Analysing and Extending Well-Founded and Partial Stable Semantics using Partial Equilibrium Logic

Pedro Cabalar[1], Sergei Odintsov[2], David Pearce[3] and Agustín Valverde[4*]

[1] Corunna University (Corunna, Spain), cabalar@dc.fi.uds.es
[2] Sobolev Institute of Mathematics (Novosibirsk, Russia), odintsov@math.nsc.ru
[3] Universidad Rey Juan Carlos (Madrid, Spain), davidandrew.pearce@urjc.es
[4] University of Málaga (Málaga, Spain), a_valverde@ctima.uma.es

Abstract. In [4] a nonmonotonic formalism called *partial equilibrium logic* (PEL) was proposed as a logical foundation for the well-founded semantics (WFS) of logic programs. PEL consists in defining a class of minimal models, called *partial equilibrium* (p-equilibrium), inside a non-classical logic called HT^2. In [4] it was shown that, on normal logic programs, p-equilibrium models coincide with Przymusinki's partial stable (p-stable) models. This paper begins showing that this coincidence still holds for the more general class of disjunctive programs, so that PEL can be seen as a way to extend WFS and p-stable semantics to arbitrary propositional theories. We also study here the problem of strong equivalence for various subclasses of p-equilibrium models, investigate transformation rules and nonmonotonic inference, and consider a reduction of PEL to equilibrium logic. In addition we examine the behaviour of PEL on nested logic programs and its complexity in the general case.

1 Introduction

Of the various proposals for dealing with default negation in logic programming the *well-founded semantics* (WFS) of Van Gelder, Ross and Schlipf [20] has proved to be one of the most attractive and resilient. Particularly its favourable computational properties have made it popular among system developers and the well-known implementation XSB-Prolog[5] is now extensively used in AI problem solving and applications in knowledge representation and reasoning.

Closely related to WFS is the semantics of *partial stable models* due to Przymusinski [15]. Partial stable (henceforth p-stable) models provide a natural generalisation of stable models [8] to a multi-valued setting and on normal logic programs capture the well-founded model as a special (minimal model) case. Although the newly developing area of *answer set programming* (ASP) has focused mainly on (2-valued) stable models, there has also been a steady stream

* This research was partially supported by CICyT project TIC-2003-9001-C02.
[5] See http://www.cs.sunysb.edu/~sbprolog/xsb-page.html

of interest in the characterisation and computation of p-stable models, eg [17, 18, 6, 7, 9].

Recently [4] proposed a solution to the following long-standing problem in the foundations of WFS: which (non-modal) logic can be considered adequate for WFS in the sense that its minimal models (appropriately defined) coincide with the p-stable models of a logic program? This problem is tackled in a similar spirit to the way in which the so-called logic of *here-and-there*, HT, has been used to capture ordinary stable models and led to the development of a general nonmonotonic formalism called *equilibrium logic*, [13]. While 2-valued stable models can be characterised using the 3-valued Kripke frames of HT, for p-stable models one requires a more complex notion of frame of a kind studied by Routley [16]. These are generalisations of HT frames, referred to as HT^2 frames, and characterised by a 6-valued logic, whose negation is different from that of intuitionistic and minimal logic. To capture p-stable models in this setting a suitable notion of minimal, total HT^2 model is defined, which for obvious reasons can be called *partial equilibrium (p-equilibrium) model*. On normal logic programs, these models were shown [4] to coincide with p-stable models and so the resulting *partial equilibrium logic* (PEL) was proposed as a logical foundation for WFS and p-stable semantics. In addition [4] axiomatises the logic of HT^2-models and proves that it captures the strong equivalence of theories PEL.

The aim of the present paper is to extend the work of [4] beyond the area of normal programs treated previously. In particular we examine the case of disjunctive logic programs and show that also here p-equilibrium models coincide with p-stable models. Thus PEL can be seen also as yielding a suitable foundation for p-stable semantics and as a natural means to extend it beyond the syntax of disjunctive programs, eg to so-called nested logic programs or to arbitrary propositional theories. In summary, we shall treat the following topics. §2 describes the basic logic, HT^2, and defines partial equilibrium models. We review the main results of [4] and show that PEL captures p-stable semantics for disjunctive programs. In §3 we extend previous results on the strong equivalence of theories to special subclasses of models: the well-founded models defined in [4] and the classes of L-stable and M-stable models studied in [7]. §4 looks briefly at some of the general properties of PEL as a nonmonotonic inference relation, while §5 considers syntactic transformations of disjunctive programs, distinguishing between those preserving equivalence and those preserving strong equivalence. §6 considers the transformation technique of [9] that captures p-stable models via stable models and extends this method to PEL in general. §7 studies the behaviour of nested logic programs under PEL and some valid unfolding techniques. Finally, §8 studies the main complexity classes for PEL over propositional theories, showing that complexity is the same as that of p-stable semantics for disjunctive programs [7], while §9 concludes the paper with some open problems for future study.

2 Logical preliminaries: the logics HT^2 and PEL

We introduce the logic HT^2 and its semantics, given in terms of HT^2 *frames*, and we define *partial equilibrium logic* (PEL) in terms of minimal HT^2 models. Formulas of HT^2 are built-up in the usual way using atoms from a given propositional signature At and the standard logical constants: $\wedge$, $\vee$, $\rightarrow$, $\neg$. A set of HT^2 formulae is called a *theory*. The axiomatic system for HT^2 is described in two stages. In the first stage we include the following inference rules:

$$\frac{\alpha,\ \alpha \rightarrow \beta}{\beta}\ \text{(Modus Ponens)} \qquad\qquad \frac{\alpha \rightarrow \beta}{\neg\beta \rightarrow \neg\alpha}$$

plus the axiom schemata of *positive logic* together with:

$A1.\ \neg\alpha \wedge \neg\beta \rightarrow \neg(\alpha \vee \beta)$ $\qquad A2.\ \neg(\alpha \rightarrow \alpha) \rightarrow \beta$ $\qquad A3.\ \neg(\alpha \wedge \beta) \rightarrow \neg\alpha \vee \neg\beta$

Thus, both De Morgan laws are provable in HT^2. Moreover, axiom A2 allows us to define intuitionistic negation, '$-$', in HT^2 as: $\quad -\alpha := \alpha \rightarrow \neg(p_0 \rightarrow p_0)$. In a second stage, we further include the rule $\frac{\alpha \vee (\beta \wedge \neg\beta)}{\alpha}$ and the axioms schemata:

A4. $\ -\alpha \vee --\alpha$
A5. $\ -\alpha \vee (\alpha \rightarrow (\beta \vee (\beta \rightarrow (\gamma \vee -\gamma))))$
A6. $\ \bigwedge_{i=0}^{2}((\alpha_i \rightarrow \bigvee_{j\neq i} \alpha_j) \rightarrow \bigvee_{j \neq i} \alpha_j) \rightarrow \bigvee_{i=0}^{2} \alpha_i$
A7. $\ \alpha \rightarrow \neg\neg\alpha$
A8. $\ \alpha \wedge \neg\alpha \rightarrow \neg\beta \vee \neg\neg\beta$
A9. $\ \neg\alpha \wedge \neg(\alpha \rightarrow \beta) \rightarrow \neg\neg\alpha$
A10. $\ \neg\neg\alpha \vee \neg\neg\beta \vee \neg(\alpha \rightarrow \beta) \vee \neg\neg(\alpha \rightarrow \beta)$
A11. $\ \neg\neg\alpha \wedge \neg\neg\beta \rightarrow (\alpha \rightarrow \beta) \vee (\beta \rightarrow \alpha)$

HT^2 is determined by the above inference rules and the schemata A1-A11.

Definition 1. *A (Routley) frame is a triple $\langle W, \leq, * \rangle$, where W is a set, $\leq$ a partial order on W and $* : W \rightarrow W$ is such that $x \leq y$ iff $y^* \leq x^*$. A (Routley) model is a Routley frame together with a valuation V ie. a function from $At \times W \longrightarrow \{0, 1\}$ satisfying:*

$$V(p, u) = 1\ \ \&\ \ u \leq w\ \ \Rightarrow\ \ V(p, w) = 1 \qquad (1)$$

The valuation V is extended to all formulas via the usual rules for intuitionistic (Kripke) frames for the positive connectives $\wedge$, $\vee$, $\rightarrow$ where the latter is interpreted via the $\leq$ order:

$$V(\varphi \rightarrow \psi, w) = 1 \text{ iff for all } w' \text{ such that } w \leq w',\quad V(\varphi, w') = 1 \Rightarrow V(\psi, w') = 1$$

The main difference with respect to intuitionistic frames is the presence of the $*$ operator that is used for interpreting negation via the following condition:

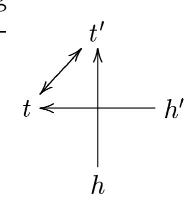
$$V(\neg\varphi, w) = 1 \ \text{ iff } \ V(\varphi, w^*) = 0.$$

A proposition φ is said to be *true* in a model $\mathcal{M} = \langle W, \leq, *, V\rangle$, if $V(\varphi, v) = 1$, for all $v \in W$. A formula φ is *valid*, in symbols $\models \varphi$, if it is true in every model. It is easy to prove by induction that condition (1) above holds for any formula φ, ie

$$V(\varphi, u) = 1 \ \ \& \ \ u \leq w \Rightarrow V(\varphi, w) = 1. \tag{2}$$

Definition 2 (HT^2 model). *An HT^2 model is a Routley model $\mathcal{M} = \langle W, \leq, R, V\rangle$ such that (i) W comprises 4 worlds denoted by h, h', t, t', (ii) $\leq$ is a partial ordering on W satisfying $h \leq t$, $h \leq h'$, $h' \leq t'$ and $t \leq t'$, (iii) the $*$ operation is determined by $h^* = t^* = t'$, $(h')^* = (t')^* = t$, (iv) V is a-valuation.*

The diagram on the right depicts the $\leq$-ordering among worlds (a strictly higher location means $\geq$) and the action of the $*$- mapping using arrows:



Truth and validity for HT^2 models are defined analogously to the previous case and from now on we let $\models$ denote the truth (validity) relation for HT^2 models. One of the main results of [4] is the following completeness theorem[6]:

Theorem 1 ([4]). *HT^2 is complete for HT^2 models, ie $\models \varphi$ iff φ is a theorem of HT^2.*

2.1 minimal models and relation to logic programs

Now, consider an HT^2 model $\mathcal{M} = \langle W, \leq, ^*, V\rangle$ and let us denote by H, H', T, T' the four sets of atoms respectively verified at each corresponding point or world h, h', t, t'. More succinctly, we can represent $\mathcal{M}$ as the pair $\langle \mathbf{H}, \mathbf{T}\rangle$ so that we group each pair of unprimed/primed worlds as $\mathbf{H} = (H, H')$ and $\mathbf{T} = (T, T')$. Notice that $H \subseteq H'$ and $T \subseteq T'$ by construction of $\mathcal{M}$ and, as a result, both $\mathbf{H}$ and $\mathbf{T}$ can be seen as 3-valued interpretations. Although the representation as a (consistent) set of literals is perhaps more frequent in the logic programming literature, a 3-valued interpretation $\mathbf{I}$ can be alternatively described by a pair of sets of atoms $I \subseteq I'$ with I containing the true atoms and I' the non-false ones. Let us use the set $\{0, 1, 2\}$ to respectively denote the possible values of atom p: *false* ($p \notin I'$), *undefined* ($p \in I' \setminus I$) and *true* ($p \in I$). As we have two 3-valued interpretations $\langle \mathbf{H}, \mathbf{T}\rangle$ we could define the possible "situations" of a formula in HT^2 by using a pair of values xy with $x, y \in \{0, 1, 2\}$. Condition (2) restricts the number of these situations to the following six $\texttt{00} := \emptyset$, $\texttt{01} := \{t'\}$, $\texttt{11} := \{h', t'\}$, $\texttt{02} := \{t, t'\}$, $\texttt{12} := \{h', t, t'\}$, $\texttt{22} := W$ where each set shows the worlds at which the formula is satisfied. Thus, an alternative way of describing HT^2 is by providing its logical matrix (see [4]) in terms of a 6-valued logic.

[6] The first stage alone defines a logic complete for the general Routley frames.

The truth-ordering relation among 3-valued interpretations $\mathbf{I}_1 \leq \mathbf{I}_2$ is defined so that $\mathbf{I}_1$ contains less true atoms and more false ones (wrt set inclusion) than $\mathbf{I}_2$. Note that by the semantics, if $\langle \mathbf{H}, \mathbf{T} \rangle$ is a model then necessarily $\mathbf{H} \leq \mathbf{T}$, since it is easy to check that this condition is equivalent to $H \subseteq T$ and $H' \subseteq T'$. Moreover, for any theory Π note that if $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi$ then also $\langle \mathbf{T}, \mathbf{T} \rangle \models \Pi$.

The ordering $\leq$ is extended to a partial ordering $\trianglelefteq$ among models as follows. We set $\langle \mathbf{H}_1, \mathbf{T}_1 \rangle \trianglelefteq \langle \mathbf{H}_2, \mathbf{T}_2 \rangle$ if (i) $\mathbf{T}_1 = \mathbf{T}_2$; (ii) $\mathbf{H}_1 \leq \mathbf{H}_2$. A model $\langle \mathbf{H}, \mathbf{T} \rangle$ in which $\mathbf{H} = \mathbf{T}$ is said to be *total*. Note that the term *total* model does not refer to the absence of undefined atoms. To represent this, we further say that a total partial equilibrium model is *complete* if $\mathbf{T}$ has the form (T, T).

We are interested here in a special kind of minimal model that we call a partial equilibrium (or p-equilibrium) model. Let Π be a theory.

Definition 3 (Partial equilibrium model). *A model $\mathcal{M}$ of Π is said to be a* partial equilibrium *model of Π if (i) $\mathcal{M}$ is total; (ii) $\mathcal{M}$ is minimal among models of Π under the ordering $\trianglelefteq$.*

In other words a p-equilibrium model of Π has the form $\langle \mathbf{T}, \mathbf{T} \rangle$ and is such that if $\langle \mathbf{H}, \mathbf{T} \rangle$ is any model of Π with $\mathbf{H} \leq \mathbf{T}$, then $\mathbf{H} = \mathbf{T}$. *Partial equilibrium logic* (PEL) is the logic determined by truth in all p-equilibrium models of a theory. Formally we can define a nonmonotonic relation of PEL-inference as follows.

Definition 4 (entailment). *Let Π be a theory, φ a formula and $\mathcal{PEM}(\Pi)$ the collection of all p-equilibrium models of Π. We say that Π entails φ in PEL, in symbols $\Pi \mid\sim \varphi$, if either (i) or (ii) holds: (i) $\mathcal{PEM}(\Pi) \neq \emptyset$ and $\mathcal{M} \models \varphi$ for every $\mathcal{M} \in \mathcal{PEM}(\Pi)$; (ii) $\mathcal{PEM}(\Pi) = \emptyset$ and φ is true in all HT^2-models of Π.*

In this definition, therefore, we consider the skeptical or cautious entailment relation; a credulous variant is easily given if needed. Clause (ii) is needed since, as Theorem 2 below makes clear, not all consistent theories have p-equilibrium models. Again (ii) represents one possible route to understanding entailment in the absence of intended models; other possibilities may be considered depending on context.

We turn to the relation between PEL and logic programs. A *disjunctive logic program* is a set of formulas (also called *rules*) of the form

$$a_1 \wedge \ldots \wedge a_m \wedge \neg b_1 \wedge \ldots \wedge \neg b_n \rightarrow c_1 \vee \ldots \vee c_k \tag{3}$$

where $m, n, k \geq 0$. For simplicity, given any rule r like (3) above, we will frequently use the names $B^+(r), B^-(r)$ and $Hd(r)$ to denote the corresponding sets $\{a_1, \ldots, a_m\}$, $\{b_1, \ldots, b_n\}$ and $\{c_1, \ldots, c_k\}$, respectively. By abuse of notation, we will also understand $B^+(r)$ as the conjunction of its atoms, whereas $B^-(r)$ and $Hd(r)$ are understood as the respective disjunctions of their atoms (remember de Morgan laws hold for negation). As usual, an empty disjunction (resp. conjunction) is understood as the constant $\bot$ (resp. $\top$). As a result, when r has the form (3) it can be represented more compactly as $B^+(r) \wedge \neg B^-(r) \rightarrow Hd(r)$. Additionally, the body of a rule r is defined as $B(r) := B^+(r) \wedge \neg B^-(r)$.

The definition of the p-stable models of a disjunctive logic program Π is given as follows. Given a 3-valued interpretation $\mathbf{I} = (I, I')$, Przymusinski's valuation[7] of formulas consists in interpreting conjunction as the minimum, disjunction as the maximum, and negation and implication as:

$$\mathbf{I}(\neg\varphi) := 2 - \mathbf{I}(\varphi) \qquad\qquad \mathbf{I}(\varphi \to \psi) := \begin{cases} 2 & \text{if } \mathbf{I}(\varphi) \leq \mathbf{I}(\psi) \\ 0 & \text{otherwise} \end{cases}$$

The constants $\bot$, $\mathbf{u}$ and $\top$ are respectively valuated as 0, 1 and 2. We say that $\mathbf{I}$ is a *3-valued model* of a formula φ, written $\mathbf{I} \models_3 \varphi$, when $\mathbf{I}(\varphi) = 2$. The *reduct* of a program Π wrt $\mathbf{I}$, denoted as $\Pi^{\mathbf{I}}$, consists in replacing each negative literal $\neg b$ in Π by the constant corresponding to $\mathbf{I}(\neg b)$. A 3-valued interpretation $\mathbf{I}$ is a *p-stable* model of Π if $\mathbf{I}$ is a $\leq -$minimal model of $\Pi^{\mathbf{I}}$.

By inspection of HT^2 and Przymusinski's interpretations of disjunctive rules it is relatively simple to check that:

Lemma 1. *For any disjunctive program Π and any HT^2 interpretation $\langle \mathbf{H}, \mathbf{T} \rangle$: $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi$ iff $\mathbf{H} \models_3 \Pi^{\mathbf{T}}$ and $\mathbf{T} \models_3 \Pi^{\mathbf{T}}$.*

Theorem 2. *A total HT^2 model $\langle \mathbf{T}, \mathbf{T} \rangle$ is a p-equilibrium model of a disjunctive[8] program Π iff the 3-valued interpretation $\mathbf{T}$ is a p-stable model of Π.*

Proof. Let $\langle \mathbf{T}, \mathbf{T} \rangle$ be a p-equilibrium model of Π. Suppose $\mathbf{T}$ is not p-stable. By Lemma 1, $\mathbf{T} \models \Pi^{\mathbf{T}}$, and so there must exist a smaller $\mathbf{H} < \mathbf{T}$ such that $\mathbf{H} \models_3 \Pi^{\mathbf{T}}$. But then $\langle \mathbf{H}, \mathbf{T} \rangle$ forms an HT^2 interpretation and, again by Lemma 1, $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi$, contradicting that $\langle \mathbf{T}, \mathbf{T} \rangle$ is in p-equilibrium. Now, let $\mathbf{T}$ be a p-stable model of Π. Then $\mathbf{T} \models \Pi^{\mathbf{T}}$ and is minimal. From Lemma 1 on $\langle \mathbf{T}, \mathbf{T} \rangle$ we conclude $\langle \mathbf{T}, \mathbf{T} \rangle \models \Pi$. Assume there exists a model $\langle \mathbf{H}, \mathbf{T} \rangle$ of Π such that $\mathbf{H} < \mathbf{T}$. By Lemma 1, $\mathbf{H} \models_3 \Pi^{\mathbf{T}}$ contradicting the minimality of $\mathbf{T}$. □

We define a further partial ordering on total models by $\langle \mathbf{T}_1, \mathbf{T}_1 \rangle \preceq \langle \mathbf{T}_2, \mathbf{T}_2 \rangle$ if both $T_1 \subseteq T_2$ and $T_2' \subseteq T_1'$. Then we say that a total HT^2 model that is $\preceq$-minimal among the p-equilibrium models of a theory Γ is a *well-founded model* of Γ. This terminology is justified by:

Theorem 3 ([4]). *If Π is a normal logic program, the unique $\preceq$-minimal p-equilibrium model of Π coincides with the well-founded model of Π in the sense of [20].*

3 Strong equivalence of theories wrt different classes of partial equilibrium models

The notion of *strong equivalence* (SE) is important both conceptually and as a potential tool for simplifying nonmonotonic programs and theories and optimising their computation. For stable semantics strong equivalence can be completely

[7] We have just directly adapted the original definitions to the current representation of 3-valued interpretations.
[8] For normal programs the theorem is proved in [4].

captured in the logic HT [10] and in ASP this fact has given rise to a lively programme of research into defining and computing different equivalence concepts [5, 22]. In the case of WFS and p-stable semantics, however, until recently there have been no studies of strong equivalence and related notions.

Here we recall the main result of [4] on strong equivalence in PEL and then consider several special classes of models. Specifically, we look at strong equivalence wrt the class of well-founded models, defined above, and the classes of L-stable and M-stable models as described by [7]. Later on we shall see that, as in the case of stable and equilibrium models, the problem of checking SE in PEL is computationally simpler than that of checking ordinary equivalence.

In the present context we say that two propositional theories Γ_1 and Γ_2 are *equivalent*, in symbols $\Gamma_1 \equiv \Gamma_2$, if they have the same p-equilibrium models and *strongly equivalent*, in symbols $\Gamma_1 \equiv_s \Gamma_2$, if for any theory Γ, theories $\Gamma_1 \cup \Gamma$ and $\Gamma_2 \cup \Gamma$ have the same p-equilibrium models.

Theorem 4 ([4]). *Theories Γ_1 and Γ_2 are strongly equivalent iff Γ_1 and Γ_2 are equivalent as HT^2 theories.*

Recall that a total model $\langle \mathbf{T}, \mathbf{T} \rangle$ is a well-founded model of Γ if it is $\preceq$ minimal in the class of all p-equilibrium models of Γ.

Definition 5. *Two HT^2 theories Γ_1 and Γ_2 are WF equivalent if for any HT^2 theory Γ, each well founded model of $\Gamma_1 \cup \Gamma$ is a well founded model of $\Gamma_2 \cup \Gamma$ and vice versa.*

Theorem 5. *Theories Γ_1 and Γ_2 are WF equivalent iff Γ_1 and Γ_2 are equivalent as HT^2 theories.*

The 'if' direction is easy. For the non-trivial converse direction we use

Lemma 2. *If theories Γ_1 and Γ_2 have different classes of p-equilibrium models, then there is a theory Γ such that theories $\Gamma_1 \cup \Gamma$ and $\Gamma_2 \cup \Gamma$ have different classes of well founded models.* ☐

Corollary 1 (of Lemma 2). *For every HT^2 theory Γ, there is an extension Γ_1 having at least one well founded model.*

We then use Lemma 2 as follows. Assume that Γ_1 and Γ_2 are not equivalent as HT^2 theories. The latter means by Theorem 4 that there is a theory Γ such that $\Gamma_1 \cup \Gamma$ and $\Gamma_2 \cup \Gamma$ have different classes of p-equilibrium models. Now we can apply Lemma 2 to obtain a theory Γ' such that $\Gamma_1 \cup \Gamma \cup \Gamma'$ and $\Gamma_2 \cup \Gamma \cup \Gamma'$ have different classes of well founded models.

Some other classes of partial stable model different from $\preceq$ minimal stable models were considered in the literature. We define the corresponding classes of p-equilibrium models.

Definition 6. *Let Γ be an HT^2 theory and $\mathcal{M} = \langle \mathbf{T}, \mathbf{T} \rangle$ a p-equilibrium model of Γ. Then (i) $\mathcal{M}$ is said to be an M* -equilibrium *model of Γ if it is $\preceq$ maximal in the class of all p-equilibrium models of Γ; (ii) $\mathcal{M}$ is said to be an L* -equilibrium *model of Γ if for any p-equilibrium model $\langle \mathbf{T_1}, \mathbf{T_1} \rangle$ of Γ the inclusion $T_1' \setminus T_1 \subseteq T' \setminus T$ implies the equality $T_1' \setminus T_1 = T' \setminus T$.*

Since the difference $T' \setminus T$ is a measure of indefiniteness of a model $\langle \mathbf{T}, \mathbf{T} \rangle$, L-equilibrium models are minimal in the class of p-equilibrium models wrt indefiniteness. Taking into account the equivalence of p-equilibrium and p-stable models of disjunctive logic programs (see Theorem **??**) we immediately obtain

Proposition 1. *Let Π be a disjunctive logic program and $\langle \mathbf{T}, \mathbf{T} \rangle$ a model of Π. Then $\langle \mathbf{T}, \mathbf{T} \rangle$ is an M (L)-equilibrium model of Π iff $\mathbf{T}$ is an M (L)-stable model of Π in the sense of [7].*

For additional motivation for L-stable and M-stable models, see [7]. The latter for example coincide on normal programs with the regular models of [23].

Definition 7. *Two HT^2 theories Γ_1 and Γ_2 are $M(L)$-equivalent if for any HT^2 theory Γ, each $M(L)$-equilibrium model of $\Gamma_1 \cup \Gamma$ is an $M(L)$- equilibrium model of $\Gamma_2 \cup \Gamma$ and vice versa.*

Theorem 6. *Theories Γ_1 and Γ_2 are $M(L)$-equivalent iff Γ_1 and Γ_2 are equivalent as HT^2 theories.*

As before the proofs of these propositions rely on the following lemma:

Lemma 3. *If theories Γ_1 and Γ_2 have different classes of p-equilibrium models, then (i) there is a theory Γ such that theories $\Gamma_1 \cup \Gamma$ and $\Gamma_2 \cup \Gamma$ have different classes of M-equilibrium models; (ii) there is a theory Γ' such that theories $\Gamma_1 \cup \Gamma'$ and $\Gamma_2 \cup \Gamma'$ have different classes of L-equilibrium models.*

4 Some Properties of Partial Equilibrium Inference

We consider some of the properties of $\mid\!\sim$ as a nonmonotonic inference relation. Generally speaking the behaviour of PEL entailment is fairly similar to that of equilibrium logic or stable model inference; however $\mid\!\sim$ fails some properties preserved by stable inference. Consider the following properties of inference:

$\varphi \in \Pi \Rightarrow \Pi \mid\!\sim \varphi$	reflexivity
$\forall i \in I, \Pi \mid\!\sim \psi_i, \Pi \cup \{\psi_i : i \in I\} \mid\!\sim \varphi \Rightarrow \Pi \mid\!\sim \varphi$	cut
$\Pi \mid\!\sim \varphi, \Pi \mid\!\sim \psi \Rightarrow \Pi \cup \varphi \mid\!\sim \psi$	cautious monotony
$\Pi \cup \varphi \mid\!\sim \alpha, \Pi \cup \psi \mid\!\sim \alpha \Rightarrow \Pi \cup (\varphi \vee \psi) \mid\!\sim \alpha$	disj. in antecedent
$\Pi \cup \varphi \mid\!\sim \alpha, \Pi \cup \neg\varphi \mid\!\sim \alpha \Rightarrow \Pi \mid\!\sim \alpha$	truth by cases
$\Pi \cup \varphi \mid\!\sim \psi \Rightarrow \Pi \mid\!\sim \varphi \rightarrow \psi$	conditionalisation
$\Pi \mid\!\sim \psi, \Pi \cup \varphi \not\mid\!\sim \psi \Rightarrow \Pi \mid\!\sim \neg\varphi$	rationality
$\Pi \mid\!\sim \psi, \Pi \cup \varphi \mid\!\sim \neg\psi \Rightarrow \Pi \mid\!\sim \neg\varphi$	weak rationality
$\Pi \mid\!\sim \varphi \rightarrow \psi, \Pi \mid\!\sim \neg\psi \Rightarrow \Pi \mid\!\sim \neg\varphi$	modus tollens

Proposition 2. *Partial equilibrium inference fails cautious monotony, truth by cases, conditionalisation, rationality and weak rationality.*

For the first condition we do however have a special case:

Proposition 3 (cautious monotony for negated formulas). *For any theory Γ, if $\Gamma \mid\!\sim \neg\varphi$ then Γ and $\Gamma \cup \{\neg\varphi\}$ have the same partial equilibrium models.*

Proposition 4. *Partial equilibrium inference satisfies reflexivity, cut, disjunction in the antecedent and modus tollens.*

5 Syntactic transformation rules for disjunctive programs

Following Brass and Dix [3], there has been considerable discussion of syntactic transformations rules that preserve the semantics of programs. For example it is well-known that while the disjunctive semantics D-WFS of [3] preserves the rule of unfolding or GPPE (see below), p-stable semantics does not. More recently [12, 5] have studied for (2-valued) stable semantics the difference between transformation rules that lead to equivalent programs and those that lead to strongly equivalent (or even uniformly equivalent) programs. With the help of HT^2 and PEL, this distinction can also be made for p-stable (p-equilibrium) semantics over disjunctive programs, or for WFS over normal programs as a special case. We consider here the situation with respect to the principal rules considered in [5]. In table 2, equivalence and strong equivalence are denoted as before by by $\equiv$, $\equiv_s$. The rules themselves are summarised in Table 1. In addition to the rules normally studied for p-stable semantics, we consider also the weaker form of unfolding, WGPPE, discussed in [5] and the rule S-IMP of Wang and Zhou [21] whose meaning is explained below.

We first give an example to show that although p-stable semantics does not obey the GPPE rule, it is not actually weaker than D-WFS.

Example 1 (from [21]). Consider the program Π comprising two rules $\neg p \rightarrow b \vee l$ and $p \vee l$. Neither b nor $\neg b$ can be derived from Π under D-WFS and the STATIC semantics. The p-equilibrium models are $\langle\{l\},\{l\}\rangle$ and $\langle\{p\},\{p\}\rangle$ and so $\Pi \mid\!\sim \neg b$.

In fact, D-WFS just allows one to derive the minimal pure disjunction $l \vee p$, whereas p-equilibrium models further derive $\neg b$. So, in this example, PEL is *strictly stronger* than D-WFS. From this and the well-known behaviour of p-stable semantics wrt GPPE, we conclude the following.

Proposition 5. *D-WFS and PEL are not comparable (even when restricted to pure disjunctions).*

Proposition 6. *Transformation WGPPE preserves strong equivalence, $\equiv_s$. In fact:* $\{(p \wedge A \rightarrow B),\ (C \rightarrow p \vee D)\} \ \vdash\ A \wedge C \rightarrow B \vee D$.

We turn now to the rule S-IMP, due to [21] and discussed in [5]. As in the case of NONMIN this is a kind of subsumption rule allowing one to eliminate a rule that is less specific than another rule belonging to the program. By definition, r stands in the S-IMP relation to r', in symbols $r \lhd r'$, iff there exists a set $A \subseteq B^-(r')$ such that (i) $Hd(r) \subseteq Hd(r') \cup A$; (ii) $B^-(r) \subseteq B^-(r') \backslash A$; (iii) $B^+(r) \subseteq B^+(r')$. For stable or equilibrium inference S-IMP is a valid rule, even preserving strong equivalence [5]. This is not so for PEL. Another rule, CONTRA, valid for stable inference, also fails in PEL.

Proposition 7. *The rules S-IMP and CONTRA are not sound for p-stable (p-equilibrium) inference.*

Table 1. Syntactic transformation rules from [5].

Name	Condition	Transformation
TAUT	$Hd(r) \cap B^+(r) \neq \emptyset$	$P' = P \setminus \{r\}$
RED^+	$a \in B^-(r_1),\ \nexists r_2 \in P:\ a \in Hd(r_2)$	$P' = P \setminus \{r_1\} \cup \{r'\}^\dagger$
RED^-	$Hd(r_2) \subseteq B^-(r_1),\ B(r_2) = \emptyset$	$P' = P \setminus \{r_1\}$
NONMIN	$Hd(r_2) \subseteq Hd(r_1),\ B(r_2) \subseteq B(r_1)$	$P' = P \setminus \{r_1\}$
GPPE	$a \in B^+(r_1),\ G_a \neq \emptyset,$ for $G_a = \{r_2 \in P \mid a \in Hd(r_2)\}$	$P' = P \setminus \{r_1\} \cup G'_a{}^\ddagger$
WGPPE	same condition as for GPPE	$P' = P \cup G'_a{}^\ddagger$
CONTRA	$B^+(r) \cap B^-(r) \neq \emptyset$	$P' = P \setminus \{r\}$
S-IMP	$r, r' \in P,\ r \lhd r'$	$P' = P \setminus \{r'\}$

† $r' : Hd(r_1) \ \leftarrow\ B^+(r_1) \cup not\,(B^-(r_1) \setminus \{a\})$.

‡ $G'_a = \{Hd(r_1) \cup (Hd(r_2) \setminus \{a\}) \ \leftarrow\ (B^+(r_1) \setminus \{a\}) \cup not\, B^-(r_1) \cup B(r_2) \mid r_2 \in G_a\}$.

Table 2. Syntactic transformations preserving equivalence

Eq.	TAUT	RED^+	RED^-	NONMIN	GPPE	WGPPE	CONTRA	S-IMP
$\equiv$	yes	yes	yes	yes	no	yes	no	no
$\equiv_s$	yes	no	yes	yes	no	yes	no	no

6 Translating partiality by atoms replication

A promising approach to implementating p-stable models for disjunctive programs has been developed by Janhunen *et al* [9]. They provide a method to capture p-stable models by (2-valued) stable models using a linear-time transformation of the program. We show here that their transformation can be extended to arbitrary propositional theories such that PEL can be reduced to ordinary equilibrium logic. Furthermore it provides an encoding of the underlying logics, of HT^2 into HT. This offers the possibility to check strong equivalence of arbitrary PEL theories by applying first this transformation, and using afterwards a satisfiability checker for arbitrary HT theories like [19].

The translation of a theory Γ, denoted $Tr(\Gamma)$, consists of a formula $p \rightarrow p'$ where p' is a new atom per each atom p occurring in Γ plus, for each $\alpha \in \Gamma$, the formula $[\alpha]$ recursively defined as follows:

$$
\begin{aligned}
[\varphi \rightarrow \psi] &:= \big([\varphi] \rightarrow [\psi]\big) \wedge [\varphi \rightarrow \psi]' &\qquad [\varphi \rightarrow \psi]' &:= [\varphi]' \rightarrow [\psi]' \\
[\neg\varphi] &:= \neg\,[\varphi]' & [\neg\varphi]' &:= \neg\,[\varphi] \\
[\varphi \oplus \psi] &:= [\varphi] \oplus [\psi] & [\varphi \oplus \psi]' &:= [\varphi]' \oplus [\psi]' \\
[p] &:= p & [p]' &:= p' \\
[\epsilon] &:= \epsilon & [\epsilon]' &:= \epsilon
\end{aligned}
$$

where $\oplus \in \{\wedge, \vee\}$ and $\epsilon \in \{\top, \bot\}$.

Example 2. The translation $\varphi = \neg(a \to \neg b) \to c$ consists of the formulas $a \to a'$, $b \to b'$, $c \to c'$ and $\neg(a' \to \neg b) \to c) \wedge \big(\neg((a \to \neg b') \wedge (a' \to \neg b)) \to c'$. $\square$

It is quite easy to see that for any disjunctive rule r like (3), its translation $[r]$ has the form $(a_1 \wedge \ldots \wedge a_m \wedge \neg b'_1 \wedge \ldots \wedge \neg b'_n \to c_1 \vee \ldots \vee c_k) \wedge$ $(a'_1 \wedge \ldots \wedge a'_m \wedge \neg b_1 \wedge \ldots \wedge \neg b_n \to c'_1 \vee \ldots \vee c'_k)$ so that $Tr(\Pi)$ amounts to Janhunen et al's transformation [9] when Π is a disjunctive logic program.

We prove next that the present generalisation of Janhunen et al's transformation works not only for representing PEL into equilibrium logic, but is actually correct at the monotonic level, i.e., it allows encoding HT^2 into HT. Let us extend first the $[\cdot]'$ notation to any set of atoms S so that $[S]' := \{p' \mid p \in S\}$.

Proposition 8. *An HT^2 interpretation $\mathcal{M}_1 = \langle (H, H'), (T, T') \rangle$ is an HT^2 model of Γ iff $\mathcal{M}_2 = \langle H \cup [H']', T \cup [T']' \rangle$ is an HT model of $Tr(\Gamma)$.*

Proposition 9. *A total HT^2 interpretation $\langle (T, T'), (T, T') \rangle$ is a* partial equilibrium model *of Γ iff $\langle T \cup [T']', T \cup [T']' \rangle$ is an equilibrium model of $Tr(\Gamma)$.*

7 Nested logic programs

The term *nested logic program* refers to the possibility of nesting default negation, conjunction and disjunction, both in the heads and bodies of the program rules. At least in what refers to rule bodies, this feature is, in fact, quite common in most Prolog interpreters, including XSB which relies on well-founded semantics. In this way, for instance, a possible XSB piece of code could look like `a :- \+ (b; c, \+ (d, \+ e))` or using logical notation:

$$\neg(b \vee c \wedge \neg(d \wedge \neg e)) \to a \tag{4}$$

The semantics for nested expressions under stable models was first described in [11]. In that paper, it was also shown that nested expressions can actually be unfolded until obtaining a non-nested program (allowing negation and disjunction in the head) by applying the following HT-valid equivalences:

(i) $F \wedge G \leftrightarrow G \wedge F$ and $F \vee G \leftrightarrow G \vee F$.
(ii) $(F \wedge G) \wedge H \leftrightarrow F \wedge (G \wedge H)$ and $(F \vee G) \vee H \leftrightarrow F \vee (G \vee H)$.
(iii) $F \wedge (G \vee H) \leftrightarrow (F \wedge G) \vee (F \wedge H)$ and $F \vee (G \wedge H) \leftrightarrow (F \vee G) \wedge (F \vee H)$.
(iv) $\neg(F \vee G) \leftrightarrow \neg F \wedge \neg G$ and $\neg(F \wedge G) \leftrightarrow \neg F \vee \neg G$.
(v) $\neg\neg\neg F \leftrightarrow \neg F$.
(vi) $F \wedge \top \leftrightarrow F$ and $F \vee \top \leftrightarrow \top$.
(vii) $F \wedge \bot \leftrightarrow \bot$ and $F \vee \bot \leftrightarrow F$.
(viii) $\neg\top \leftrightarrow \bot$ and $\neg\bot \leftrightarrow \top$.
(ix) $(F \wedge G \leftarrow H) \leftrightarrow (F \leftarrow H) \wedge (G \leftarrow H)$.
(x) $(F \leftarrow G \vee H) \leftrightarrow (F \leftarrow G) \wedge (F \leftarrow H)$.
(xi) $(F \leftarrow G \wedge \neg\neg H) \leftrightarrow (F \vee \neg H \leftarrow G)$.
(xii) $(F \vee \neg\neg G \leftarrow H) \leftrightarrow (F \leftarrow \neg G \wedge H)$.

Proposition 10. *The formulas (i)-(x) are valid in HT^2.*

Transformations (xi) and (xii), however, are not valid in HT^2. As a result the occurrence of double negation cannot be reduced in the general case to a disjunctive logic program format as shown by:

Proposition 11. *The theory $\{\neg\neg p \rightarrow p\}$ is not HT^2-equivalent to any disjunctive logic program Π (even allowing negation in the head) for signature $\{p\}$.*

One might object that this behaviour is peculiar to HT^2 and not the expected one for a well-founded semantics for nested expressions. Consider, however, the following example due to V. Lifschitz. Take the programs $\Pi_1 = \{\neg\neg p \rightarrow p\}$ and $\Pi_2 = \{p \vee \neg p\}$ which, by (xi) are HT-equivalent. Intuitively, if we could not use double negation or negation in the head, we could replace $\neg p$ by an auxiliary atom $\overline{p}$ and "define" this atom with a rule like $\overline{p} \leftarrow \neg p$. As a result, Π_1 would become $\Pi_1' = \{(\neg\overline{p} \rightarrow p), (\neg p \rightarrow \overline{p})\}$ whereas Π_2 would be now $\Pi_2' = \{(p \vee \overline{p}), (\neg p \rightarrow \overline{p})\}$. The normal program Π_1' is a typical example where p and $\overline{p}$ should become undefined in WFS. On the other hand, for Π_2' one would expect two complete models, one with p true and $\overline{p}$ false, and the symmetric one. If we remove the auxiliary atom, these two different behaviours agree, in fact, with the results in PEL for Π_1 and Π_2.

Although Proposition 11 observes that we cannot generally get rid of double negation without extending the signature, we show next that the auxiliary atom technique used in the example is in fact general enough for dealing with double negation in rule bodies, and so, thanks to transformations (i)-(x), provides a method for unfolding bodies with nested expressions.

A *disjunctive logic program with double negation* is a set of rules of the form:

$$a_1 \wedge \cdots \wedge a_n \wedge \neg b_1 \wedge \cdots \wedge \neg b_m \wedge \neg\neg c_1 \wedge \cdots \wedge \neg\neg c_s \rightarrow d_1 \vee \cdots \vee d_t \qquad (5)$$

with $m, n, s, t \geq 0$. We extend the previously defined notation so that, given a rule r like (5) $B^{--}(r)$ denotes the set of atoms $\{c_1, \ldots, c_s\}$ or, when understood as a formula, their conjunction.

Proposition 12. *Let Π be a disjunctive logic program with double negation for alphabet V. We define the disjunctive program Π' consisting of a rule*

$$\neg c \rightarrow \overline{c} \qquad (6)$$

for each double-negated literal $\neg\neg c$ occurring in Π, where $\overline{c}$ is a new atom, plus a rule r' for each rule $r \in \Pi$ where: $B^+(r') := B^+(r)$, $B^-(r') := B^-(r) \cup \{\overline{c} \mid c \in B^{--}(r)\}$ and $Hd(r') := Hd(r)$. Then Π and Π' are strongly equivalent modulo the original alphabet At, that is, $\Pi \cup \Gamma$ and $\Pi' \cup \Gamma$ have the same partial equilibrium models for any theory Γ for alphabet At. □

Example 3. Take the program consisting of rule (4). Applying transformations (i)-(x) we get that it is strongly equivalent to the pair of rules $\neg b \wedge \neg c \rightarrow a$ and $\neg b \wedge \neg\neg d \wedge \neg e \rightarrow a$ which by Proposition 12 are strongly equivalent to

$$\neg d \rightarrow \overline{d} \qquad \neg b \wedge \neg c \rightarrow a \qquad \neg b \wedge \neg\overline{d} \wedge \neg e \rightarrow a$$

modulo the original alphabet.

8 Complexity results for HT^2 and PEL

We denote by SAT_{CL} and VAL_{CL} the classes of satisfiable formulas and valid formulas respectively in Classical Logic, and SAT_{HT^2} and VAL_{HT^2} the classes of satisfiable formulas and valid formulas respectively in HT^2 logic.

Theorem 7. *SAT_{HT^2} is NP-complete and VAL_{HT^2} is coNP-complete.*

For finite-valued logics it is straightforward that the satisfiability and validity problems are at most NP-hard and coNP-hard respectively. Let φ be a formula over $\{\neg, \rightarrow, \wedge, \vee\}$ and consider the formula φ' obtained by replacing every variable p in φ by $\neg(p \rightarrow \neg p)$. The formula φ' has the following properties: every HT^2-assignment, V, verifies that $V(\varphi) \in \{00, 22\}$; if φ is satisfiable, then it has a model satisfying $V(p) \in \{00, 22\}$ for every variable p in φ'; if $W(\varphi) = 00$ for some assignment W, then there exists an assignment V such that $V(\varphi) = 00$ and $V(p) \in \{00, 22\}$ for every variable p in φ'. Finally, we have also: $\varphi \in SAT_{CL}$ if and only if $\varphi' \in SAT_{HT^2}$ and $\varphi \notin VAL_{CL}$ if and only if $\varphi' \notin VAL_{HT^2}$. Thus, the polynomial transformation of φ in φ' reduce the satisfiability and validity in classical logic to the corresponding problems in HT^2 and therefore SAT_{HT^2} is NP-complete and VAL_{HT^2} is coNP-complete.

Corollary 2. *The problem of checking the strong equivalence of theories is coNP-complete.*

Theorem 8. *The problem of deciding whether a formula in HT^2 has partial equilibrium models, partial equilibrium consistency, is Σ_2^P-hard.*

It is straightforward from the finite-valued semantics of HT^2 that the complexity is at most Σ_2^P. To prove that the complexity is in fact Σ_2^P we use that the equilibrium consistency is Σ_2^P-hard. Given a formula φ in HT, we define

$$\varphi' = \varphi \wedge \bigwedge_{p \text{ occurs in } \varphi} (\neg p \vee \neg\neg p)$$

The formula φ' has the following properties: any HT^2-model of φ', V, verifies $V(p) \in \{00, 02, 12, 22\}$ for every variable p in φ; if V is a model of φ such that $V(p) \in \{00, 02, 12, 22\}$, then the assignment V' defined as follows is also a model of φ: $V'(p) = 12$ if $V(p) = 02$ and $V'(p) = V(p)$ otherwise (this fact can be proved easily by inspection of the truth tables). So, for the formula φ', we can "forget" the value 02 and the bijection $00 \leftrightarrow 0$, $12 \leftrightarrow 1$, $22 \leftrightarrow 2$ lets us conclude that φ has equilibrium models if and only if φ' has partial equilibrium models. Thus, the polynomial transformation of φ in φ' reduces the equilibrium consistency to partial-equilibrium consistency and so this problem is Σ_2^P-hard.

Corollary 3. *The decision problem for equilibrium entailment is Π_2^P-hard.*

9 Conclusions and future work

Until recently, the well-founded and p-stable semantics have lacked a firm logical foundation of the kind that the logic of here-and-there provides for stable semantics and ASP[9]. Partial equilibrium logic supplies such a foundation and opens the way to extending these semantics beyond the syntax of normal and disjunctive programs. Here we have seen that PEL captures p-stable semantics for disjunctive programs and we have examined its behaviour on nested logic programs. An open problem for future work is whether this semantics agrees with implementations of WFS such as XSB-Prolog which allow nested expressions in rule bodies. We have also seen here how various special classes of p-stable (p-equilibrium) models, including the L-stable and M-stable models, possess a strong equivalence theorem. Moreover our complexity results for HT^2 and PEL show that testing strong equivalence in the general case (ie. PEL over theories) is computationally simpler than testing ordinary equivalence. In this respect there is agreement with the case of stable models. A major open problem is the question of strong equivalence for normal and disjunctive programs. Clearly if such programs are equivalent in HT^2 they are strongly equivalent; but, if not, it remains to be seen whether in general the addition of new formulas in the form of program rules is sufficient to establish non-strong equivalence.

The technique of [9] for capturing p-stable semantics over disjunctive programs via a reduction to ordinary stable models has been shown here to extend to arbitrary formulas and thus provide a reduction of PEL to equilibrium logic. We have seen however that nonmonotonic inference in PEL lacks several properties enjoyed by ordinary stable inference. Similarly we observed that some of the equivalence-preserving syntactic transformations applicable in ASP are no longer sound for PEL. Our results here show that PEL, like p-stable semantics, is non-comparable with extensions of WFS such as D-WFS and STATIC. However the situation wrt to the semantics WFDS of [21] is still unclear: PEL is evidently not stronger (since S-IMP fails in it), but is not yet proven to be weaker.

We hope to have shown here how PEL can provide a conceptual foundation as well as a practical tool for investigating extensions of WFS and p-stable semantics. Future work will explore the above open questions besides further issues such as how to add strong or explicit negation to PEL (and capture the WFSX semantics [14]) and how to construct a complete proof theory.

References

1. J. Alcantara, C. Damasio & L. M. Pereira. A Frame-based Characterisation of the Paraconsistent Well-founded Semantics with Explicit Negation. Unpublished draft, available at `http://centria.di.fct.unl.pt/~jfla/publications/`
2. A. Bochman. A logical foundation for logic programming I: Biconsequence relations and nonmonotonic completion, II: Semantics for logic programs. *Journal of Logic Programming*, 1998, 35: 151-170 & 171-194.

[9] The approach of [2] has a more proof theoretic flavour, while that of [1] is semantical but lacks a clear logical axiomatisation.

3. S. Brass & J. Dix. A disjunctive semantics based on unfolding and bottom-up evaluation. In *IFIP'94 Congress, Workshop FG2: Disjunctive Logic Programming and Disjunctive Databases*, 1994, pp. 83–91.
4. P. Cabalar, S. Odintsov & D. Pearce. Logical Foundations of Well-Founded Semantics in *Proceedings KR 2006*, to appear.
5. T. Eiter, M. Fink, H. Tompits and S. Woltran. Simplifying Logic Programs under Uniform and Strong Equivalence V. Lifschitz & I. Niemela (eds.), *Proceedings LPNMR 2004*, Springer, LNAI 2923, 2004.
6. T. Eiter, N. Leone & D. Saccà. On the Partial Semantics for Disjunctive Deductive Databases. *Ann. Math, & Artificial Intelligence* 17 (1997), 59-96.
7. T. Eiter, N. Leone & D. Saccà. Expressive Power and Complexity of Partial Models for Disjunctive Deductive Databases. *Theoretical Computer Science* 206 (1998), 181-218.
8. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of ICLP'88*, pp. 1070–1080, 1988. The MIT Press.
9. T. Janhunen, I. Niemelä, D. Seipel, P. Simons, and J.-H. You. Unfolding partiality and disjunctions in stable model semantics. *ACM Transactions on Computational Logic*, to appear.
10. V. Lifschitz, D. Pearce, and A. Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2(4):526–541, October 2001.
11. V. Lifschitz, L.R. Tang, and H. Turner. Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence*, 25(3–4):369–389, 1999.
12. M. Osorio, J. Navarro & J. Arrazola. Equivalence in Answer Set Programming. in *Proc. LOPSTR 2001*, LNCS 2372, Springer, 2001, pp. 57-75.
13. D. Pearce. A new logical characterization of stable models and answer sets. In *Proc. of NMELP 96*, LNCS 1216, pp. 57–70. Springer, 1997.
14. L. M. Pereira & J. J. Alferes, Well Founded Semantics for Logic Programs with Explicit Negation, in: *European Conference on Artificial Intelligence*, B. Neumann (ed.), John Wiley & Sons 1992, pp.102-106
15. Przymusinski, T. Stable semantics for disjunctive programs. *New Generation Computing* 9 (1991), 401-424.
16. R. Routley and V. Routley. The Semantics of First Degree Entailment. *Noûs*, 6, 335–359, 1972.
17. C. Ruiz & J. Minker. Compuing Stable and Partial Stable Models of Extended Disjunctive Logic Programs. In *Nonmonotonic Extensions of Logic Programming*, Springer LNCS 927, 1995, 205-229.
18. D. Seipel, J. Minker & C. Ruiz. A Characterization of the Partial Stable Models for Disjunctive Deductive Databases. in *Int. Logic Programming Symp.*, MIT Press, 1997, 245-259.
19. A. Valverde. `tabeql`: A Tableau Based Suite for Equilibrium Logic. *Proc. of JELIA'04*, LNAI 3229, pp.734–737, 2004.
20. A. van Gelder, K.A. Ross, and J.S. Schlipf. Unfounded sets and well-founded semantics for general logic programs. *JACM*, 38(3):620–650, 1991
21. K. Wang & L. Zhou. Comparisons and computation of well-founded semantics for disjunctive logic programs. *ACM Transactions on Computational Logic*, 6(2): 295-327, 2005.
22. S. Woltran. Characterizations for Relativized Notions of Equivalence in Answer Set Programming. In J. J. Alferes & J. Leite (eds), *Proc. of JELIA'04*, Springer, LNAI 3229, 2004.
23. J. You & L. Y. Yuan. Three-valued formalization of logic programming: is it needed? in *Proc. PODS 90*, ACM, 1990, 172-182.