrow, \neg$, standing respectively for conjunction, disjunction, implication, and negation. Negation is often denoted by the symbol '*not*' and referred to as negation-as-failure or negation-by-default. The nonlogical vocabulary comprises a fixed set of predicate symbols (no function symbols) and a nonempty set of names. This language will be left implicit throughout.

Program formulas with free variables are treated as shorthand for the set of their ground instances, so that a logic program can be represented as a collection Π of closed program formulas having the following form:

$$A_1 \wedge \ldots \wedge A_m \wedge \neg\, A_{m+1} \wedge \ldots \wedge \neg\, A_n \rightarrow B_1 \vee \ldots \vee B_k \tag{1}$$

where the A_i and B_j are ground (closed) atoms. These formulas are often called *rules* and written in a different notation, actually back-to-front; but the form is always the same. There is a single arrow, preceded by conjunctions of atoms and negated atoms, and followed by a disjunction of atoms. We may have m or n zero, or $m = n$; but we always have $k \geq 1$. When $k = 1$ in each formula, ie. no disjunctions occur in Π, the program is said to be *normal*.

We denote the set of all ground atoms (resp. ground literals) in the language of Π by $\mathcal{H}_\Pi$ (resp. $\mathcal{L}_\Pi$). Whenever Π is given we drop the subscript 'Π'. For any literal L, let $\neg.L$ denote the complement of L, ie. for an atom A, $\neg.A$ is $\neg A$, and for a negated literal, $L = \neg B$, $\neg.L = B$. If X is a set of literals, set $\neg.X = \{\neg.L : L \in X\}$. Given a program formula φ of form (1), we denote by $a(\varphi)$ the set of literals occurring in the antecedent ('body') of φ and by $c(\varphi)$ the set of atoms occurring in the consequent ('head') of φ.

Given a program Π and a set S of ground atoms in the language of Π we consider what it means for certain formulas to be satisfied by S. In particular, we say that S *satisfies* an atom A if $A \in S$. Moreover a conjunction of atoms is satisfied if each conjunct is satisfied, and a disjunction of atoms is satisfied if at least one of the disjuncts is satisfied. If α is a conjunction of atoms and β is a disjunction of atoms, then S satisfies $\alpha \rightarrow \beta$ if S satisfies β whenever it satisfies α. Given a program Π and a set $S \subseteq \mathcal{H}$, we define as usual the Gelfond-Lifschitz reduction of Π as follows. Given a formula $\varphi \in \Pi$ of the form (1), let φ^S be the formula

$$A_1 \wedge \ldots \wedge A_m \rightarrow B_1 \vee \ldots \vee B_k \tag{2}$$

if for each $i = m+1, \ldots, n$, $A_i \notin S$. Otherwise, if for some $i = m+1, \ldots, n$, $A_i \in S$, let φ^S be the empty formula. Set $\Pi^S = \{\varphi^S : \varphi \in \Pi\}$.

2.2 Intuitionistic Logic

Classical logic is based on the (classical) notion of *truth*, whereby every proposition is either true or false, independent of our knowledge of its truth-value. In constructive reasoning this principle (of *tertium non datur*) is rejected, and logic is built upon the notion of (constructive) *proof*. The standard formalisation of the logic of constructive reasoning is due to Heyting [16], and called *intuitionistic* logic. We denote it by **H**. In intuitionistic logic, terms and formulas are built-up in the usual manner, using the logical constants of **H**: $\wedge, \vee, \neg, \rightarrow$ and the quantifiers $\exists, \forall$; but one explains the meaning of the connectives and quantifiers not in terms of classical truth-conditions but in terms of constructions acting on proofs. There are many variations on this explanation, which, when made sufficiently precise, lead to logical calculi equivalent to Heyting's (see any standard text, eg. [3]). Intuitionistic logic can readily be presented in any of the usual 'deductive' styles, eg. as a tableau system, a natural deduction system or as a Gentzen-style sequent calculus. For example, the sequent calculus for **H** restricts sequents on the right to single formulas; the natural deduction rules follow those of classical logic, except that the classical rule *reductio ad absurdem* is omitted. **H** is a proper subsystem of classical logic: every intuitionistically valid formula is also classically valid, but not conversely. The derivability relation for **H** will be denoted by $\vdash_H$.

There are several types of semantics for intuitionistic logic. We mention here only the method of Kripke models, also known as *possible worlds semantics.* Formally, one starts with a so-called Kripke *frame* $\mathcal{F}$,where

$$\mathcal{F} = \langle W, \leq \rangle,$$

W is a set and $\leq$ is a partial-ordering on W. The elements of W are sometimes called possible worlds. One may also think of them intuitively as *stages* (in the growth of knowledge). At each world or stage $w \in W$ some primitive propositions (atoms) are verified as *true*, and, once verified at some stage w, an atom A remains true at every 'later' stage, ie. at all w' such that $w \leq w'$ (in this sense knowledge may grow in several directions, but verified propositions are never subsequently forgotten). A Kripke *model* $\mathcal{M}$ can therefore be represented as a frame $\mathcal{F}$ together with an assignment i of sets of atoms to each element of W, such that if $w \leq w'$ then $i(w) \subseteq i(w')$. An assignment is then extended inductively to all formulas via the following rules:

$\varphi \wedge \psi \in i(w)$ iff $\varphi \in i(w)$ and $\psi \in i(w)$
$\varphi \vee \psi \in i(w)$ iff $\varphi \in i(w)$ or $\psi \in i(w)$
$\varphi \rightarrow \psi \in i(w)$ iff for all w' such that $w \leq w'$ $\varphi \in i(w')$ implies $\psi \in i(w')$
$\neg\varphi \in i(w)$ iff for all w' such that$w \leq w'$ $\varphi \notin i(w')$

(we omit here the semantics of quantification).

A formula φ is true in a Kripke model $\mathcal{M}$ at world w, in symbols $\mathcal{M}, w \models \varphi$, iff $\varphi \in i(w)$. φ is true in a Kripke model $\mathcal{M}$, in symbols $\mathcal{M} \models \varphi$, if it is true at all worlds in $\mathcal{M}$. A formula φ is said to be *valid*, in symbols, $\models_H \varphi$, if it is true in all Kripke models. φ is said to be an **H**-consequence of a set Π of formulas, written $\Pi \models_H \varphi$, iff for all models $\mathcal{M}$ and any world $w \in \mathcal{M}$, $\mathcal{M}, w \models \Pi \Rightarrow \mathcal{M}, w \models \varphi$. The Kripke semantics is *complete* for **H** in the sense that for all T and φ

$$T \vdash_H \varphi \quad \text{iff} \quad T \models_H \varphi. \tag{3}$$

We use $Cn_H(T)$ to denote the set of all **H**-consequences of a theory T, and we denote by $Th(\mathcal{M})$ the set of all sentences true in a Kripke model $\mathcal{M}$.

2.3 Intermediate Logics

We also consider *intermediate* logics, obtained by adding additional axioms to **H**; they are complete wrt a generalised notion of Kripke frame. An intermediate logic is called *proper* if it is strictly contained in classical logic. In the lattice of intermediate propositional logics (extensively investigated in the literature, see eg. [2]) classical logic has a unique lower cover which is the supremum of all proper intermediate logics. This greatest proper intermediate logic we shall denote by **J**. It is often referred to as the logic of "here-and-there", since it is characterised by linear Kripke frames having precisely two elements or worlds:'here' and 'there'. **J** is also characterised by the three element Heyting algebra, and is known by a variety of other names, including the Smetanich logic. Truth tables for **J** were already given by Heyting [16], and the logic was further used by Gödel in a paper of 1932, [13]. However, it was apparently first axiomatised by Łukasiewicz [18]. He characterised **J** by adding to **H** the axiom schema

$$(\neg\alpha \to \beta) \to (((\beta \to \alpha) \to \beta) \to \beta).$$

He also showed that disjunction is definable in **J**. An algebraic characterisation of **J** is straightforward; for present purposes, however, it is more practical to use the Kripke-model characterisation.

2.4 Minimal Models

Let **J** be the Smetanich logic of here-and-there. Then **J** is determined by Kripke models based on the 2-element, 'here-and-there' frame. Each **J**-model can be represented as a structure $\langle\{h,t\},\leq,i\rangle$, where the worlds h and t are reflexive, and $h \leq t$. For any worlds h,t we use the corresponding upper-case letters H,T to denote the set of all atoms true at those worlds; eg. $H = \{A : A \in i(h)\}$. Note that for any model $\langle\{h,t\},\leq,i\rangle$ we always have $H \subseteq T$. We may equivalently represent a model simply as a pair $\langle H,T\rangle$. Until further notice, by "model" we mean a here-and-there model of this kind.

Our semantics will be based on a notion of *minimal* model defined as follows.

Definition 1 *Define a partial ordering $\preceq$ on* **J***-models by $\langle H,T\rangle \preceq \langle H',T'\rangle$ iff $H \subseteq H'$ and $T = T'$. Then a model $\langle H,T\rangle$ of Π is said to be a minimal model of Π iff it is minimal among models of Π under the $\preceq$-ordering.*

The following lemmas will be useful.

Lemma 1 ([22]) *Let Π be a logic program and let $\mathcal{M} = \langle H,T\rangle$ be a model of Π.Then $\mathcal{M} \models \Pi^T$.*

Lemma 2 ([22]) *Let Π be a logic program and $\mathcal{M} = \langle H,T\rangle$ be a model of Π. $\mathcal{M}$ is a minimal model of Π iff H is a minimal set of atoms satisfying Π^T.*

3 Back-and-forth semantics and WFS

We now introduce a semantics for logic programs, based on minimal models in **J**, and show that on normal programs it is equivalent to the well-founded semantics.

Definition 2 *Let Π be a logic program and $\mathcal{M} = \langle H,T\rangle$ a model of Π. $\mathcal{M}$ is said to be a* back-and-forth *model of Π iff (i) $\mathcal{M}$ is a minimal model of Π, and (ii) T is a minimal set of atoms satisfying Π^H.*

By Lemma 2, condition (i) is equivalent to the property that H is a minimal set satisfying Π^T, giving rise to the label "back-and-forth". The *back-and-forth* semantics for logic programs is determined by the collection of all back-and-forth models of a program. We show that on normal programs it is equivalent to the well-founded semantics.

First, recall that a set $H \subset \mathcal{H}_\Pi$ is said to be a stable model of a program Π if H is a minimal Herbrand model of Π^H ([10, 11]). In the case of a normal program Π, a minimal set of atoms satisfying Π^H is actually unique (ie. least) and is often denoted by $\Gamma_\Pi(H)$ (we usually drop the subscript Π). Thus for normal Π, H is a stable model of Π iff

$$H = \Gamma(H).$$

Note that, by uniqueness, in the case of a normal program a back-and-forth model of Π can equivalently be described as a model $\langle H, T\rangle$ of Π such that $\Gamma(H) = T$ and $\Gamma(T) = H$.

Viewed thus as an operator on sets of atoms, Γ is antimonotone, from which it follows that $\Gamma^2 (= \Gamma \bullet \Gamma)$ is monotone and has a least fixpoint, say I_Π. Then $\Gamma(I_\Pi)$ is the greatest fixpoint of Γ^2.

Definition 3 ([8]) *For any normal program* Π *the* well-founded model *of* Π*, is characterised by* $WFM(\Pi) = \langle I_\Pi, \Gamma(I_\Pi)\rangle$.

The *well-founded semantics* (WFS) of a program Π is determined by the well-founded model, as follows: an atom A is *true* if $A \in I_\Pi$, *false* if $A \notin \Gamma(I_\Pi)$, and *undecided* otherwise.[1] More generally,

Definition 4 ([25]) *For a normal program* Π*, and set* J *of atoms, if* $\Gamma^2(J) = J \subseteq \Gamma(J)$*, then the pair* $\langle J, \Gamma(J)\rangle$ *is said to be a* stationary *model of* Π.

So the well-founded model is simply the least stationary model.

Proposition 1 *For normal programs and atomic queries the back-and-forth semantics is equivalent to the well-founded semantics.*

Proof. It is straightforward to show that back-and-forth models determine stationary models, and *vice versa.* One can then verify that the well-founded semantics of a program agrees with the back-and-forth semantics (we restrict attention to atomic queries for which WFS is usually defined).

The stable models of a normal program Π can be regarded as stationary models in which there are no undefined atoms. Likewise, stable models can be viewed (even in the disjunctive case) as back-and-forth models with no undefined atoms. In fact, minimality and definiteness suffice.

Definition 5 ([22]) *Let* Π *be a logic program. A minimal model* $\langle H, T\rangle$ *of* Π *such that* $H = T$ *is said to be an* equilibrium *model of* Π.

Proposition 2 ([22]) *A set* S *of atoms is a stable model of a program* Π *iff it is the set of atoms true in an equilibrium model of* Π.

It follows that inference based on stable models (for short *stable* inference) as well as inference based on stationary models (for short *well-founded* inference) can be regarded as extending in each case the logic **J**. Moreover these extensions are well-behaved in the sense that two programs that are equivalent (ie. interderivable) in **J** must have the same stable models and the same stationary models, since, by completenss, the must have in each case the same here-and-there models. A monotonic logic that underlies in this well-behaved sense a given nonmonotonic inference relation is called a *deductive basis* for the relation in question, [5]. It is easy to see that classical logic does not in this sense form a deductive basis for either well-founded or stable inference, even though the latter is a strengthening of classical inference. However, since we know **J** to be the greatest proper intermediate logic, we have established

Corollary 1 *The logic* **J** *of here-and-there is a maximal deductive basis of back-and-forth inference, hence (for atomic queries) for well-founded inference, as well as for stable inference.*

[1]The well-founded semantics is often equivalently characterised by defining $WFM(\Pi)$ as the pair $\langle I, \overline{\Gamma(I)}\rangle$, where $\overline{X}$ denotes the complement of X (in the language of Π, or in some suitable, universal language). In this case $\overline{\Gamma(I)}$ represents the set of false atoms.

4 Disjunctive Logic Programs

There is no general agreement on how to extend WFS or stationary semantics to disjunctive programs; different proposals can be found eg. in [27], [28], [4], [6, 7]. The back-and-forth semantics turns out to be equivalent to the partial stable (or P-stable) semantics recently extended to disjunctive programs by Eiter, Leone and Saccà [6, 7]; this is in turn equivalent to Przymusinski's [27] 3-valued stable semantics.

The P-stable semantics is defined in terms of what are called *unfounded* sets. The authors use a slightly different notion of model or *interpretation.* For them an interpretation $\mathcal{M} = \mathcal{M}^+ \cup \mathcal{M}^-$ is a consistent set of ground literals, where $\mathcal{M}^+$ is a set of atoms and $\mathcal{M}^-$ is a set of atoms prefixed by $\neg$. To make comparison with here-and-there models easier, in what follows we shall represent interpretations as pairs of sets of atoms $\mathcal{M} = \langle H, T \rangle$, where H represents the true atoms ($\mathcal{M}^+$) and T represents the set of atoms that are not false. As in the case of here-and-there models, the complement $\overline{T} = \mathcal{H} - T$ of T denotes the set of false atoms, and therefore $\overline{T}$ corresponds to the set $\neg.\mathcal{M}^-$. By the consistency requirement, we always have $H \subseteq T$. Clearly the two notions of interpretation are equivalent and fully intertranslatable. Using our reformulated version, the concepts of *unfounded set* and *founded* interpretation from [7] can be described as follows.

Definition 6 ([7]) *Let $\mathcal{M} = \langle H, T \rangle$ be an interpretation for a disjunctive logic program Π, (ie. $H \subseteq T \subset \mathcal{H}_\Pi$). A set X of ground atoms is said to be an* unfounded *set for Π wrt $\mathcal{M}$ iff for each $A \in X$ and each formula φ in Π such that A belongs to the consequent $c(\varphi)$ of φ, either (i) $a(\varphi) \cap (\neg.H \cup \overline{T} \cup X) \neq \emptyset$ or (ii) $c(\varphi) \not\subseteq (\neg.H \cup \overline{T} \cup X)$.*

Definition 7 ([7]) *An interpretation $\langle H, T \rangle$ of a program Π is said to be* founded *if H is a minimal set of atoms satisfying Π^T.*

Definition 8 ([7]) *Let $\mathcal{M} = \langle H, T \rangle$ be an interpretation of a logic program Π. $\mathcal{M}$ is said to be a* P-stable *or* partial stable *model of Π iff (a) $\mathcal{M}$ is founded; (b) $\overline{T}$ is a maximal unfounded set for Π wrt $\mathcal{M}$.*

We now show that P-stable models and back-and-forth models are equivalent. We do this in three stages.

Lemma 3 *Let $\mathcal{M} = \langle H, T \rangle$ be a minimal model of Π, such that T satisfies Π^H. Then $\mathcal{M}$ is founded and $\overline{T}$ is an unfounded set for Π wrt $\mathcal{M}$.*

Proof. Assume the hypothesis of the lemma. Then clearly $\mathcal{M}$ is a (consistent) interpretation of Π and that this model is minimal is equivalent by Lemma 2 to the property that H is a minimal set satisfying Π^T, so $\mathcal{M}$ is certainly founded. It remains to verify that $\overline{T}$ is unfounded for Π wrt $\mathcal{M}$. This will be the case, if, for every $A \notin T$, and each $\varphi \in \Pi$ such that $A \in c(\varphi)$, either condition (i) or condition (ii) of Definition 7 holds. Setting $X = \overline{T}$, these conditions simplify to

$$(i)' \quad a(\varphi) \cap (\neg.H \cup \overline{T}) \neq \emptyset$$

$$(ii)' \quad c(\varphi) \not\subseteq (\neg.H \cup \overline{T}).$$

Since the head or consequent of φ can only contain (positive) atoms, *(ii)'* further simplifies to

$$(ii)'' \quad c(\varphi) \cap T \neq \emptyset.$$

Now, since T satisfies Π^H, in particular, T satisfies each φ^H of the form $\alpha \rightarrow \beta$. If β contains an atom $A \notin T$, then, if β is true wrt T, some head atom B, different from A, must belong to T, implying that *(ii)"* holds. If T does not satisfy β, then it must also fail to satisfy α. Then for some atom C in α, $C \notin T$, implying that $C \in \overline{T}$. Alternatively, it may happen that φ^H is empty, which is the case if the antecedent of φ contains a literal $\neg A$ such that $A \in H$. These last two possibilities imply that $a(\varphi) \cap (\neg.H \cup \overline{T}) \neq \emptyset)$, as required. □

Lemma 4 *Let* Π *be a disjunctive program and* $\mathcal{M} = \langle H, T \rangle$ *an interpretation for* Π *such that* $\mathcal{M}$ *is founded and* $\overline{T}$ *is an unfounded set for* Π *wrt* $\mathcal{M}$*.Then* $\mathcal{M}$ *is a minimal here-and-there model of* Π*, and* T *satisfies* Π^H*.*

Proof. Assume the hypothesis and regard $\mathcal{M}$ equivalently as a here-and-there model $\langle \{h, t\}, \leq, i \rangle$. We first show that $\mathcal{M} \models \Pi$. Since $\overline{T}$ is unfounded, conditions (i) and (ii) of Definition 7 hold, and simplify as before to $a(\varphi) \cap (\neg.H \cup \overline{T}) \neq \emptyset)$, and $c(\varphi) \cap T \neq \emptyset$. In other words, for every $A \notin T$ and every $\varphi \in \Pi$ such that $A \in c(\varphi)$, either (a) the body of φ contains a negative literal $\neg B$, such that $B \in H$, or (b) the body of φ contains an atom C such that $C \notin T$, or (c) the head of φ contains an atom B belonging to T. Let φ be of the form $\alpha \rightarrow \beta$. If (a) holds, $\mathcal{M}, h \models B$, hence $\mathcal{M}, h \models \neg B \rightarrow \beta$ and $\mathcal{M}, h \models \alpha \rightarrow \beta$, and so $\mathcal{M} \models \varphi$. If (b) holds, then $\mathcal{M}, h \models \neg C$. So again $\alpha \rightarrow \beta$ is true at h in $\mathcal{M}$, hence $\mathcal{M} \models \varphi$. Finally, suppose that neither (a) nor (b) holds. By (c), β, hence $\alpha \rightarrow \beta$ is true in $\mathcal{M}$ at t. It remains to verify that $\alpha \rightarrow \beta$ is also true at h.

First, since $\mathcal{M}$ is founded, H satisfies Π^T. Therefore for all nonempty φ^T, H satisfies φ^T hence also φ. Therefore, since $\mathcal{M}, t \models \varphi$, also $\mathcal{M}, h \models \varphi$. Suppose that φ^T is empty. Then α contains a literal $\neg B$ such that $B \in T$. Consequently, $\mathcal{M} \models \neg\neg B$. So $\mathcal{M}, h \models \alpha \rightarrow \beta$, as required.

Lastly, we need to consider those $\varphi \in \Pi$ such that conditions (i) and (ii) of Definition 7 do not apply. In this case all the atoms in the consequent of φ belong to T, so trivially $\mathcal{M}, t \models \varphi$. Also $\mathcal{M}, h \models \varphi$ by the earlier argument, namely that H satisfies Π^T and so H satisfies φ for all nonempty φ^T. If, alternatively, φ^T is empty, then the antecedent of φ contains a literal $\neg B$, such that $B \in T$. Consequently, $\mathcal{M} \models \neg\neg B$ and so $\mathcal{M} \models \varphi$.

Since $\mathcal{M}$ is a here-and-there model of Π and is founded, clearly $\mathcal{M}$ is minimal. It remains to check that T satisfies Π^H. By our earlier Lemma, T satisfies Π^T, so it remains to verify that T satisfies the formulas in $\Pi^H - \Pi^T$. These are formulas φ^H such that the antecedent of φ contains some literal $\neg B$, such that $B \in T$ but $B \notin H$. Consider any such φ^H of the form $\alpha \rightarrow \beta$. If all the atoms in β belong to T then clearly T satisfies β hence also $\alpha \rightarrow \beta$. If not, then conditons (a) or (b) or (c) above hold of φ. Of these (a) is excluded by the fact that φ^H is nonempty. If (b) holds, then α contains an atom C not in T, so T satisfies $\alpha \rightarrow \beta$. If (c) holds, then β contains an atom belonging to T, whence T satisfies β and therefore $\alpha \rightarrow \beta$. Consequently, T satisfies Π^H, completing the proof of the lemma.□

Proposition 3 *Let* Π *be a disjunctive logic program and let* $\mathcal{M} = \langle H, T \rangle$ *be an interpretation of* Π*.* $\mathcal{M}$ *is a back-and-forth model of* Π *if and only if it is a P-stable model of* Π*.*

Proof. Immediate from the lemmas. One need only add the observation that maximising the unfounded set $\overline{T}$ is equivalent to minimising the set T such that T satisfies Π^H. □

It follows from the results of [7] that back-and-forth semantics, being equivalent to P-stable, is also equivalent to the 3-valued stable semantics of Przymusinski [27]. It is worth noting, however, that Przymusinski's 3-valued semantics is not defined on the basis of the logic **J** of here-and-there. In particular, his truth-value assignments are such that the third truth-value can be understood as 'undefined' or 'indeterminate'; hence if an atom takes this value, so does its negation. By contrast, in **J** the third truth-value, as Heyting noted, corresponds to "cannot be false, yet not provably true", yielding the consequence that if an atom A takes this value, then $\neg A$ takes the value 'false'. In addition, Przymusinski's intended models are defined using a 3-valued generalisation of the Gelfond-Lifschitz reduction of a program, rather than, as here, a back-and-forth construction involving the ordinary Gelfond-Lifschitz reduction.

Other subclasses of P-stable models, studied in [7], such as M-stable and L-stable models, can readily be defined and investigated in the present framework. For instance, a back-and-forth model $\mathcal{M}$ of Π is an M-stable (maximal stable) model of Π if and only if there is no back-and-forth model $\mathcal{M}'$ of Π such that $Th(\mathcal{M}) \cap \mathcal{L}$ is a proper subset of $Th(\mathcal{M}') \cap \mathcal{L}$. Similarly, an L-stable (or least undefined stable) model corresponds to a back-and-forth model $\langle H, T \rangle$ in which the set of atoms $T - H$ is minimal among the back-and-forth models of the program in question.

5 Extended Logic Programs

Several authors have proposed extensions of the syntax of logic programs to include a second negation operator, representing explicit, direct falsity, as opposed to negation by default [21, 11, 26, 23]. We shall denote this new negation by '$\sim$'. The formulas of such extended logic programs thus have the form

$$L_1 \wedge \ldots \wedge L_m \wedge \neg\, L_{m+1} \wedge \ldots \wedge \neg\, L_n \rightarrow K_1 \vee \ldots \vee K_k \tag{4}$$

where now the L and K with subscripts range over what we might call the *strict* literals, ie. over atoms and atoms prefixed by the new negation $\sim$. Let *Lit* denote the collection of all strict literals.

In logic, a natural way to represent explicit falsity is to use the notion of *strong* negation, introduced by Nelson [20]. Nelson's logic **N** is known as *constructive logic with strong negation* and was developed as an alternative approach to constructive reasoning. It adds to Heyting's logic the insight that primitive propositions may not only be constructively *verified* but also constructively *falsified*. The language of intuitionistic logic is accordingly extended by adding a new, strong negation symbol, '$\sim$', and giving it the intepretation that $\sim A$ is true if A is constructively false. This is quite different from the meaning of $\neg A$ in **H**, where $\neg A$ is the same as $A \rightarrow \bot$. The meaning of strong negation, '$\sim$', in Nelson's logic can be explained by extending the usual interpretation of the intuitionistic connectives and quantifiers to include a notion of *disproof* (or *refutation*). An axiom system for **N** is obtained by taking the axiom schemata and rules of **H** together with the following axiom schemata (due to Vorob'ev [29]) involving strong negation (where '$\alpha \leftrightarrow \beta$' abbreviates $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$:

N1. $\sim (\alpha \rightarrow \beta) \leftrightarrow \alpha \wedge \sim \beta$
N2. $\sim (\alpha \wedge \beta) \leftrightarrow \sim \alpha \vee \sim \beta$
N3. $\sim (\alpha \vee \beta) \leftrightarrow \sim \alpha \wedge \sim \beta$
N4. $\alpha \leftrightarrow \sim\sim \alpha$
N5. $\sim \neg\alpha \leftrightarrow \alpha$
N6. (for atomic α) $\quad \sim \alpha \rightarrow \neg\alpha$.

N is a conservative extension of **H** in the sense that any formula without strong negation is a theorem of **N** if and only if it is a theorem of **H**. Notice that Nelson's negation '$\sim$' is aptly termed 'strong', since in **N**, $\sim\varphi \rightarrow \neg\varphi$ is a theorem, for all φ (not only atomic φ). (See eg. [15, 3]). The derivability relation for **N** is denoted by $\vdash_N$.

A semantics for **N** can be obtained by a straightforward generalisation of the Kripke semantics for **H** discussed earlier. One may take the same Kripke-frames as for intuitionistic logic; what changes is the nature of the assignments or truth-valuations. In the case of **H**, atoms were simply verified (directly) at any stage or world, and falsity was a derived notion. However, for constructive logic with strong negation a direct notion of falsification is available as a primitive concept. Accordingly, we can now imagine that at every stage or world some atoms are verified and some are falsified. For reasons of space we do not repeat the definitions here, but see eg. [15] or [22].

Given any intermediate logic **I** one may form the least strong negation extension of **I** by adding $\sim$ and the Vorob'ev axioms N1-N6. This is always a conservative extension [17]. If **I** is complete for a given class $\mathcal{K}$ of Kripke-frames then its least strong extension of is complete for the same class $\mathcal{K}$ of frames under the generalised assignments. Consequently, the logic **J** of here-and-there has a least strong negation extension, which we shall denote by **N2**, that is complete for the class of 2-element, here-and-there frames. An **N2** model can also be represented as a pair $\langle H, T\rangle$, where now H and T are sets of strict literals. An algebraic characterisation of $N2$ is straightforward; see [17] where $N2$ is presented as a 5-valued logic.

The logic **N2** may be used to capture inference based on the answer set semantics [12] for extended logic programs. One defines minimal models and equilibrium models as above, but with respect to **N2** rather than **J**. Analogous to Proposition 2 one can then show that equilibrium models correspond to answer sets, see [22]. Similarly, we obtain a simple generalisation of back-and-forth semantics by reformulating Definitions 1 and 2 for the logic **N2**. The details are straightforward and will not be repeated here. We merely note that a back-and-forth model $\langle H, T\rangle$ is now a pair of sets of strict literals.

Back-and-forth semantics therefore generalises WFS to the case of extended logic programs, with or without disjunction. How does it compare with other generalisations of WFS? First, notice that a naive adaptation of WFS to extended programs formed by replacing sets of atoms with sets of literals (see [26]) does not lead to an adequate semantics, as [24] points out. The problem arising there is addressed by Pereira *et al* [23, 24] who propose an alternative extension of the well-founded semantics, called WFSX, for non-disjunctive programs. WSFX is a conservative extension of WFS in the sense that it agrees with WFS on all normal programs. It also shares with WFS the property that, whenever the semantics is defined for some program, a single, intended model of the program is defined and may be characterised by an iterative process. The inference relation defined by WSFX is weaker than stable inference, not only on the class of normal programs. In fact it can be shown that whenever an extended logic program has an answer set, the set of literals defined by WFSX to be true is a subset (not necessarily a proper subset) of those literals derivable from the program in the answer set semantics. WFSX is, however, weaker than one might wish, not only by virtue of its being a conservative extension of WFS. It can be shown that explicit negation '$\sim$' in this semantics does not correspond to Nelson's strong negation. Consider the following example.

Example 1 *Let* $\Pi = \{\sim B; C \rightarrow B; \neg C \rightarrow D; \neg D \rightarrow C\}$*, where* B, C, D *are distinct atoms. In the WFSX semantics, the intended model of* Π *is non-empty but* Π *does not derive the atom* D*, although* $\Pi \vdash_N D$*.*

Consequently, $\mathbf{N}$ is not a deductive basis, and indeed not even a monotonic sublogic, for the inference relation corresponding to WFSX. However, $\mathbf{N}$ is the least constructive (strong negation) extension of intuitionistic logic, which is certainly a deductive basis for WFS-inference. We conclude that no strong negation extension of any intermediate logic forms a deductive basis for WSFX- inference. In fact, the latter simply fails to conform to the Vorob'ev axioms (N1–N6). One reason seems to be that although WFSX supports the *inference* from $\sim A$ to $\neg A$, it does not support the validity of the strong negation axiom N6: $\sim A \rightarrow \neg A$. From the standpoint of ordinary constructive reasoning, the main weakness of WSFX-inference is that it can sometimes assign a non-trivial semantics to a program that is inconsistent according to the logic $\mathbf{N}$.

Example 2 *Consider the following program* $\Pi = \{A \rightarrow C; B \rightarrow A; \neg B \rightarrow B; \sim A\}$ *taken from [23], where* A, B, C *are distinct atoms. According to WSFX, the intended model of* Π *contains* $\neg A, \sim A, \neg C, \neg \sim C$ *and* $\neg \sim B$*. However, it is easily seen that in constructive logic* Π *is inconsistent, since both* B *and* $\neg B$ *are* $\mathbf{N}$*-derivable.*

In the above examples, back-and-forth semantics behaves as one would expect. The program Π of Example 1 has a single back-and-forth model $\langle\{\sim B, D\}, \{\sim B, D\}\rangle$ which coincides with the answer set of Π. In Example 2 there are no back-and-forth models because, by inconsistency, there are no $\mathbf{N2}$-models at all. This is slightly different from the case of answer set semantics which does not return the inconsistent answer set Lit, but rather no answer set at all. However, it follows from the definitions and the results of [22] that every consistent answer set is (equivalent to) a back-and-forth model of the program in question. Since the converse does not hold in general, back-and-forth semantics is weaker than that of answer sets, though defined on a larger class of programs.

6 Conclusions and Future Work

We have provided a semantics, called *back-and-forth*, for logic programs and proved its equivalence with WFS or stationary semantics on normal programs and P-stable semantics on disjunctive programs. Though back-and-forth semantics is in this sense therefore not strictly new, it has a more logical flavour than other approaches and helps us to understand P-stable inference as a form of minimal model reasoning based on a well-known nonclassical logic. This in turn helps to compare P-stable inference with answer sets and other nonmonotonic reasoning systems. A further advantage is its naturalness and simplicity. The notion of unfounded set is not very intuitive, and it is not easy to see why the extension of this concept to the disjunctive case should be precisely as formulated in [6, 7]. In the case of back-and-forth semantics, the extension to the disjunctive case is trivial and so we gain additional grounds for regarding P-stable and 3-valued stable models as 'natural' generalisations of well-founded semantics.

We also sketched the way in which back-and-forth semantics can readily be extended to handle a second, strong negation in logic programs, apparently different from other similar extensions of WFS, such as WFSX. In an extended version of the paper we plan to study this semantics at greater length, in particular to explore its metalogical properties and its relation to other approaches in more detail.

There is an important problem which this paper leaves unsolved, but which future work should certainly address. The first condition on back-and-forth models (Definition 2, (i)) is a purely logical, minimality requirement, independent of the syntax of logic programs. The second condition, however, makes use of the Gelfond-Lifschitz reduct of a program

Π. It is not therefore a purely logical condition on Π itself and is not independent of the program's syntax (unlike the analogous condition for answer sets). It is of considerable interest, therefore, to find a means to re-express condition (ii) of Definition 2 algebraically, say in terms of a preferential ordering on the models of a program Π. This would open the way to extend the range of the partial stable semantics to more general classes of theories.[2]

References

[1] Alferes, J J, Damásio, C V, & Pereiera, L M, A Logic Programming System for Nonmonotonic Reasoning,*J Automated Reasoning* 14 (1995), 93-147.

[2] Chagrov, A & Zakharyaschev, M, *Modal Logic*, Oxford, Clarendon Press, 1997.

[3] van Dalen, D, Intuitionistic Logic, in D Gabbay & F Guenthner (eds), *Handbook of Philosophical Logic, Vol II*, Kluwer, Dordrecht, 1986.

[4] Brass, S & Dix, J, Characterising D-WFS: Confluence and Iterated GCWA, in L M Pereira, J J Alferes & E Orlowska (eds), *Logics in Artificial Intelligence (JELIA 96*, Springer LNAI 1126, 268-283.

[5] Diedrich, J & Herre, H, Outline of Nonmonotonic Model Theory, NTZ Report, Universität Leipzig, 1994.

[6] Eiter, T, Leone, N, & Saccà, D, Unfounded Sets and Partial Stable Models for Disjunctive Deductive Databases, *Proceedings APPIA-GULP-PRODE 1996*, 1996.

[7] Eiter, T, Leone, N, & Saccà, D, On the Partial Semantics for Disjunctive Deductive Databases, *Ann Math & Artificial Intelligence* 19 (1997), 59-96.

[8] van Gelder, A, The Alternating Fixpoint of Logic Programs with Negation, in *Proc Symp Principles of Database Systems*,ACM SIGACT-SIGMOD, ACM, 1989, 1-10.

[9] van Gelder, A, Ross, K, & Schlipf, J,The Well-Founded Semantics for General Logic Programs,*Journal of the ACM*, 38 (1991), 620-650.

[10] Gelfond, M, & Lifschitz, V, The Stable Model Semantics for Logic Programs, *Proceedings 5th Int Conf and Sympos of Logic Programming*, MIT Press, 1988, 1070-1080.

[11] Gelfond, M, & Lifschitz, V, Logic Programs with Classical Negation, in D Warren & P Szeredi (eds), *Logic Programming: Proc of the Seventh Int Conf*, MIT Press, 1990, 579-597.

[12] Gelfond, M, & Lifschitz, V, Classical Negation in Logic Programs and Disjunctive Databases, *New Generation Computing* 9 (1991), 365-387.

[2] I am very grateful to José Julio Alferes for his help and advice during the early stages of work on this paper. I would also like to thank Nicola Leone, Pasquale Rullo, Mimmo Saccà and Francesco Scarcello for helpful discussions and encouragement.

[13] Gödel, K, Zum intuitionistischen Aussagenkalkül, *Anzeiger der Akademie der Wissenschaften in Wien 69* 65-66; reprinted in [14].

[14] Gödel, K, *Collected Works, Volume I*, edited by S Feferma n*et al.* , Oxford University Press, 1986.

[15] Gurevich, Y, Intuitionistic Logic with Strong Negation, *Studia Logica* 36 (1977), 49-59.

[16] Heyting, A, Die formalen Regeln der intuitionistischen Logik, *Sitz. Berlin* 1930, 42-56.

[17] Kracht, M, On Extensions of Intermediate Logics by Strong Negation, *J Philosophical Logic*, forthcoming.

[18] Lukasiewicz, J, Die Logik und das Grundlagenproblem, in *Les Entretiens de Zurich sur les Fondaments et la Methode des Sciences Mathematiques 1938*, Zurich, 1941.

[19] Leone, N, Rullo, P, & Scarcello, F, Declarative and Fixpoint Characterizations of Disjunctive Stable Models, in *Proceedings ISLP 95*, 1995.

[20] Nelson, D, Constructible Falsity, *J Symbolic Logic* 14 (1949), 16-26.

[21] Pearce, D, & Wagner, G, Reasoning with Negative Information, I: Strong Negation in Logic Programs, *Acta Philosophica Fennica* 49, 1990.

[22] Pearce, D, A New Logical Characterisation of Stable Models and Answer Sets, in J Dix, L M Pereira & T Przymusinski (eds), *Non-monotonic Extensions of Logic Programming. Proc NMELP 96.* Springer, LNAI 1216, 1997.

[23] Pereira, L M, & Alferes, J J, Well-Founded Semantics for Logic Programs with Explicit Negation, in B Neumann (ed), *Proceedings ECAI 92*, Wiley, 1992.

[24] Pereira, L M, Alferes, J J, & Aparicio, J N, Default Theory for Well Founded Semantics with Explicit Negation, in D Pearce & G Wagner (eds),*Logics in AI. Proc JELIA 92*, LNAI 475, Springer-Verlag, 1992, 339-356.

[25] Przymusinska, H & Przymusinski,, T, Stationary Extensions of Default Theories, in *Proc 4th Workshop on Non-MonotonicReasoning*, Plymouth, VT, 1992.

[26] Przymusinski, T, Extended Stable Semantics for Normal and DisjunctivePrograms, in D Warren & P Szeredi (eds), *Logic Programming:Proceedings of the 7th International Conference*, MIT Press, 1990, 459-477.

[27] Przymusinski, T, Stable Semantics for Disjunctive Programs, *New Generation Computing* 9 (1991), 401-424.

[28] Przymusinski, T, Static Semantics for Normal and Disjunctive Logic Programs, *Ann Math & Artificial Intelligence*, 14 (1995), 323-357.

[29] Vorob'ev, N N, A Constructive Propositional Calculus with Strong Negation (in Russian), *Doklady Akademii Nauk SSR* 85 (1952), 465-468.