

Partial Equilibrium Logic ^{*}

Pedro Cabalar¹, Sergei Odintsov², David Pearce³, and Agustín Valverde⁴

¹ Corunna University, Corunna, Spain
cabalar@udc.es

² Sobolev Institute of Mathematics, Novosibirsk, Russia
odintsov@math.nsc.ru

³ Computing Science and Artificial Intelligence Rey Juan Carlos University, Madrid, Spain
davidandrew.pearce@urjc.es

⁴ Applied Mathematics Dpt., University of Málaga, Spain
a_valverde@ctima.uma.es

Abstract. Partial equilibrium logic (PEL) is a new nonmonotonic reasoning formalism closely aligned with logic programming under well-founded and partial stable model semantics. In particular it provides a logical foundation for these semantics as well as an extension of the basic syntax of logic programs. In this paper we describe PEL, study some of its logical properties and examine its behaviour on disjunctive and nested logic programs. In addition we consider computational features of PEL and study different approaches to its computation.

Keywords: partial equilibrium logic, well-founded semantic, nonclassical logics, logic programming

PREPRINT DRAFT

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 [35] 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-Prolog⁵ 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 Przymusiński [29]. Partial stable (henceforth p-stable) models provide a natural generalisation of stable models 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 of interest in the characterisation and computation of p-stable models, eg [32, 33, 12, 13, 16].

Recently [6] proposed a solution to the following long-standing problem in the foundations of WFS:

^{*} Partially supported by MEC projects TIC-2003-9001-C02 and TIN2006-15455-CO3

⁵ See <http://www.cs.sunysb.edu/~sbprolog/xsb-page.html>

- 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 [6] 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* [24]. 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 [31]. These are generalisations of HT frames, referred to as HT^2 frames, and characterised by a 6-valued logic. To capture p-stable models in this setting a suitable notion of minimal, total HT^2 model is defined, which we call *partial equilibrium (p-equilibrium) model*. These models were shown to coincide with p-stable models for normal logic programs in [6] and for disjunctive logic programs in [7]. In addition [6] axiomatises the logic of HT^2 -models and proves that HT^2 captures the strong equivalence of theories. The resulting logic of p-equilibrium models is called *partial equilibrium logic* (PEL) and was proposed as a logical foundation for WFS and p-stable semantics. It can be also seen as yielding a natural means to extend WFS and p-stable semantics beyond the syntax of normal and disjunctive programs, eg to so-called nested logic programs or to arbitrary propositional theories.

In this paper we examine a range of logical and computational issues associated with PEL and its underlying logics: HT^2 and the logic of total HT^2 models, which we denote by HT^* . The paper is organised into three main parts. In the first part, comprising Sections 2, 3 and 4, we describe the logic HT^2 , introduce partial equilibrium models and review their relation to partial stable models and the well-founded semantics for logic programs. We also recall the strong equivalence theorem, we study the complexity of the main reasoning tasks associated with PEL and we axiomatise the logic HT^* . The second part comprises Sections 5, 6 and 7. Here we look at metalogical properties of the entailment relation of PEL and examine transformation rules that preserve the equivalence (ie sameness of p-equilibrium models) or strong equivalence of theories. In a similar vein we apply properties of the underlying logic, HT^2 , to determine the extent to which nested logic programs can be reduced to simpler programs. The third part of the paper deals with strategies for computing p-equilibrium models and implementing PEL. First, in Section 8, tableaux calculi are presented for HT^2 and PEL; the former is of interest in its own right as a tool for testing the strong equivalence of theories. Secondly, in Section 9 we extend the technique of Janhunen *et al* [16] that uses program transformations to reduce the computation of p-stable models for disjunctive programs to that of stable models; we show that this method can be extended to reduce PEL to equilibrium logic. Lastly, in Section 10, we consider the method of *splitting* a logic program, a familiar technique for optimising computation under the stable model semantics [18, 14]. We derive a splitting theorem for disjunctive and nested logic programs under PEL. Some further topics are discussed briefly by way of conclusion in Section 11.

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: $\wedge, \vee, \rightarrow, \neg$. We write $\mathcal{L}(At)$ to stand for the set of all well-formed formulae (ie the language) under signature At . 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:

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

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

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

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

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

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 $At \times W \rightarrow \{0, 1\}$ satisfying: (I) $V(p, u) = 1$ & $u \leq w \Rightarrow V(p, w) = 1$.

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:

$$V(\varphi \rightarrow \psi, w) = 1 \text{ iff for all } w' \geq w, V(\varphi, w') = 1 \Rightarrow V(\psi, w') = 1.$$

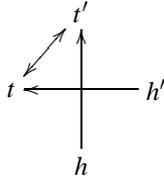
The main difference with respect to intuitionistic frames is the presence of the $*$ operator that is used for interpreting negation via the following condition:

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

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) in Definition 1 above holds for any formula φ , ie

$$V(\varphi, u) = 1 \ \& \ u \leq w \Rightarrow V(\varphi, w) = 1. \quad (1)$$

Definition 2 (HT^2 model). An HT^2 model is a Routley model $\mathcal{M} = \langle W, \leq, *, V \rangle$ 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 left 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. We have the following completeness theorem:⁶

Theorem 1 ([6]). $\models \varphi$ iff φ is a theorem of HT^2 .

2.1 HT^2 as a 6-valued logic

Now, consider an HT^2 model $\mathcal{M} = \langle W, \leq, *, V \rangle$ and let us denote by H, H', T and 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 world as $\mathbf{H} = (H, H')$ and $\mathbf{T} = (T, T')$. By construction, each of these pairs $\mathbf{I} = (I, I')$ satisfies $I \subseteq I'$, so that \mathbf{I} can be seen as a 3-valued interpretation. Given \mathbf{I} and an atom p , we use the values $\{0, 1, 2\}$ to respectively denote $p \in I$, $p \in I' \setminus I$ and $p \notin I'$. As we have two pairs like this, $\langle \mathbf{H}, \mathbf{T} \rangle$, the possible “situations” of a formula in HT^2 can be defined by a pair of values xy with $x, y \in \{0, 1, 2\}$. Condition (1) restricts the number of these situations to the following six: $00 := \emptyset$, $01 := \{t'\}$, $11 := \{h', t'\}$, $02 := \{t, t'\}$, $12 := \{h', t, t'\}$, $22 := W$, 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 in terms of a 6-valued logic. As a result, the above setting becomes an algebra of 6 cones: $\mathcal{A}^{HT^2} := \langle \{00, 01, 11, 02, 12, 22\}, \vee, \wedge, \rightarrow, \neg \rangle$ where \vee and \wedge are set theoretical join and meet, whereas \rightarrow and \neg are defined as follows: $x \rightarrow y := \{w : w \leq w' \Rightarrow (w' \in x \Rightarrow w' \in y)\}$, $\neg x := \{w : w^* \notin x\}$. The only distinguished element is 22. The lattice structure of this algebra can be described by the condition $xy \leq zt \Leftrightarrow x \leq z \ \& \ y \leq t$ and is shown in Figure 1, together with the resulting truth-tables.

2.2 Minimal models and relation to logic programs

Given a pair of 3-valued interpretations $\mathbf{I}_1 = (I_1, I_1')$ and $\mathbf{I}_2 = (I_2, I_2')$, the *truth-ordering relation* $\mathbf{I}_1 \leq \mathbf{I}_2$ holds when both $I_1 \subseteq I_2$ and $I_1' \subseteq I_2'$. 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

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

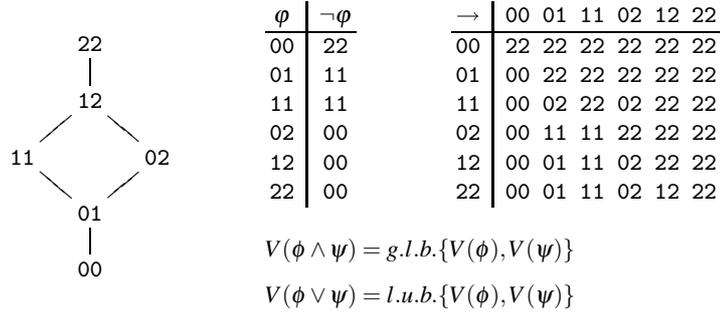


Fig. 1. Lattice structure and truth tables for the 6-valued HT^2 description.

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 $\mathbf{H} = \mathbf{T}$. We will sometimes use the abbreviation $\mathbf{T} \approx \Pi$ to denote that $\langle \mathbf{T}, \mathbf{T} \rangle$ is a p-equilibrium model of theory Π . *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 $\Pi \sim \phi$, if either (i) or (ii) holds: (i) $\mathcal{PEM}(\Pi) \neq \emptyset$ and $\mathcal{M} \models \phi$ 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

$$a_1 \wedge \dots \wedge a_m \wedge \neg b_1 \wedge \dots \wedge \neg b_n \rightarrow c_1 \vee \dots \vee c_k \quad (2)$$

where the a, b, c with subscripts range over atoms and $m, n, k \geq 0$. For simplicity, given any rule r like (2) above, we also use the symbols $B^+(r), B^-(r)$ and $Hd(r)$ to denote the corresponding sets $\{a_1, \dots, a_m\}, \{b_1, \dots, b_n\}$ and $\{c_1, \dots, c_k\}$ respectively. By abuse of notation, we understand $B^+(r)$ as the conjunction of its atoms, whereas $B^-(r)$ and $Hd(r)$ are understood as the disjunctions of their atoms (de Morgan laws hold for negation). As usual, an empty disjunction (resp. conjunction) is understood as the constant \perp (resp. \top). As a result, when r has the form (2) 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 $\mathbf{I} = (I, I')$, Przymusiński's valuation of formulas consists in interpreting conjunction as the minimum, disjunction as the maximum, negation as $\mathbf{I}(\neg\varphi) := 2 - \mathbf{I}(\varphi)$ and implication as $\mathbf{I}(\varphi \rightarrow \psi) := 2$ if $\mathbf{I}(\varphi) \leq \mathbf{I}(\psi)$ and $\mathbf{I}(\varphi \rightarrow \psi) := 0$ otherwise. The constants \perp, \mathbf{u} and \top are respectively valued 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 $\Pi^{\mathbf{I}}$.

Theorem 2 ([7]). *A total HT^2 model $\langle \mathbf{T}, \mathbf{T}' \rangle$ is a p-equilibrium model of a disjunctive program Π iff the 3-valued interpretation \mathbf{T} is a p-stable model of Π .*

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 $T'_1 \subseteq T'_2$ (intuitively, \mathbf{T}_1 has less information or defined atoms than \mathbf{T}_2). 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 the fact that 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 [35].

The notion of strong equivalence for logic programs was introduced in [19] and logically characterised for the case of programs under stable model semantics. The study of strong equivalence, its generalisations and computation, has since become a lively research area within ASP, with potential for application to program optimisation. Until now there was no analogous research programme for p-stable and WF semantics. A basis is provided however by Theorem 3 below and several extensions proved in [7].

Definition 5 ((strongly) equivalent theories). *Two theories Π, Π' are said to be (PEL)-equivalent or simply equivalent (resp. strongly equivalent), in symbols $\Pi \equiv \Pi'$ (resp. $\Pi \equiv_s \Pi'$), iff they have the same p-equilibrium models (resp. iff for any Γ , $\Pi \cup \Gamma \equiv \Pi' \cup \Gamma$).*

Theorem 3 ([6]). *Two theories Π, Π' are strongly equivalent iff they are HT^2 equivalent, ie have the same HT^2 models.*

This provides added interest in computational proof systems for HT^2 .

3 Complexity of reasoning in 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 and valid formulas respectively in HT^2 logic.

Theorem 4. SAT_{HT^2} is NP-complete and VAL_{HT^2} is coNP-complete.

Proof. 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 $p \rightarrow \neg p$. The formula φ' has the following properties: every HT^2 -assignment, V , verifies that $V(\varphi) \in \{00, 02\}$; 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 into φ' reduces the satisfiability and validity problems 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. \square

Corollary 1. The problem of checking the strong equivalence of theories is coNP-complete.

Theorem 5. The problem of deciding whether a formula in HT^2 has partial equilibrium models (partial equilibrium consistency) is Σ_2^P -complete.

Proof. It is straightforward from the finite-valued semantics of HT^2 to see that the complexity is at most Σ_2^P . To prove that the complexity is in fact Σ_2^P we use the fact that equilibrium consistency is Σ_2^P -complete. 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 into φ' reduces equilibrium consistency to partial-equilibrium consistency and so this problem is Σ_2^P -complete. \square

Corollary 2. The decision problem for equilibrium entailment is Π_2^P -complete.

4 The Logic of Total Models

Total models play an important role in the definition of PEL since p-equilibrium models are a special kind of total model. We describe their logic. First note that total models can be distinguished among all HT^2 -models via the scheme $\neg\neg\varphi \rightarrow \varphi$. For an HT^2 model $\mathcal{M} = \langle (H, H'), (T, T') \rangle = \langle W^{HT^2}, \leq, *, V \rangle$, where $W^{HT^2} = \{h, h', t, t'\}$ set

$$\Delta_w^{\mathcal{M}} := \{\varphi : V(\varphi, w) = 1\}$$

for $w \in W^{HT^2}$. Obviously, $H \subseteq \Delta_h^{\mathcal{M}}$, $H' \subseteq \Delta_{h'}^{\mathcal{M}}$, etc. We omit the superscript \mathcal{M} if it does not lead to confusion.

Proposition 1. *The following items are equivalent:*

1. $\langle \mathbf{H}, \mathbf{T} \rangle \models \neg\neg\varphi \rightarrow \varphi$ for any φ ,
2. $\mathbf{H} = \mathbf{T}$,
3. $\Delta_h = \Delta_t$ and $\Delta_{h'} = \Delta_{t'}$.

Proof. Condition 1 is equivalent to $\neg\neg\varphi \in \Delta_w \Rightarrow \varphi \in \Delta_w$ for all w and φ . By definition of the $*$ operation $\neg\neg\varphi \in \Delta_w \Leftrightarrow \varphi \in \Delta_{w^{**}}$. Taking into account $t^{**} = t$ and $t'^{**} = t'$ we obtain that condition 1 is equivalent to the inclusions

$$\Delta_t \subseteq \Delta_h \text{ and } \Delta_{t'} \subseteq \Delta_{h'}.$$

Inverse inclusions hold in all HT^2 -models, therefore $1 \Leftrightarrow 3$. Implication $3 \Rightarrow 2$ is obvious. The inverse implication follows by a routine induction on the structure of formulas. \square

Let us set $HT^* := HT^2 + \{\neg\neg p \rightarrow p\}$. From the last proposition it follows that the number of possible situations of a formula in a total HT^2 -model is reduced to the following three, $00 := \emptyset$, $11 := \{h', t'\}$, $22 := \{h, h', t, t'\}$, where each set shows the worlds at which the formula is satisfied. Thus, the logic HT^* can be characterised by the three-element algebra: $\mathcal{A}^{HT^*} := \langle \{00, 11, 22\}, \vee, \wedge, \rightarrow, \neg \rangle$ with the only distinguished element 22 and operations determined as the restrictions of the respective operation of the algebra \mathcal{A}^{HT^2} . It is routine to check that the set $\{00, 11, 22\}$ is closed under \mathcal{A}^{HT^2} -operations.

At the same time, HT^* differs from Przymusiński's logic Prz_3 [30] as well as from \mathbf{N}_3 [34, 25], classical explosive logic with strong negation. All these logics are three-valued and the operations \vee and \wedge determine the structure of a linearly ordered lattice on the set of truth-values. If we denote the least truth-value in all these logics by 00, the greatest by 22, and the intermediate by 11, we see that all the logics have the same connectives \neg, \vee, \wedge , but different implications (see Fig 2). Comparing HT^* and \mathbf{N}_3 we note the following

Proposition 2. $HT^* \not\subseteq \mathbf{N}_3$, $\neg(p \rightarrow q) \leftrightarrow (p \wedge \neg q) \in \mathbf{N}_3 \setminus HT^*$.

For the comparison of HT^* and Prz_3 , recall that the language of Prz_3 contains also the necessity operator l ($l22 = 22$, $lx = 00$ otherwise) and \rightarrow_{Prz_3} can be defined via \neg, \vee, \wedge and l : $\varphi \rightarrow_{Prz_3} \psi := (\neg l\varphi \vee l\psi) \wedge (\neg l\neg\psi \vee l\neg\varphi)$.

At the same time, $l\varphi$ can be defined in HT^* as $\neg(\varphi \rightarrow_{HT^*} \neg\varphi)$.

\rightarrow_{HT^*}	00 11 22
00	22 22 22
11	00 22 22
22	00 11 22

\rightarrow_{N_3}	00 11 22
00	22 22 22
11	22 22 22
22	00 11 22

\rightarrow_{Prz_3}	00 11 22
00	22 22 22
11	00 22 22
22	00 00 22

Fig. 2. Truth tables for implication.

Proposition 3. *The logic Prz_3 is definable in HT^* .* □

A simple axiomatisation of HT^* modulo the basic logic N^* is given by the following

Proposition 4. $HT^* = N^* + \{p \vee (p \rightarrow q) \vee \neg q, p \leftrightarrow \neg\neg p, p \wedge \neg p \rightarrow q \vee \neg q\}$.

In fact, the proof of this statement is a simplified version of the completeness proof for HT^2 in [6]. Thus, we obtain HT^* by extending the intuitionistic fragment to HT and adding the elimination of double negation and the Kleene axiom. Despite the fact that HT^* and HT have the same intuitionistic fragment, they have different negations and $HT^* \neq HT$. We can obtain HT from HT^2 in the following way.

Proposition 5. *The addition to HT^2 of axiom (I) = $\neg\phi \wedge \phi \rightarrow \perp$, is equivalent to the condition $T = T'$.*

Proposition 6. *The addition to HT^2 of De Jongh and Hendrik's axiom (used to obtain HT from intuitionistic logic), (dJH) = $\phi \vee (\phi \rightarrow \psi) \vee \neg\phi$ is equivalent to the condition: $T, H' \in \{H, T'\}$.*

The last two statements can be checked directly, and the next one follows from them.

Proposition 7 (reduction to HT). $HT = HT^2 \cup (I) \cup (dJH)$.

5 Some Properties of Partial Equilibrium Inference

Since the early days of research on nonmonotonic logics it became commonplace to study and compare logical systems with respect to general conditions on inference that they satisfy. The aim was not only to classify systems without monotonicity but also to select properties considered to be especially interesting or desirable. These properties were catalogued in works such as [21, 17] which established many of the standard conditions on inference that have been studied thereafter. For an authoritative account see [22].

Likewise it became a matter of routine to compare different approaches to the semantics of logic programs according to the abstract properties satisfied by their associated inference relations, see eg [9, 10] for the case of normal programs.

We consider some of the properties of \vdash 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 \vdash fails some properties preserved by stable inference. Consider the following properties of inference:

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

Proposition 8. *Partial equilibrium inference fails cautious monotony, truth by cases, conditionalisation, rationality and weak rationality.*

Proof. The following variation of an example in [29] provides a counterexample to cautious monotony. Let Π be the disjunctive program:

$$\begin{aligned} \neg c \rightarrow work \vee sleep \vee tired, \quad \neg c \wedge \neg tired \rightarrow work, \quad \neg c \wedge \neg sleep \rightarrow tired, \\ \neg c \wedge \neg work \rightarrow sleep, \quad a \vee b, \quad a \rightarrow c \end{aligned}$$

Program Π has just one partial equilibrium model $(\{a, c\}, \{a, c\})$, and so entails both a and c . However, program $\Pi \cup \{c\}$ has an additional partial equilibrium model $(\{b, c\}, \{b, c\})$ and so a is not entailed any more.

The second property, truth by cases, is not a valid principle of constructive reasoning and fails already in the underlying monotonic logic HT^2 . To see this, let Π be empty and set $\alpha = (\varphi \vee \neg \varphi)$.

Now consider the program Π :

$$g \vee d, \neg f \rightarrow f, f \neg g \rightarrow a \vee b \vee c, \neg b \rightarrow a, \neg c \rightarrow b, \neg a \rightarrow c$$

Π has two p-equilibrium models $\mathcal{M}_1 = (\{d\}, \{a, b, c, f, d\})$ and $\mathcal{M}_2 = (\{g\}, \{a, b, c, f, g\})$. Evidently $\mathcal{M}_1 \not\models f \rightarrow g$. However, $\Pi \cup \{f\}$ has the unique p-equilibrium model: $(\{f, g\}, \{a, b, c, f, g\})$ in which g is true; falsifying the property of conditionalisation. \square

For the first condition we do however have two special cases, first:

Proposition 9 (cautious monotony for negated formulas). *For any theory Γ , if $\Gamma \vdash \neg \varphi$ then Γ and $\Gamma \cup \{\neg \varphi\}$ have the same partial equilibrium models.*

Secondly, since \vdash agrees with well-founded inference on normal programs, when Π has this special form cautious monotony holds.

Proposition 10. *Partial equilibrium inference satisfies reflexivity, cut, disjunction in the antecedent and modus tollens.*

To see this we use a lemma easily derivable by a simple inspection on semantic definitions for PEL:

Lemma 1. (i) For any HT^2 model \mathcal{M} and formulas φ, ψ , $\mathcal{M} \models \varphi \vee \psi \Rightarrow \mathcal{M} \models \varphi$ or $\mathcal{M} \models \psi$. (ii) Let \mathcal{M} be a partial equilibrium model of Π such that $\mathcal{M} \models \varphi$. Then \mathcal{M} is an partial equilibrium model of $\Pi \cup \{\varphi\}$.

Proof (Proposition 9). Clearly, by Lemma 1 any partial equilibrium model of Γ in which $\neg\varphi$ holds is also a partial equilibrium model of $\Gamma \cup \{\neg\varphi\}$. So it remains to show the converse. Let $\mathcal{M} = \langle (T, T'), (T, T') \rangle$ be a partial equilibrium model of $\Gamma \cup \{\neg\varphi\}$. As \mathcal{M} is also a model for Γ , it remains to prove its \leq -minimality. Assume there exists some strictly \triangleleft -lower model $\mathcal{M}' = \langle (H, H'), (T, T') \rangle$. As $\mathcal{M} \models \neg\varphi$ is equivalent to $\mathcal{M}, t' \not\models \varphi$, we actually have that $\mathcal{M}', t' \not\models \varphi$ too, and so $\mathcal{M}' \models \neg\varphi$. But then \mathcal{M}' is a model for $\Gamma \cup \{\neg\varphi\}$ which contradicts the minimality of \mathcal{M} as a partial equilibrium model of that program. \square

Proof (Proposition 10). Reflexivity is straightforward. For cut, apply Lemma 1. For disjunction in the antecedent observe that every partial equilibrium model of $\Pi \cup (\varphi \vee \psi)$ is a total model of Π in which either φ holds or ψ holds. By Lemma 1 (ii) it follows that in the former case it must be a p-equilibrium model of $\Pi \cup \varphi$ and in the latter case it is a p-equilibrium model of $\Pi \cup \psi$. In each case by assumption α is true in the model.

Assume the hypotheses of modus tollens. Then for any p-equilibrium model \mathcal{M} and world t' , $\mathcal{M}, t' \not\models \psi$, while for all worlds u , $\mathcal{M}, u \models \varphi \Rightarrow \mathcal{M}, u \models \psi$. Hence $\mathcal{M}, t' \not\models \varphi$ and so $\mathcal{M} \models \neg\varphi$. \square

6 Syntactic transformation rules for disjunctive programs

Following Brass and Dix [1–4], there has been considerable discussion of syntactic transformation rules that preserve the semantics of programs. For example, while the disjunctive semantics D-WFS of [1, 2] is well-known to preserve the rule of unfolding or the General Principle of Partial Evaluation (GPPE for short, see below), p-stable semantics does not (see Example 6.3 in [33] for a counterexample). More recently [23, 11] 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 uniform 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 [11]. In Table 2, equivalence and strong equivalence are denoted as before by \equiv , \equiv_s . The rules themselves are summarised in Table 1. In addition to the rules normally studied for p-stable, we consider also the weaker form of unfolding, WGPPE, discussed in [11] and the rule S-IMP of Wang and Zhou [36] whose meaning is explained below. We first give an example to show that although p-stable does not obey the GPPE rule, it is not actually weaker than D-WFS.

Example 1 (from [36]). 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 $\Pi \sim \neg b$.

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*

Table 1. Syntactic transformation rules from [11].

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	$a \in B^+(r_1), G_a \neq \emptyset, \text{ for } G_a = \{r_2 \in P \mid a \in Hd(r_2)\}$	$P' = P \setminus \{r_1\} \cup G_a^\ddagger$
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	$r, r' \in P, r \triangleleft r'$	$P' = P \setminus \{r'\}$

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

[‡] $G_a^\ddagger = \{Hd(r_1) \cup (Hd(r_2) \setminus \{a\}) \leftarrow (B^+(r_1) \setminus \{a\}) \cup \text{not} B^-(r_1) \cup B(r_2) \mid r_2 \in G_a\}$.

than D-WFS. On the other hand, unsoundness of GPPE in PEL also leads to examples where PEL is strictly weaker than D-WFS. For instance, given the program $\Pi_1 = \{a \vee b, \neg a \rightarrow a, a \wedge b \rightarrow c\}$, in D-WFS we can apply GPPE for atom b and then TAUT to obtain $\Pi_2 = \{a \vee b, \neg a \rightarrow a\}$ and derive $\neg c$. However, Π_1 has two p-equilibrium models corresponding to $\langle \{a\}, \{a\} \rangle$ and $\langle \{b\}, \{a, b, c\} \rangle$, and since the latter leaves c undefined, $\neg c$ cannot be derived. To sum up:

Proposition 11. *D-WFS and PEL are not comparable (even when restricted to pure disjunctions).*

Proposition 12. *The transformation WGPPE preserves strong equivalence, \equiv_s . In fact: $\{(p \wedge A \rightarrow B), (C \rightarrow p \vee D)\} \vdash A \vee C \rightarrow B \vee D$.*

Proof. First, it is easy to see that transformations TAUT and NONMIN preserve strong equivalence \equiv_s , by simple inspection of HT^2 semantics. Then, on the one hand, by NONMIN applied on $C \rightarrow p \vee D$ we obtain $A \wedge C \rightarrow p \vee D$. Since $A \wedge C \rightarrow A \vee D$ by TAUT we can mix both into $A \wedge C \rightarrow (p \vee D) \wedge (A \vee D)$ – see property (ix) in Section 7. By (iii) in that section, this is equivalent to $\alpha : A \wedge C \rightarrow p \wedge A \vee D$. On the other hand, applying NONMIN on $p \wedge A \rightarrow B$ we obtain $p \wedge A \rightarrow B \vee D$. As $D \rightarrow B \vee D$ follows from TAUT, we can combine both into $\beta : p \wedge A \vee D \rightarrow B \vee D$ (see (x) in Section 7). Finally, the result follows from transitivity of \rightarrow applied to α and β . \square

We turn now to the rule S-IMP, due to [36] and discussed in [11]. As in the case of NONMIN this is a kind of subsumption transformation 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 $r \triangleleft r'$, iff there exists a set $A \subseteq B^-(r')$ such that (i) $Hd(r) \subseteq Hd(r') \cup A$; (ii) $B^-(r) \subseteq B^-(r') \setminus A$; (iii) $B^+(r) \subseteq B^+(r')$. For stable or equilibrium inference S-IMP is a valid rule, even preserving strong equivalence [11]. This is not so for PEL. Another rule, CONTRA, valid for stable inference, also fails in PEL.

Proposition 13. *The rules S-IMP and CONTRA are not sound for p-stable (p-equilibrium) inference.*

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

Proof. The proof is by counterexample: let Π be the program

$$r_1 : a \rightarrow b \vee c, \quad r_2 : a \wedge \neg c \rightarrow b \quad r_3 : \neg a \rightarrow a \quad r_4 : \neg a \rightarrow c$$

And let Π' be $\Pi \setminus \{r_2\}$. We claim: Π' is an S-IMP reduction of Π . Evidently $r_1 \triangleleft r_2$ and so r_2 is removed from Π . The only other candidates to be in the \triangleleft relation are r_3, r_4 where $\{a\}$ is the subset of $B^-(r_3)$ or $B^-(r_4)$, but they fail the second requirement for $r \triangleleft r'$ that $B^-(r)$ is a subset of $B^-(r') \setminus \{a\}$. There are no other possibilities to obtain an S-IMP reduction. It is straightforward to check that Π has the unique p-equilibrium model $\langle \{\}, \{a, b, c\} \rangle$ while Π' has the unique p-equilibrium model $\langle \{\}, \{a, b\} \rangle$. Since Π and Π' have different p-equilibrium models, S-IMP reduction is not a sound transformation. \square

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 as far as rule bodies are concerned, 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:

$$\neg(b \vee c \wedge \neg(d \wedge \neg e)) \rightarrow a \quad (3)$$

The semantics for nested expressions under stable models was first described in [20]. In that paper, it was also shown that nested expressions can actually be unfolded until a non-nested program (allowing negation and disjunction in the head) is obtained 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) $\neg\neg F \leftrightarrow F$.
- (vi) $F \wedge \top \leftrightarrow F$ and $F \vee \top \leftrightarrow \top$.
- (vii) $F \wedge \perp \leftrightarrow \perp$ and $F \vee \perp \leftrightarrow F$.
- (viii) $\neg\top \leftrightarrow \perp$ and $\neg\perp \leftrightarrow \top$.
- (ix) $(F \rightarrow G \wedge H) \leftrightarrow (F \rightarrow G) \wedge (F \rightarrow H)$.

- (x) $(F \vee G \rightarrow H) \leftrightarrow (F \rightarrow H) \wedge (G \rightarrow H)$.
- (xi) $(F \wedge \neg\neg G \rightarrow H) \leftrightarrow (F \rightarrow H \vee \neg G)$.
- (xii) $(F \rightarrow G \vee \neg\neg H) \leftrightarrow (F \wedge \neg H \rightarrow G)$.

Proposition 14. *The formulas (i)-(x) are valid in HT^2 .*

Proof. Validity of the equivalences (i),(ii),(iii),(vi),(vii) and (viii) are straightforward, since HT^2 satisfaction of conjunction, disjunction and truth constants is defined as in classical logic. For the remaining proofs, let us use $\forall w'$ to refer to any world w' such that wRw' .

- (iv) This follows from the chain of equivalent conditions: $M, w \models \neg(F \vee G) \Leftrightarrow M, w^* \not\models (F \vee G) \Leftrightarrow (M, w^* \not\models F \text{ and } M, w^* \not\models G) \Leftrightarrow (M, w \models \neg F \text{ and } M, w \models \neg G) \Leftrightarrow M, w \models \neg F \wedge \neg G$. The proof for negated conjunctions is completely analogous.
- (v) This is obtained by the equivalence $M, w \models \neg\neg\neg F \Leftrightarrow M, ((w^*)^*)^* \not\models F$ and the fact that $((w^*)^*)^* = w^*$ (see diagram in Definition 2).
- (ix) This similarly follows from the sequence:

$$\begin{aligned} M, w \models F \rightarrow G \wedge H &\Leftrightarrow \forall w'. M, w' \not\models F \text{ or } (M, w' \models G \text{ and } M, w' \models H) \\ &\Leftrightarrow \forall w'. (M, w' \not\models F \text{ or } M, w' \models G) \text{ and } (M, w' \not\models F \text{ or } M, w' \models H) \\ &\Leftrightarrow M, w \models (F \rightarrow G) \wedge (F \rightarrow H). \end{aligned}$$

- (x) This follows from the equivalent conditions:

$$\begin{aligned} M, w \models F \vee G \rightarrow H &\Leftrightarrow \forall w'. M, w' \not\models (F \vee G) \text{ or } M, w' \models H \\ &\Leftrightarrow \forall w'. (M, w' \not\models F \text{ and } M, w' \not\models G) \text{ or } M, w' \models H \\ &\Leftrightarrow \forall w'. (M, w' \not\models F \text{ or } M, w' \models H) \text{ and } (M, w' \not\models G \text{ or } M, w' \models H) \\ &\Leftrightarrow M, w \models (F \rightarrow H) \wedge (G \rightarrow H). \quad \square \end{aligned}$$

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 15. *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 $\{p\}$.*

Proof. Let us try to build a program Π with the same HT^2 models as $\{\neg\neg p \rightarrow p\}$. To construct a given rule $r \in \Pi$, we may have that p may occur both positively or negatively and both in the head or in the body. This leads to $2^4 = 16$ possible rules. If we ignore the rules that directly constitute an inconsistency or a tautology, it is not difficult to see that the remaining set of non-trivial rules amounts to the following eight: $\{r_1 : (p \rightarrow \perp), r_2 : (\neg p \rightarrow \perp), r_3 : (p \wedge \neg p \rightarrow \perp), r_4 : p, r_5 : (\neg p \rightarrow p), r_6 : \neg p, r_7 : (p \leftarrow \neg p), r_8 : (p \vee \neg p)\}$. Now, the models of $\neg\neg p \rightarrow p$ are exactly $M_1 = \langle\langle\emptyset, \{p\}\rangle, \langle\emptyset, \{p\}\rangle\rangle$ and $M_2 = \langle\langle\{p\}, \{p\}\rangle, \langle\{p\}, \{p\}\rangle\rangle$. It can be easily seen that M_1 is countermodel of rules r_1, r_2, r_3, r_4 and r_8 whereas M_2 is countermodel of r_6, r_7 . So none of these rules can be included in Π and the only remaining possibility is $\Pi = \{r_5\}$. However, this program has in fact more models apart from M_1 and M_2 – for instance, $\langle\langle\emptyset, \emptyset\rangle, \langle\{p\}, \{p\}\rangle\rangle$. \square

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 \bar{p} and “define” this atom with a rule like $\neg p \rightarrow \bar{p}$. As a result, Π_1 would become $\Pi'_1 = \{(\neg\bar{p} \rightarrow p), (\neg p \rightarrow \bar{p})\}$ whereas Π_2 would be now $\Pi'_2 = \{(p \vee \bar{p}), (\neg p \rightarrow \bar{p})\}$. The normal program Π'_1 is a typical example where p and \bar{p} should become undefined in WFS. On the other hand, for Π'_2 one would expect two complete models, one with p true and \bar{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 15 shows 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:

$$a_1 \wedge \dots \wedge a_n \wedge \neg b_1 \wedge \dots \wedge \neg b_m \wedge \neg\neg c_1 \wedge \dots \wedge \neg\neg c_s \rightarrow d_1 \vee \dots \vee d_t \quad (4)$$

with $m, n, s, t \geq 0$. We extend the previously defined notation so that, given a rule r like (4) $B^{--}(r)$ denotes the set of atoms $\{c_1, \dots, c_s\}$ or, when understood as a formula, their conjunction.

Proposition 16. *Let Π be a disjunctive logic program with double negation for alphabet V . We define the disjunctive program Π' consisting of a rule*

$$\neg c \rightarrow \bar{c} \quad (5)$$

for each double-negated literal $\neg\neg c$ occurring in Π , where \bar{c} is a new atom, plus a rule r' for each rule $r \in \Pi$ where: $B^+(r') := B^+(r)$, $B^-(r') := B^-(r) \cup \{\bar{c} \mid c \in B^{--}(r)\}$ 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 .

Lemma 2. *For any pair of interpretations $M' \sqsubseteq M$ and for any world w , we have $M', w \models \neg\phi$ iff $M, w \models \neg\phi$.*

Lemma 3. *Let M be a total model of $p \leftrightarrow \neg\phi$. Then, for any $M' \sqsubseteq M$, we have: $M' \models p \leftrightarrow \neg\phi$ iff $M' \models \neg\phi \rightarrow p$.*

Proof. The left to right direction is trivial. For the right to left direction, assume $M' \models p \leftrightarrow \neg\phi$ but $M' \not\models p \rightarrow \neg\phi$. Then, for some world w , we get $M', w \models p$ and $M', w \not\models \neg\phi$. By Lemma 2 the latter implies $M, w \not\models \neg\phi$, but as $M \models p \leftrightarrow \neg\phi$, we conclude $M, w \not\models p$. However, as $M' \sqsubseteq M$, $M, w \not\models p$ implies $M', w \not\models p$ reaching a contradiction. \square

Proof (Proposition 16). By $M|_V$ we denote model M modulo alphabet V . Assume M_0 is a partial equilibrium model of $\Pi \cup \Gamma$. Then, we can always build M by adding to M_0 ,

at each world w , all the atoms \bar{c} for which $M_0, w \models \neg c$. Clearly, M is still a total model, whereas by construction, $M \models \bar{c} \leftrightarrow \neg c$. As a result, it is easy to see that $M \models \Pi'$, and thus, $M \models \Pi' \cup \Gamma$ as Γ is restricted to alphabet V , for which M_0 and M coincide. Now, to see that M is in equilibrium for $\Pi' \cup \Gamma$, assume we had a smaller model $M' \triangleleft M$. Since for any $c \in B^{-}(r)$, $M' \models \bar{c} \leftrightarrow \neg c$ whereas $M \models \bar{c} \leftrightarrow \neg c$ we apply Lemma 3 to conclude that $M' \models \bar{c} \leftrightarrow \neg c$ too. From this fact and Lemma 2 we get that $M', w \models \bar{c}$ iff $M, w \models \bar{c}$, for all w . Thus, M and M' coincide in valuation of atoms lc , and so $M'|_V$ is strictly smaller than $M|_V = M_0$. Finally, as $M' \models \bar{c} \leftrightarrow \neg c$ formulas \bar{c} and $\neg c$ are interchangeable in M' , and so $M' \models \Pi \cup \Gamma$, ie, $M'|_V \models \Pi \cup \Gamma$ which contradicts the minimality of M_0 .

Assume now that a given M is a partial equilibrium model of $\Pi' \cup \Gamma$. Now, note that the only rule head for each auxiliary atom \bar{c} is the one from (5). By supportedness, if $M, w \models \bar{c}$ then $M, w \models \neg c$ for all worlds w . This fact, together with $M \models (5)$ implies $M \models \bar{c} \leftrightarrow \neg c$, and so it is easy to see that $M \models \Pi \cup \Gamma$, ie $M|_V \models \Pi \cup \Gamma$. To show that $M|_V$ is in equilibrium, assume we had some smaller $M_0 \models \Pi \cup \Gamma$. Then we extend M_0 to M' so that $M', w \models \bar{c}$ if $M_0, w \models \neg c$, for all worlds w . Now, we have the equivalence chain: $M, w \models \bar{c} \Leftrightarrow M, w \models \neg c \Leftrightarrow M|_V, w \models \neg c$, which by Lemma 2 is equivalent to $M_0, w \models \neg c$ and this is equivalent to $M', w \models \bar{c}$ by construction of M' . In this way, M' and M coincide in valuation of atoms \bar{c} , and so $M_0 \triangleleft M|_V$ implies $M' \triangleleft M$. Finally, by construction, $M' \models \bar{c} \leftrightarrow \neg c$, and so $M' \models \Pi' \cup \Gamma$ contradicting the minimality of M . \square

Example 2. Take the program consisting of rule (3). Applying transformations (i)-(x) we get that it is strongly equivalent to the pair of rules $\neg b \wedge \neg c \rightarrow a$ and $\neg b \wedge \neg \bar{d} \wedge \neg e \rightarrow a$ which by Proposition 16 are strongly equivalent to

$$\neg d \rightarrow \bar{d} \quad \neg b \wedge \neg c \rightarrow a \quad \neg b \wedge \neg \bar{d} \wedge \neg e \rightarrow a$$

modulo the original alphabet.

8 A Tableau Calculus for PEL

We describe a tableaux system for HT^2 using the standard methods for finite-valued logics [15, 25]. The formulas in the tableau nodes are labelled with a set of truth-values, named *signs*, and these signs are propagated to the subformulas using the *expansion rules*. The family of the signs depends on the logic in question and it is possible to describe several tableaux systems for the same logic. For HT^2 we are going to use the following family of signs

$$\{00\}, \{01\}, \{11\}, \{02\}, \{22\}, \{01, 11\}, [\leq 01], [\leq 11], [\leq 12], [\geq 01], [\geq 02], [\geq 12],$$

where $[\geq v] = \{w \in 6 \mid w \geq v\}$, and $[\leq v] = \{w \in 6 \mid w \leq v\}$. Each expansion rule is obtained by inspection of the truth tables. For example, the rule

$$\frac{[\geq 01]: \varphi \rightarrow \psi}{\{22\}: \varphi \mid [\geq 01]: \psi}$$

in Figure 3 means that to evaluate the formula $\varphi \rightarrow \psi$ in $\{v \mid v \geq 01\}$, it is necessary either to assign 22 to φ or to assign a value greater than 01 to ψ .

$\{22\}: \varphi \rightarrow \psi$				$\{00\}: \varphi \rightarrow \psi$		$[\leq 01]: \varphi \rightarrow \psi$			
$\{00\}: \varphi$	$\{22\}: \psi$	$[\leq 01]: \varphi$	$[\leq 12]: \varphi$	$\{11\}: \varphi$	$\{02\}: \varphi$	$[\geq 01]: \varphi$	$[\geq 01]: \varphi$		
		$[\geq 01]: \psi$	$[\geq 12]: \psi$	$\{11\}: \psi$	$\{02\}: \psi$	$\{00\}: \psi$	$\{00\}: \psi$		
$[\leq 12]: \varphi \rightarrow \psi$									
$[\geq 01]: \varphi \rightarrow \psi$									
$\{22\}: \varphi$	$[\geq 01]: \psi$	$[\geq 01]: \varphi$	$\{11\}: \varphi$	$\{11\}: \varphi$	$\{02\}: \varphi$	$[\geq 12]: \varphi$	$[\geq 12]: \varphi$		
		$\{00\}: \psi$	$[\leq 01]: \psi$	$\{02\}: \psi$	$\{01,11\}: \psi$	$[\leq 11]: \psi$	$\{02\}: \psi$		
$[\geq 12]: \varphi \rightarrow \psi$		$\{11\}: \varphi \rightarrow \psi$		$\{02\}: \varphi \rightarrow \psi$					
$\{00\}: \varphi$	$[\geq 12]: \psi$	$[\leq 01]: \varphi$	$\{11\}: \varphi$	$\{02\}: \varphi$	$\{02\}: \varphi$	$[\geq 02]: \varphi$	$\{11\}: \varphi$		
		$[\geq 01]: \psi$	$\{11\}: \psi$	$\{02\}: \psi$	$\{01,11\}: \psi$	$\{11\}: \psi$	$\{01\}: \psi$		
$\{01,11\}: \varphi \rightarrow \psi$		$\{01\}: \varphi \rightarrow \psi$		$[\leq 11]: \varphi \rightarrow \psi$		$[\geq 02]: \varphi \rightarrow \psi$			
$[\geq 02]: \varphi$	$[\geq 12]: \varphi$	$[\geq 01]: \varphi$	$[\geq 02]: \varphi$	$\{22\}: \varphi$	$[\geq 02]: \psi$	$\{01,11\}: \varphi$	$\{01,11\}: \varphi$		
$\{01,11\}: \psi$	$\{01\}: \psi$	$\{00\}: \psi$	$[\leq 11]: \psi$			$\{01,11\}: \psi$	$\{01,11\}: \psi$		
$\{01\}: \neg \varphi$	$\{02\}: \neg \varphi$	$\{22\}: \neg \varphi$	$[\geq 12]: \neg \varphi$	$[\geq 02]: \neg \varphi$	$\{00\}: \neg \varphi$	$\{00\}: \neg \varphi$	$\{00\}: \neg \varphi$		
\perp	\perp	$\{00\}: \varphi$	$\{00\}: \varphi$	$\{00\}: \varphi$	$\{00\}: \varphi$	$\{00\}: \varphi$	$[\geq 02]: \varphi$		
$[\leq 01]: \neg \varphi$	$[\leq 11]: \neg \varphi$	$[\leq 12]: \neg \varphi$	$[\geq 01]: \neg \varphi$	$\{11\}: \neg \varphi$	$\{01,11\}: \neg \varphi$	$\{01,11\}: \neg \varphi$	$\{01,11\}: \neg \varphi$		
$[\geq 02]: \varphi$	$[\geq 01]: \varphi$	$[\geq 01]: \varphi$	$[\leq 11]: \varphi$	$\{01,11\}: \varphi$	$\{01,11\}: \varphi$	$\{01,11\}: \varphi$	$\{01,11\}: \varphi$		
				$[\geq v]: \varphi \wedge \psi$					
For $v \in \{00, 11, 12\}$:				$\frac{[\leq v]: \varphi \wedge \psi}{[\leq v]: \varphi \mid [\leq v]: \psi}$				for $v \in \{01, 02, 12, 22\}$:	$\frac{[\geq v]: \varphi \wedge \psi}{[\geq v]: \varphi \mid [\geq v]: \psi}$
$\{01\}: \varphi \wedge \psi$		$\{11\}: \varphi \wedge \psi$		$\{02\}: \varphi \wedge \psi$					
$\{01\}: \varphi$	$[\geq 01]: \varphi$	$\{11\}: \varphi$	$\{02\}: \varphi$	$\{11\}: \varphi$	$\{11\}: \varphi$	$[\geq 12]: \varphi$	$\{02\}: \varphi$		
$[\geq 01]: \psi$	$\{01\}: \psi$	$\{02\}: \psi$	$\{11\}: \psi$	$\{11\}: \psi$	$[\geq 12]: \psi$	$\{11\}: \psi$	$[\geq 02]: \psi$		
$\{01,11\}: \varphi \wedge \psi$		$[\leq 01]: \varphi \wedge \psi$							
$\{01,11\}: \varphi$	$[\geq 01]: \varphi$	$[\leq 01]: \varphi$	$[\leq 01]: \psi$	$\{11\}: \varphi$	$\{02\}: \varphi$	$\{02\}: \varphi$	$\{11\}: \varphi$		
$[\geq 01]: \psi$	$\{01,11\}: \psi$			$\{02\}: \psi$	$\{11\}: \psi$	$\{11\}: \psi$	$\{11\}: \psi$		
				$[\leq v]: \varphi \vee \psi$					
For $v \in \{00, 01, 11, 12\}$:				$\frac{[\leq v]: \varphi \vee \psi}{[\leq v]: \varphi \mid [\leq v]: \psi}$				for $v \in \{01, 02, 22\}$:	$\frac{[\geq v]: \varphi \vee \psi}{[\geq v]: \varphi \mid [\geq v]: \psi}$
$\{01\}: \varphi \vee \psi$		$\{11\}: \varphi \vee \psi$		$\{02\}: \varphi \vee \psi$					
$[\leq 01]: \varphi$	$\{01\}: \varphi$	$[\leq 11]: \varphi$	$\{11\}: \varphi$	$[\leq 01]: \varphi$	$\{02\}: \varphi$	$\{02\}: \varphi$	$\{02\}: \varphi$		
$\{01\}: \psi$	$[\leq 01]: \psi$	$\{11\}: \psi$	$[\leq 11]: \psi$	$\{02\}: \psi$	$[\leq 01]: \psi$	$\{02\}: \psi$	$\{02\}: \psi$		
$\{01,11\}: \varphi \vee \psi$		$[\geq 12]: \varphi \vee \psi$							
$\{01,11\}: \varphi$	$[\leq 11]: \varphi$	$[\geq 12]: \varphi$	$[\geq 12]: \psi$	$\{11\}: \varphi$	$\{02\}: \varphi$	$\{02\}: \varphi$	$\{11\}: \varphi$		
$[\leq 11]: \psi$	$\{01,11\}: \psi$			$\{02\}: \psi$	$\{11\}: \psi$	$\{11\}: \psi$	$\{11\}: \psi$		

Fig. 3. Expansion rules for HT^2

$\{22\}: \varphi \rightarrow \psi$		$[\geq 11]: \varphi \rightarrow \psi$	$[\leq 11]: \varphi \rightarrow \psi$	$\{00\}: \varphi \rightarrow \psi$
$\{00\}: \varphi$	$\{22\}: \psi$	$\{00\}: \varphi$	$[\geq 11]: \psi$	$[\geq 11]: \varphi$
	$[\leq 11]: \varphi$	$[\geq 11]: \varphi$	$\{22\}: \varphi$	$\{00\}: \psi$
	$[\geq 11]: \psi$	$\{00\}: \neg \varphi$	$\{22\}: \neg \varphi$	
	$[\leq 11]: \neg \varphi$	$[\geq 11]: \neg \varphi$	$\{22\}: \varphi$	$\{00\}: \varphi$
	$[\leq 11]: \varphi$	$[\geq 11]: \varphi$	$\{22\}: \varphi$	$\{00\}: \varphi$
For $v \in \{00, 11, 22\}$:				
$[\leq v]: \varphi \wedge \psi$	$[\geq v]: \varphi \wedge \psi$	$[\leq v]: \varphi \vee \psi$	$[\geq v]: \varphi \vee \psi$	
$[\leq v]: \varphi$	$[\leq v]: \psi$	$[\leq v]: \varphi$	$[\geq v]: \varphi$	$[\geq v]: \psi$
	$[\geq v]: \psi$	$[\leq v]: \psi$	$[\geq v]: \psi$	$[\geq v]: \psi$

Fig. 4. Expansion rules for total models of HT^2 , ie. for HT^*

The usual notions of *closed* and *terminated* tableaux in conjunction with a specific *initial tableau* allow us to treat different problems such as satisfiability, validity, equivalence, partial equilibrium property, etc. In the following definition we introduce the concept of *closed tableau* in order to characterise validity in HT^2 .

Definition 6. Let φ be a formula in HT^2 :

1. The initial tableau to check the validity of φ is: $T_0 = [\leq 12]: \varphi$
2. If T is a tableau and T' is the tree obtained from T by applying one of the expansion rules in Figure 3, then T' is a tableau for φ .
3. A branch B in a tableau T is called closed if one of the following conditions hold: (i) it contains the constant \perp ; (ii) it contains signed literals, $s_1: p, \dots, s_n: p$, such that $\bigcap_{i=1}^n S_i = \emptyset$. A tableau T is called closed if every branch is closed.

Intuitively, with the initial tableau $[\leq 12]: \varphi$ we ask if it is possible to find an assignment that evaluates φ in $[\leq 12]$, in other words a countermodel. The expansion rules propagate this question through the subformulas to the atoms. If every non-atomic formula has been expanded but we have a non-closed branch, then we can construct a countermodel of the initial formula assigning a value to every variable chosen from its sign in the branch. This sketches the proof of the following result.

Theorem 6 (Soundness and completeness of the tableaux system). The formula φ is valid in HT^2 if and only if there exists a closed tableau for it.

8.1 Partial equilibrium models

Tableaux systems can also be used to study additional properties and relations [25, 27]. In this section we define a system based on auxiliary tableaux in order to generate the partial equilibrium models of a theory. We proceed in two phases. First, we generate the total models of a theory by means of a tableau system in which we search for a

terminated tableau. Then, for every total model, an auxiliary tableau is constructed to check whether the model in question is in partial equilibrium.

The total assignments evaluate formulas in $\{00, 11, 22\}$ and thus we only need to work with the following system of signs: $[\leq 11] = \{00, 11\}$, $[\leq 00] = \{00\}$, $[\geq 11] = \{11, 22\}$, $[\geq 11] = \{22\}$.

Definition 7. Let $\Pi = \{\varphi_1, \dots, \varphi_n\}$ be a theory in HT^2 :

1. The initial tableau to generate total models is a single branch tree containing the following signed formulas: $\{22\}:\varphi_1, \dots, \{22\}:\varphi_n$.
2. If T is a tableau and T' is the tree obtained from T by applying one of the expansion rules in Figure 4, then T' is tableau for φ . As usual in tableaux systems for propositional logics, if a formula can be used to expand the tableau, then the tableau is expanded in every branch below the formula using the corresponding rule, the formula is marked and it is no longer used.
3. A branch in a tableau T is called closed if the signed literals for a variable p , $s_1:p, \dots, s_m:p$, verify $\bigcap_{i=1}^m S_i = \emptyset$. It is called open otherwise.
4. A branch in a tableau T is called finished if it doesn't contain non-marked formulas.
5. A tableau T is called closed if every branch is closed, and it is terminated if every branch is either closed or finished.

In this case the tableau begins with formulas signed with 22, since we are looking for models. The expansion rules guarantee the construction of all possible models in such a way that when all formulas have been expanded (marked), all the models can be determined on the basis of open branches.

Theorem 7. Let T be a non-closed terminated tableau for Π , and $\{s_1:p_1, \dots, s_n:p_n\}$ the set of signed literals in an open branch. Then every assignment V verifying $V(p_i) \in S_i$, for all i , is a total model of φ . Moreover, all the total models of Π are generated from T in this way.

Example 3. (Taken from [7]) Figure 5 shows the tableau for the theory $\Pi = \{\neg p \rightarrow q \vee r, p \vee r\}$. The tableau is finished and allows us to construct the set of total models of Π also shown in the Figure.

Auxiliary tableau to check the partial equilibrium property A total model is in partial equilibrium if there is no other strictly smaller model of the theory, under the partial ordering \triangleleft . In terms of the many-valued semantics, this ordering is defined between assignments based on the following relations between truth-values: $01 \triangleleft 11$, $02 \triangleleft 12 \triangleleft 22$. So, a procedure to generate partial equilibrium models is check every total model in order to determine if there exists a model less than it. To do that, we are going to use another tableau system using the expansion rules in Figure 3 but a different notion of closed tableau over an specific initial tableau.

Definition 8. Let φ be a formula in HT^2 and V a total model of φ .

1. The initial tableau to check the partial equilibrium property of V for φ is a single branch tree containing the following signed formulas: $\{22\}:\varphi$, $\{00\}:p$ for every p such that $V(p) = 00$, $\{01, 11\}:p$ for every p such that $V(p) = 11$, and $\{02, 12, 22\}:p$ for every p such that $V(p) = 22$.

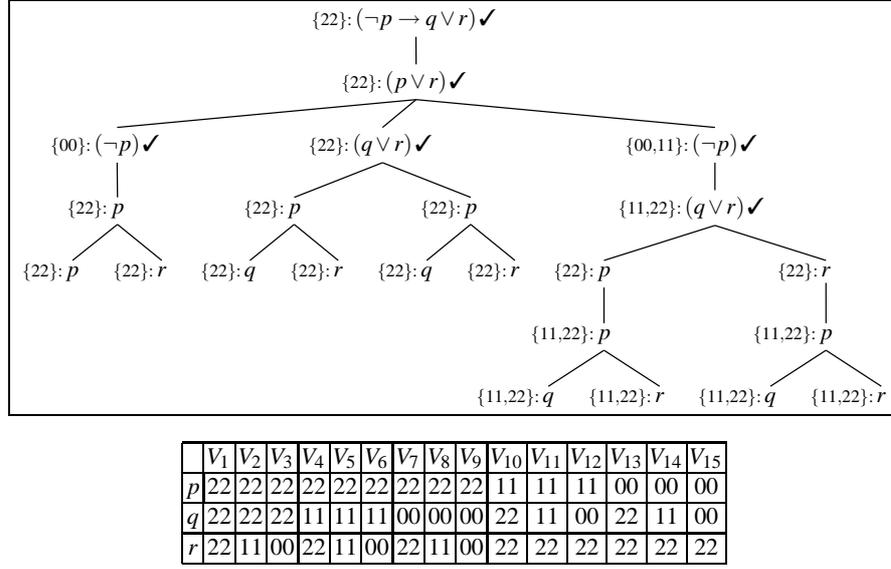


Fig. 5. Main tableau for example 3 and the resulting total models.

2. If T is a tableau and T' is the tree obtained from T applying one of the expansion rules in Figure 3, then T' is ϕ .
3. A branch B in a tableau T is called V -closed if one of the following conditions holds: (i) it contains the constant \perp ; (ii) it contains signed literals, $s_1:p, \dots, s_n:p$, such that $\bigcap_{i=1}^n S_i = \emptyset$; (iii) all the formulas in the branch have been expanded and, for every variable p , it contains signed literals, $s_1:p, \dots, s_n:p$, such that $\bigcap_{i=1}^n S_i = \{V(p)\}$.
4. A tableau T is called V -closed if every branch is V -closed.

Adding literals of the form $\{01,11\}:p$, $\{02,12,22\}:p$ or $\{00\}:p$, depending on $V(p)$, to the initial tableau, forces the tableau to generate only models less than V . Nevertheless, we know that one model will always be found, V itself, and therefore we must include one more conditions on closure: a branch closes if it generates only the model V .

Theorem 8. Let V be a total model of ϕ . V is a partial equilibrium model of Π if and only if there exists a V -closed tableau for ϕ .

In Figure 6 we show that, for the previous example, the model V_9 is a partial equilibrium model; observe that the leftmost branch closes because V is the only model generated, while all other branches close due to inconsistencies provoked by the three signed literals added to the initial tableau. In the second tableau in the same figure we check that the model V_{12} is not a partial equilibrium model.

We have chosen this presentation of the auxiliary tableau systems for the sake of simplicity, but sacrificing efficiency. Following the techniques of [25, 26] we can improve the methods of this section to obtain more efficient systems.

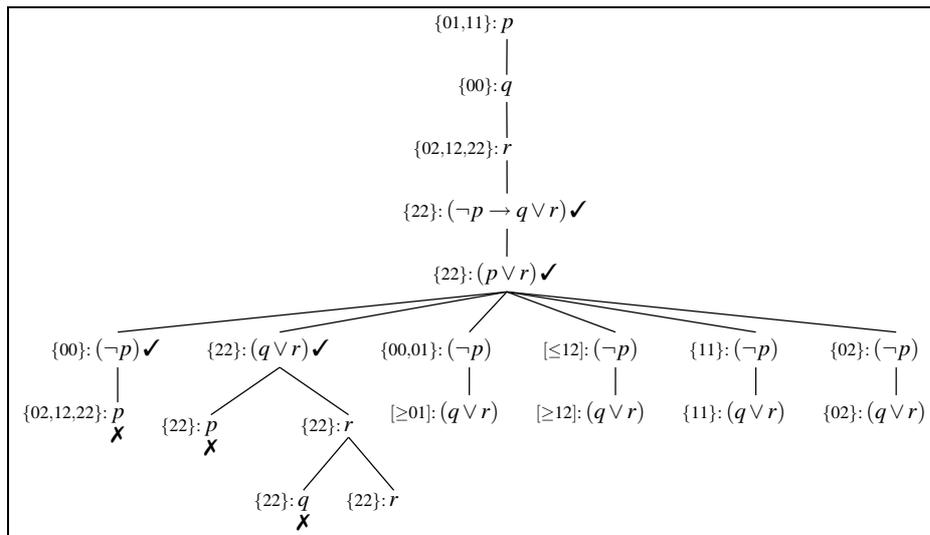
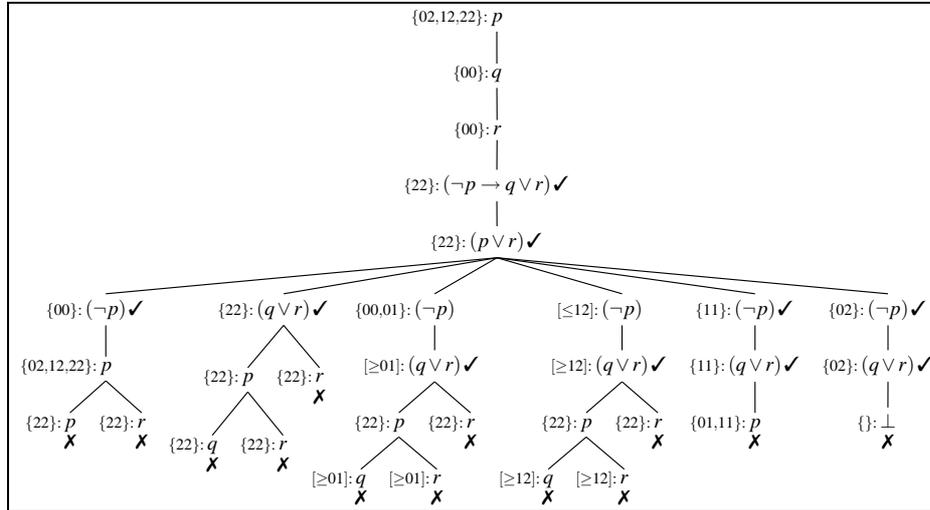


Fig. 6. Auxiliary tableaux for two total models of example 3

9 Translating partiality by atoms replication

A promising approach to implementing p-stable models for disjunctive programs has been developed by Janhunen *et al* [16]. 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 in such a way that PEL can be reduced to ordinary equilibrium logic. Furthermore the method provides an encoding of the underlying logics, of HT^2 into HT .

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:

$$\begin{array}{ll} [\varphi \rightarrow \psi] := ([\varphi] \rightarrow [\psi]) \wedge [\varphi \rightarrow \psi]' & [\varphi \rightarrow \psi]' := [\varphi]' \rightarrow [\psi]' \\ [\neg\varphi] := \neg[\varphi]' & [\neg\varphi]' := \neg[\varphi] \\ [\varphi \oplus \psi] := [\varphi] \oplus [\psi] & [\varphi \oplus \psi]' := [\varphi]' \oplus [\psi]' \\ [p] := p & [p]' := p' \\ [\varepsilon] := \varepsilon & [\varepsilon]' := \varepsilon \end{array}$$

where $\oplus \in \{\wedge, \vee\}$ and $\varepsilon \in \{\top, \perp\}$.

Example 4. The translation $\varphi = \neg(a \rightarrow \neg b) \rightarrow c$ consists of the formulas $a \rightarrow a'$, $b \rightarrow b'$, $c \rightarrow c'$ and $\neg(a' \rightarrow \neg b') \rightarrow c) \wedge (\neg((a \rightarrow \neg b') \wedge (a' \rightarrow \neg b)) \rightarrow c'$.

It is quite easy to see that for any disjunctive rule r like (2), its translation $[r]$ has the form $(a_1 \wedge \dots \wedge a_m \wedge \neg b'_1 \wedge \dots \wedge \neg b'_n \rightarrow c_1 \vee \dots \vee c_k) \wedge (a'_1 \wedge \dots \wedge a'_m \wedge \neg b_1 \wedge \dots \wedge \neg b_n \rightarrow c'_1 \vee \dots \vee c'_k)$ so that $Tr(\Pi)$ amounts to Janhunen *et al*'s transformation [16] when Π is a disjunctive logic program.

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

Proposition 17. *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 $Tr(\Gamma)$.*

Proof. As \mathcal{M}_1 is an HT^2 interpretation, $H \subseteq H'$ and $T \subseteq T'$ by construction, and thus $\mathcal{M}_2 \models p' \leftarrow p$ for any p in the signature of Γ . Analogously, if we take any HT interpretation \mathcal{M}_2 for the extended signature satisfying rules $p \rightarrow p'$, then \mathcal{M}_1 is a correctly constructed HT^2 interpretation. So, it just remains to show that $\mathcal{M}_1 \models \varphi$ iff $\mathcal{M}_2 \models [\varphi]$, for any $\varphi \in \Gamma$. In fact, we will show that, for any $w \in \{h, t\}$, both:

$$(i) \mathcal{M}_1, w \models \varphi \text{ iff } \mathcal{M}_2, w \models [\varphi] \quad \text{and} \quad (ii) \mathcal{M}_1, w' \models \varphi \text{ iff } \mathcal{M}_2, w \models [\varphi]'$$

We proceed by structural induction.

1. For \perp, \top it is trivial. For an atom p in the original alphabet, $\mathcal{M}_1, h \models p \Leftrightarrow p \in H \Leftrightarrow p \in H \cup [H']' \Leftrightarrow \mathcal{M}_2, h \models p$ and the same holds when replacing h/H by t/T . To prove (ii) we have: $\mathcal{M}_1, h' \models p \Leftrightarrow p \in H' \Leftrightarrow p' \in [H']' \Leftrightarrow p' \in H \cup [H']' \Leftrightarrow \mathcal{M}_2, h \models p'$ and the same when replacing h/H by t/T .

2. For $\varphi \wedge \psi$ and $\varphi \vee \psi$: direct application of the induction hypothesis on φ, ψ .
3. For $\neg\varphi$, to prove (i) we have $[\neg\varphi] = \neg[\varphi]'$. Then, $\mathcal{M}_2, w \models \neg[\varphi]' \Leftrightarrow \mathcal{M}_2, t \not\models [\varphi]'$ \Leftrightarrow (by induction (ii)) $\mathcal{M}_1, t' \not\models \varphi \Leftrightarrow \mathcal{M}_1, w \models \neg\varphi$. The proof for (ii) is similar: $\mathcal{M}_2, w \models \neg[\varphi] \Leftrightarrow \mathcal{M}_2, t \not\models [\varphi] \Leftrightarrow$ (by induction (i)) $\mathcal{M}_1, t \not\models \varphi \Leftrightarrow \mathcal{M}_1, w' \models \neg\varphi$.
4. Finally, for $\varphi \rightarrow \psi$ and condition (i): $\mathcal{M}_2, h \models [\varphi \rightarrow \psi] \Leftrightarrow \mathcal{M}_2, h \models ([\varphi] \rightarrow [\psi]) \wedge [\varphi \rightarrow \psi]'$ \Leftrightarrow ($\mathcal{M}_2, w \not\models [\varphi]$ or $\mathcal{M}_2, w \models [\psi]$) for $w \in \{h, t\}$ and $\mathcal{M}_2, h \models [\varphi \rightarrow \psi]'$ \Leftrightarrow (by induction (i) and (ii)) ($\mathcal{M}_1, w \not\models \varphi$ or $\mathcal{M}_1, w \models \psi$) for $w \in \{h, t\}$ and $\mathcal{M}_1, h' \models \varphi \rightarrow \psi \Leftrightarrow \mathcal{M}_1, h \models \varphi \rightarrow \psi$. For world $w = t$ it suffices to replace above all occurrences of h and h' respectively by t and t' . As for condition (ii), we have: $\mathcal{M}_2, h \models [\varphi \rightarrow \psi]'$ $\Leftrightarrow \mathcal{M}_2, h \models [\varphi]' \rightarrow [\psi]'$ \Leftrightarrow ($\mathcal{M}_2, w \not\models [\varphi]'$ or $\mathcal{M}_2, w \models [\psi]'$) for $w \in \{h, t\} \Leftrightarrow$ ($\mathcal{M}_1, w' \not\models \varphi$ or $\mathcal{M}_1, w' \models \psi$) for $w \in \{h, t\} \Leftrightarrow \mathcal{M}_1, h' \models \varphi \rightarrow \psi$. Again, the proof for $w = t$ is obtained by replacing h by t and h' by t' . \square

Proposition 18. *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 $Tr(\Gamma)$.*

Proof. By Proposition 17 it suffices to prove that model minimisations yield the same effect in both cases. In fact, we show next that there exists a one to one correspondence between both model ordering relations:

$$\begin{aligned}
\langle (H_1, H_1'), (T, T') \rangle \leq \langle (H_2, H_2'), (T, T') \rangle &\Leftrightarrow H_1 \subseteq H_2 \text{ and } H_1' \subseteq H_2' \\
&\Leftrightarrow H_1 \subseteq H_2 \text{ and } [H_1] \subseteq [H_2]' \Leftrightarrow H_1 \cup [H_1]' \subseteq H_2 \cup [H_2] \\
&\Leftrightarrow \langle H_1 \cup [H_1]', T \cup [T']' \rangle \leq \langle H_2 \cup [H_2]', T \cup [T']' \rangle \quad \square
\end{aligned}$$

10 A splitting theorem for PEL

The previous tableau calculus offers a general method for satisfiability testing in HT^2 and PEL, given any arbitrary theory. When we restrict the syntax to (some class of) logic programs, we usually expect, however, that simpler computation methods can be applied. Consider for instance the case of disjunctive logic programs. As shown in [7], PEL also coincides with p-stable models for this syntactic class. Maintaining the same minimisation criterion, we may easily find that a disjunctive program yields several well-founded models (even no well-founded model at all), and the typical incremental algorithm for computing WFS for normal programs is not applicable. However, it is still possible to apply a form of incremental reasoning if we can divide or “split” the program into blocks without cyclic dependences among them. As an example, consider the simple program $\Pi_0 = \{p \vee q\}$ which yields two p-stable models (also well-founded), making p true and q false in one case, and vice versa. Now, assume we have the enlarged program $\Pi_1 = \Pi_0 \cup \{\neg r \wedge p \rightarrow r, q \wedge \neg p \rightarrow s, \neg s \rightarrow s\}$. It seems natural to use this second set of formulas to compute atoms r and s , once p and q are still fixed by the rule in Π_0 . This technique is called “splitting” and was first introduced in [18] for the case of stable models. We now establish a similar result for PEL in the more general syntactic case where theories are sets of implications.

Given a pair $\mathbf{T} = (T, T')$ and a set of atoms U , we denote $\mathbf{T}|_U = (T \cap U, T' \cap U)$. We apply a similar notation for theories too. If Π is some theory in language $\mathcal{L}(V)$, and $U \subseteq V$, then we write $\Pi|_U$ to stand for set of formulas $\Pi \cap \mathcal{L}(U)$. We respectively call *bottom* and *top* the subtheories $\Pi|_U$ and $\Pi \setminus \Pi|_U$.

Definition 9 (Splitting set). Given a set of implications Π on signature V , a subset $U \subseteq V$ is called a splitting set for Π if for all $(\varphi \rightarrow \psi) \in \Pi \setminus \Pi|_U$, $\psi \in \mathcal{L}(V \setminus U)$.

Theorem 9 (Splitting theorem). Let Π be a set of implications, U a splitting set for Π and \mathbf{T} a pair (T, T') of sets of atoms $T \subseteq T'$. Then $\mathbf{T} \approx \Pi$ iff both (i) $\mathbf{T}|_U \approx \Pi|_U$ and (ii) $\mathbf{T} \approx \Pi'$, where

$$\Pi' := (\Pi \setminus \Pi|_U) \cup \quad (6)$$

$$\cup (T \cap U) \quad (7)$$

$$\cup \{\neg p \mid p \in U \setminus T'\} \quad (8)$$

$$\cup \{p \leftrightarrow \mathbf{u} \mid p \in (T' \setminus T) \cap U\}$$

Proof. “ \Rightarrow ” (i). Assume $\mathbf{T} \approx \Pi$ but $\mathbf{T}|_U \not\approx \Pi|_U$. As $\langle \mathbf{T}, \mathbf{T} \rangle \models \Pi$, in particular, $\langle \mathbf{T}, \mathbf{T} \rangle \models \Pi|_U$, but then, clearly $\langle \mathbf{T}|_U, \mathbf{T}|_U \rangle \models \Pi|_U$. Thus, there must exist some $\mathbf{H} = (H, H')$, $\mathbf{H} < \mathbf{T}|_U$ such that $\langle \mathbf{H}, \mathbf{T}|_U \rangle \models \Pi|_U$. Since $\Pi|_U$ exclusively refers to atoms in U , this means that any interpretation extending $H, H', T \cap U$ and $T' \cap U$ with atoms not in U will still be a model of that program. In particular, it is easy to see that for $\langle \mathbf{H}_2, \mathbf{T} \rangle$ with $\mathbf{H}_2 = \langle H \cup (T \setminus U), H' \cup (T' \setminus U) \rangle$: (1) it is a well constructed interpretation; (2) it satisfies $\Pi|_U$; (3) it is strictly lower than $\langle \mathbf{T}, \mathbf{T} \rangle$; and (4) for any world w and any formula $\alpha \in \mathcal{L}(V \setminus U)$, $\langle \mathbf{T}, \mathbf{T} \rangle, w \models \alpha$ iff $\langle \mathbf{H}_2, \mathbf{T} \rangle, w \models \alpha$. As $\mathbf{T} \approx \Pi$, (2) and (3) mean there must exist some implication $\varphi \rightarrow \psi \in \Pi \setminus \Pi|_U$ such that $\langle \mathbf{H}_2, \mathbf{T} \rangle \not\models \varphi \rightarrow \psi$. This means $\langle \mathbf{H}_2, \mathbf{T} \rangle, w \models \varphi$ and $\langle \mathbf{H}_2, \mathbf{T} \rangle, w \not\models \psi$ for some world w , but as $\langle \mathbf{T}, \mathbf{T} \rangle \models \varphi \rightarrow \psi$ this reduces the possibilities to $w \in \{h, h'\}$. Assume $w = h$; then $\langle \mathbf{H}_2, \mathbf{T} \rangle, h \models \varphi$ implying, by the hereditary property, that $\langle \mathbf{H}_2, \mathbf{T} \rangle, t \models \varphi$ which implies $\langle \mathbf{T}, \mathbf{T} \rangle, h \models \varphi$. This, together with $\langle \mathbf{T}, \mathbf{T} \rangle \models \varphi \rightarrow \psi$ entails $\langle \mathbf{T}, \mathbf{T} \rangle, h \models \psi$, but as ψ does not refer to atoms in U , by (4) we obtain $\langle \mathbf{H}_2, \mathbf{T} \rangle, h \models \psi$ reaching a contradiction. The proof for $w = h'$ is completely analogous, replacing the above h and t by their primed versions.

“ \Rightarrow ” (ii). Trivially $\langle \mathbf{T}, \mathbf{T} \rangle \models \Pi'$. Assume $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi'$ with $\mathbf{H} < \mathbf{T}$. It is easy to see that Π' fixes the interpretation of atoms in U , and so, $\mathbf{H}|_U = \mathbf{T}|_U$ which implies $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi|_U$. On the other hand, as $\langle \mathbf{H}, \mathbf{T} \rangle$ is model of Π' , we also have $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi \setminus \Pi|_U$, so that we get $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi$ contradicting $\mathbf{T} \approx \Pi$.

“ \Leftarrow ”. From (i) and (ii) it is clear that $\langle \mathbf{T}, \mathbf{T} \rangle \models \Pi$. Assume $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi$ with $\mathbf{H} < \mathbf{T}$. As $\Pi|_U$ only refers to atoms in U , $\langle \mathbf{H}|_U, \mathbf{T}|_U \rangle \models \Pi|_U$, but this, together with (i), leaves $\mathbf{H}|_U = \mathbf{T}|_U$ as the only possibility. Now, as $\langle \mathbf{T}, \mathbf{T} \rangle \models (6) \cup (7) \cup (8)$ and these formulas exclusively refer to signature U , we obtain $\langle \mathbf{H}, \mathbf{T} \rangle \models (6) \cup (7) \cup (8)$. Finally, as $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi$ we also have $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi \setminus \Pi|_U$ and so $\langle \mathbf{H}, \mathbf{T} \rangle \models \Pi'$ contradicting $\mathbf{T} \approx \Pi'$. \square

The previous theorem is completed with the following result. Let $\Pi[\varphi/p]$ denote the replacement in theory Π of any occurrence of atom p by the formula φ .

Theorem 10 (Replacement theorem). For any theory Π and any model \mathcal{M} :

- (i) $\mathcal{M} \models \Pi \cup \{p\}$ iff $\mathcal{M} \models \Pi[\top/p] \cup \{p\}$
- (ii) $\mathcal{M} \models \Pi \cup \{\neg p\}$ iff $\mathcal{M} \models \Pi[\perp/p] \cup \{\neg p\}$
- (iii) $\mathcal{M} \models \Pi \cup \{p \leftrightarrow \mathbf{u}\}$ iff $\mathcal{M} \models \Pi[\mathbf{u}/p] \cup \{p \leftrightarrow \mathbf{u}\}$

Proof. For (i) the satisfaction of p at h means that p will hold at any world and so it can be replaced by \top . For (ii), the satisfaction of $\neg p$ at h means p will be false at t' and so

false at any world, so it can be replaced by \perp . Finally, it is easy to see that $\mathcal{M} \models p \leftrightarrow \mathbf{u}$ iff p is true at h', t' but false at h, t , which coincides with the valuation of \mathbf{u} . \square

Returning to the example program Π_1 , $U = \{p, q\}$ is a splitting set dividing Π_1 into the bottom Π_0 and the top $\Pi_1 \setminus \Pi_0$. As we saw, Π_0 has two p-equilibrium models: $\mathbf{T}_1 = (\{p\}, \{p\})$ and $\mathbf{T}_2 = (\{q\}, \{q\})$. Now, fixing \mathbf{T}_1 , we consider the theory $\Pi' = \Pi_1 \setminus \Pi_0 \cup \{p\} \cup \{\neg q\}$ which, by the replacement theorem, is equivalent to $\{\neg r \wedge \top \rightarrow r, \perp \wedge \neg \top \rightarrow s, \neg s \rightarrow s, p, \neg q\}$. After some trivial simplifications, this amounts to $\{\neg r \rightarrow r, \neg s \rightarrow s, p, \neg q\}$ whose unique p-equilibrium model is defined by $\mathbf{T}_3 = (\{p\}, \{p, r, s\})$. Following similar steps, when fixing \mathbf{T}_2 we finally get the program $\{s, \neg s \rightarrow s, q, \neg p\}$ with the only p-equilibrium model $\mathbf{T}_4 = (\{q, s\}, \{q, s\})$.

11 Concluding remarks

We have seen that partial equilibrium logic (PEL) provides a foundation for and generalisation of the p-stable semantics of logic programs. We would argue that it is therefore also a suitable framework for studying and extending the well-founded semantics of programs. In this paper we have focused on logical properties of PEL and its underlying logics HT^2 and HT^* as well as on various methods of computing partial equilibrium models. In this direction we have examined some general methods, such as tableaux calculi and reduction techniques, that apply to arbitrary propositions, as well as specific techniques, including splitting, that apply to logic programs. Further optimisation of these computational methods is a topic for future work.

Since WFS is already successfully implemented and applied in practice, we do not envisage our work as having an immediate impact on mature systems of logic programming that already use WFS or p-stable models. Our motivation is rather to provide an appropriate instrument for extending the basic syntax of logic programs and to understand better the logical foundations of the p-stable approach. Nevertheless, these logical foundations may also have an indirect influence on programming with WFS, even where the language of normal programs is concerned. A possible example is provided by the topic of strong equivalence, discussed briefly in Section 2. In answer set programming the study of strong equivalence and related concepts is contributing to efforts devoted to program optimisation and simplification. No analogous technique seems to have been available so far in the case of programming with WFS and its variants. This avenue is one that we think worth exploring in the future.

Another area of ongoing research concerns the addition of a second, strong or explicit negation to the syntax of HT^2 . Several approaches were recently explored in a preliminary work [8], where relations to the semantics WSFX of [28] and other variants of well-founded semantics with a second negation operator were investigated.

References

1. S. Brass and J. Dix. A disjunctive semantics based on unfolding and bottom-up evaluation. In *Proceedings of IFIP 1994 Congress, Workshop FG2: Disjunctive Logic Programming and Disjunctive Databases*, pp. 83–91. Springer 1994.

2. S. Brass and J. Dix. Characterizations of the Disjunctive Stable Semantics by Partial Evaluation. In *Journal of Logic Programming* 30 (1997), 207–228.
3. S. Brass and J. Dix. Characterizations of the Disjunctive Well-founded Semantics: Confluent Calculi and Iterated GCWA. In *Journal of Automated Reasoning* 20 (1998), 143–165.
4. S. Brass and J. Dix. Semantics of (Disjunctive) Logic Programs Based on Partial Evaluation. In *Journal of Logic Programming* 40 (1999), 1–46.
5. S. Brass, J. Dix, I. Niemelä and T. Przymusiński. On the equivalence of the Static and Disjunctive Well-Founded Semantics and its computation. In *Theoretical Computer Science*, 258(1–2):523–553, 2001.
6. P. Cabalar, S. Odintsov and D. Pearce. Logical Foundations of Well-Founded Semantics. In *Proceedings of KR 2006*, pp. 25–35. AAAI Press, 2006.
7. P. Cabalar, S. Odintsov, D. Pearce and A. Valverde. Analysing and Extending Well-Founded and Partial Stable Semantics using Partial Equilibrium Logic. In S. Etalle and M. Truszczynski (eds.), *Proceedings of ICLP 2006*, LNCS 4079, pp. 346–360. Springer, 2006.
8. P. Cabalar, S. Odintsov and D. Pearce. Strong Negation in Well-Founded and Partial Stable Semantics for Logic Programs. In J. Simão Sichman, H. Coelho and S. Oliveira Rezende (eds.), *Proceedings of IBERAMIA-SBIA 2006*, LNCS 4140, pp. 592–601. Springer, 2006.
9. J. Dix. A Classification-Theory of Semantics of Normal Logic Programs: I. Strong Properties. In *Fundamenta Informaticae*, 22(3):227–255, 1995.
10. J. Dix. A Classification-Theory of Semantics of Normal Logic Programs: II. Weak Properties. In *Fundamenta Informaticae*, 22(3):257–288, 1995.
11. T. Eiter, M. Fink, H. Tompits and S. Woltran. Simplifying Logic Programs under Uniform and Strong Equivalence. In V. Lifschitz and I. Niemelä (eds.), *Proceedings of LPNMR 2004*, LNAI 2923, pp. 87–99. Springer, 2004.
12. T. Eiter, N. Leone and D. Saccà. On the Partial Semantics for Disjunctive Deductive Databases. In *Annals of Mathematics and Artificial Intelligence* 17:59–96, 1997.
13. T. Eiter, N. Leone and D. Saccà. Expressive Power and Complexity of Partial Models for Disjunctive Deductive Databases. In *Theoretical Computer Science* 206:181–218, 1998.
14. S. T. Erdogan and V. Lifschitz. Definitions in Answer Set Programming. In V. Lifschitz and I. Niemelä (eds.), *Proceedings of ICLP 04*, LNAI 2923, pp. 114–126. Springer, 2004.
15. R. Hähnle. *Automated Deduction in Multiple-Valued Logics*. Volume 10 of *International Series of Monographs on Computer Science*. Oxford University Press, 1994.
16. T. Janhunen, I. Niemelä, D. Seipel, P. Simons and J.-H. You. Unfolding partiality and disjunctions in stable model semantics. In *ACM Transactions on Computational Logic* 7(1):1–37, 2006.
17. S. Kraus, D. Lehmann and M. Magidor. Nonmonotonic Reasoning, Preferential Models and Cumulative Logics. In *Artificial Intelligence* 44:167–207, 1990.
18. V. Lifschitz and H. Turner. Splitting a Logic Program. In P. van Hentenryck (ed.), *Proceedings of ICLP 1994*, pp. 23–37. MIT Press, 1994.
19. V. Lifschitz, D. Pearce and A. Valverde. Strongly equivalent logic programs. In *ACM Transactions on Computational Logic*, 2(4):526–541, 2001.
20. V. Lifschitz, L.R. Tang and H. Turner. Nested expressions in logic programs. In *Annals of Mathematics and Artificial Intelligence*, 25(3–4):369–389, 1999.
21. D. Makinson. General Theory of Cumulative Inference. In M. Reinfrank *et al* (eds.), *Non-Monotonic Reasoning*, LNAI 346, pp. 1–18. Springer, 1989.
22. D. Makinson. General Patterns in Nonmonotonic Reasoning. In D. Gabbay, C.J. Hogger and J.A. Robinson (eds.), *Handbook of Logic in Artificial Intelligence*. Clarendon Press, 1994.
23. M. Osorio, J. Navarro and J. Arrazola. Equivalence in Answer Set Programming. In A. Pettorossi (ed.), *Proceedings of LOPSTR 2001*, LNCS 2372, pp. 57–75. Springer, 2001.
24. D. Pearce. Equilibrium Logic. In *Annals of Mathematics and Artificial Intelligence*, 47:3–41. 2006.

25. D. Pearce, I.P. de Guzmán and A. Valverde. A tableau calculus for equilibrium entailment. In R. Dyckhoff (ed.), *Proceedings of TABLEAUX 2000*, LNAI 1847, pp. 352–367. Springer, 2000.
26. D. Pearce, I.P. de Guzmán and A. Valverde. Computing equilibrium models using signed formulas. In *Proceedings of CL2000*, LNCS 1861, pages 688–703. Springer, 2000.
27. D. Pearce and A. Valverde. Uniform equivalence for equilibrium logic and logic programs. In V. Lifschitz and I. Niemelä (eds.), *Proceedings of LPNMR 2004*, LNAI 2923, pp. 194–206. Springer, 2004.
28. L. M. Pereira and J. J. Alferes, Well Founded Semantics for Logic Programs with Explicit Negation, In B. Neumann (ed.), *European Conference on Artificial Intelligence*, pp.102-106. John Wiley and Sons. 1992.
29. T. Przymusiński. Stable semantics for disjunctive programs. In *New Generation Computing* 9:401-424, 1991.
30. T. Przymusiński. Well-founded and stationary models of logic programs. In *Annals of Mathematics and Artificial Intelligence* 12:141–187, 1994.
31. R. Routley and V. Routley. The Semantics of First Degree Entailment. In *Noûs* 6:335–359, 1972.
32. C. Ruiz and J. Minker. Computing Stable and Partial Stable Models of Extended Disjunctive Logic Programs. In *Nonmonotonic Extensions of Logic Programming*, LNCS 927, pp. 205–229. Springer, 1995.
33. D. Seipel, J. Minker and C. Ruiz. A Characterization of the Partial Stable Models for Disjunctive Deductive Databases. In *Int. Logic Programming Symp.*, MIT Press, 1997, 245-259.
34. D. Vakarelov. Notes on constructive logic with strong negation. In *Studia Logica*, 36: 89-107, 1977.
35. A. van Gelder, K.A. Ross and J.S. Schlipf. Unfounded sets and well-founded semantics for general logic programs. In *Journal of ACM*, 38(3):620–650, 1991
36. K. Wang and L. Zhou. Comparisons and computation of well-founded semantics for disjunctive logic programs. In *ACM Transactions on Computational Logic*, 6(2): 295–327, 2005.