Revisiting Explicit Negation in Answer Set Programming

FELICIDAD AGUADO[1], PEDRO CABALAR[1], JORGE FANDINNO[2]

DAVID PEARCE[3], GILBERTO PÉREZ[1], CONCEPCIÓN VIDAL[1]

[1] *Universidade da Coruña,*
(University of Corunna), SPAIN
(*e-mail:* {aguado,cabalar,gperez,eicovima}@udc.es)

[2] *IRIT, University of Toulouse, CNRS, FRANCE*
(*e-mail:* jorge.fandinno@irit.fr)

[3] *Universidad Politécnica de Madrid, SPAIN*
(*e-mail:* david.pearce@upm.es)

submitted 1 January 2003; revised 1 January 2003; accepted 1 January 2003

Abstract

A common feature in Answer Set Programming is the use of a second negation, stronger than default negation and sometimes called explicit, strong or classical negation. This explicit negation is normally used in front of atoms, rather than allowing its use as a regular operator. In this paper we consider the arbitrary combination of explicit negation with nested expressions, as those defined by Lifschitz, Tang and Turner. We extend the concept of reduct for this new syntax and then prove that it can be captured by an extension of Equilibrium Logic with this second negation. We study some properties of this variant and compare to the already known combination of Equilibrium Logic with Nelson's strong negation.

KEYWORDS:

1 Introduction

Although the introduction of *stable models* (Gelfond and Lifschitz 1988) in logic programming was motivated by the search of a suitable semantics for default negation, their early application to knowledge representation revealed the need of a second negation to represent explicit falsity. This second negation was already proposed in (Gelfond and Lifschitz 1991) under the name of *classical negation*, an operator only applicable on atoms that, when present in the syntax, led to a change in the name of stable models to become *answer sets*. Classical negation soon became common in applications for commonsense reasoning and action theories (Gelfond and Lifschitz 1993) and was also extrapolated to the Well-Founded Semantics (Pereira and Alferes 1992) under the name of *explicit negation*. Later on, it was incorporated to the paradigm of *Answer Set Programming* (Niemelä 1999; Marek and Truszczyński 1999) (ASP), being nowadays present in the input language of most ASP solvers.

To understand the difference for knowledge representation between default negation (in this paper, written as $\neg$) and explicit negation (represented as $\sim$), a typical example is to distinguish the rule $\neg train \rightarrow cross$, that captures the criterion "you can cross if you have no information on a train coming," from the (safier) encoding $\sim train \rightarrow cross$ that means "you can cross if you have evidence that no train is coming." In ASP, this explicit negation can only be used in front of atoms[1] so it is not seen as a real connective. In an attempt of providing more flexibility to logic program connectives, Lifschitz et al. (1999) introduced programs with *nested expressions* where conjunction, disjunction and default negation could be arbitrarily nested both in the heads and bodies of rules, but classical negation was still restricted to an application on atoms. To see an example, suppose that a given moment, three trains should be crossing, and we have an alarm that fires if one of them is known to be missing. Using nested expressions, we can rewrite the program:

$$\begin{array}{rcl} \sim train_1 & \rightarrow & alarm \\ \sim train_2 & \rightarrow & alarm \\ \sim train_3 & \rightarrow & alarm \end{array}$$

as a single rule with a disjunction in the body:

$$\sim train_1 \vee \sim train_2 \vee \sim train_3 \quad \rightarrow \quad alarm$$

but we cannot further apply De Morgan to rewrite the rule above as:

$$\sim(train_1 \wedge train_2 \wedge train_3) \quad \rightarrow \quad alarm$$

It is easy to imagine that providing a semantics for this kind of expressions would be interesting if we plan to jump from the propositional case to programs with variables and aggregates (where, for instance, the number of trains is some arbitrary value $n \geq 0$).

An important breakthrough that meant a purely logical treatment, was the characterisation of stable models in terms of *Equilibrium Logic* proposed by Pearce (1997). This non-monotonic formalism is defined in terms of a models selection criterion on top of the (monotonic) intermediate logic of *Here-and-There* (HT) (Heyting 1930) and captures default negation $\neg\varphi$ as a derived operator in terms of implication $\varphi \rightarrow \bot$, as usual in intuitionistic logic. The definition of Equilibrium Logic also included a second, constructive negation '$\sim$' corresponding to Nelson's *strong negation* (Nelson 1949) for intermediate logics. In the case of HT, this extension yields a five-valued logic called $\mathcal{N}_5$ where, although '$\sim$' can now be nested as the rest of connectives, there exists a reduction for shifting it in front of atoms, obtaining a *negative normal form* (NNF). Once in NNF, the obtained equilibrium models actually coincide with answer sets for the syntactic fragments of nested expressions (Lifschitz et al. 1999) or for regular programs (Gelfond and Lifschitz 1993). For this reason, most papers on Equilibrium Logic for ASP assumed a reduction to NNF from the very beginning, and little attention was paid to the behaviour of formulas in the scope of strong negation under a logic programming perspective. There are, however, cases in which this behaviour is not aligned with the reduct-based understanding of nested expressions in ASP. Take, for instance, the formula:

$$\sim\neg p \rightarrow p \tag{1}$$

Its NNF reduction removes the combination of negations $\sim\neg$ and produces the tautological rule

[1] In fact, the construct "$\sim train$" is normally treated in ASP as a new atom $train'$ and an implicit constraint $train \wedge train' \rightarrow \bot$ is used to guarantee that both atoms cannot be true simultaneously.

$p \to p$ whose unique equilibrium model is $\emptyset$, i.e., neither p nor $\sim p$ hold. However, if we start instead from the formula $\sim\neg\neg\neg p \to p$, the NNF reduction removes again the first pair of negations producing the rule $\neg\neg p \to p$ with a second answer set $\{p\}$. This illustrates that we cannot replace $\neg p$ by $\neg\neg\neg p$ in the scope of strong negation, even though they would produce the same effect in any reduct of the style of (Lifschitz et al. 1999) for nested expressions.

In this paper, we consider a different characterisation of '$\sim$' in HT and Equilibrium Logic. We call this variant *explicit negation* to differentiate it from Nelson's strong negation. To test its adequacy, we start generalising the definition of nested expression by introducing an arbitrary nesting of '$\sim$', adapting the definitions of reduct and answer set from (Lifschitz et al. 1999) to that context. After that, we prove that equilibrium models (with explicit negation) capture the answer sets for these extended nested expressions and, in fact, preserve the strong equivalences from (Lifschitz et al. 1999) even for arbitrary formulas (including implication). We also prove several properties of HT with explicit negation and provide a reduction to NNF that produces a different effect from $\mathcal{N}_5$ when applied on implications or default negation.

The rest of the paper is organised as follows. In the next section, we introduce the extended definition of answer sets for programs with nested expressions, where explicit negation can be arbitrarily combined both in the rule bodies and the rule heads. In Section 3, we present Equilibrium Logic with explicit negation and in particular, its new monotonic basis, $\mathcal{X}_5$, since the selection of equilibrium models is the same one as in (Pearce 1997). Section 4 provides a five-valued characterisation of $\mathcal{X}_5$ and studies different types of equivalence relations, including variants of strong equivalence. In Section 5, we briefly explain the main differences between explicit ($\mathcal{X}_5$) and strong ($\mathcal{N}_5$) negations. Finally, Section 6 concludes the paper.

2 Nested expressions with explicit negation

We begin describing the syntax of nested expressions, starting from a set of atoms At. A *nested expression* F is defined with the following grammar:

$$F ::= \top \mid \bot \mid p \mid F \vee F \mid F \wedge F \mid \neg F \mid \sim F$$

where p is any atom $p \in At$. The two negations $\neg$ and $\sim$ are respectively called *default* and *explicit* negation (the latter is also called *classical* in the ASP literature). An *explicit literal* is either an atom p or its explicit negation $\sim p$. A *default literal* is either an explicit literal A or its default negation $\neg A$. Thus, given atom p, we can form the default literals $p, \sim p, \neg p$ and $\neg\sim p$. As we can see, the main difference with respect to (Lifschitz et al. 1999) is that, in that case, the explicit negation[2] operator $\sim$ was only used for explicit literals, whereas in this definition, it can be arbitrarily nested. For instance, $\sim(p \vee \neg q)$ is a nested expression under this new definition, but it is not under (Lifschitz et al. 1999). A *rule* is an implication of the form $F \to G$ where F and G are nested expressions respectively called the *body* and the *head* of the rule. A rule of the form $\top \to G$ is sometimes abbreviated as G and is further called a *fact* if G is an explicit literal. A *logic program* is a set of rules. We say that a nested expression, a rule or a program is *explicit* if it does not contain default negation.

A program rule $F \to G$ is said to be *regular* if the body $F = B_1 \wedge \cdots \wedge B_n$ is a conjunction of

[2] To be precise, (Lifschitz et al. 1999) used a different notation and names for operators: $\wedge$, $\vee$ and $\neg$ were respectively denoted as comma, semicolon and 'not' in (Lifschitz et al. 1999), whereas explicit negation $\sim$was denoted as $\neg$ and called *classical negation*.

default literals and the head $G = H_1 \vee \cdots \vee H_m$ is a disjunction of default literals. In a regular rule, we allow an empty body $n = 0$ and write $F = \top$ or an empty head $m = 0$ and $G = \bot$ but not both. A program is *regular* if all its rules are regular.

An *interpretation* is a set of explicit literals that is consistent, that is, it does not contain both p and $\sim p$ for any atom p. We define when an interpretation T *satisfies* (resp. *falsifies*) a nested expression F, written $T \models F$ (resp. $T \mathrel{=\!\!|} F$) providing the following recursive conditions:

$$
\begin{array}{lll@{\qquad}lll}
T \models \top & & & T \not\mathrel{=\!\!|} \top & \\
T \not\models \bot & & & T \mathrel{=\!\!|} \bot & \\
T \models p & \text{if } p \in T & & T \mathrel{=\!\!|} p & \text{if } \sim p \in T \\
T \models \varphi \wedge \psi & \text{if } T \models \varphi \text{ and } T \models \psi & & T \mathrel{=\!\!|} \varphi \wedge \psi & \text{if } T \mathrel{=\!\!|} \varphi \text{ or } T \mathrel{=\!\!|} \psi \\
T \models \varphi \vee \psi & \text{if } T \models \varphi \text{ or } T \models \psi & & T \mathrel{=\!\!|} \varphi \vee \psi & \text{if } T \mathrel{=\!\!|} \varphi \text{ and } T \mathrel{=\!\!|} \psi \\
T \models \sim\varphi & \text{if } T \mathrel{=\!\!|} \varphi & & T \mathrel{=\!\!|} \sim\varphi & \text{if } T \models \varphi \\
T \models \neg\varphi & \text{if } T \not\models \varphi & & T \mathrel{=\!\!|} \neg\varphi & \text{if } T \models \varphi
\end{array}
$$

As an example, given $At = \{p, q\}$ and $T = \{\sim p\}$ we have $T \models \sim p \vee q$ because $T \models \sim p$ (i.e. $T \mathrel{=\!\!|} p$) although neither $T \models q$ nor $T \mathrel{=\!\!|} q$, that is, q is undefined. The latter can be expressed as $T \models \neg q \wedge \neg \sim q$ (i.e., q is neither true nor false). As another example, $T \mathrel{=\!\!|} p \wedge q$ because $T \mathrel{=\!\!|} p$ even though, as we said, q is undefined. We say that φ is *valid* if we have $T \models \varphi$ for every interpretation T. The logic induced by these valid expressions precisely corresponds to *classical logic with strong negation* as studied by Vakarelov (1977). Note that, as usual in classical logic, $\varphi \rightarrow \psi$ is definable as $\neg\varphi \vee \psi$ in this context.

Let Π be an explicit program. A consistent set of literals T is a *model* of Π if, for every rule $F \rightarrow G$ in Π, $T \models G$ whenever $T \models F$.

Definition 1 (*reduct*)
The reduct of a nested expression F with respect to an interpretation T is denoted as F^T and defined recursively as follows:

$$
\begin{array}{rcll}
p^T & \stackrel{\text{def}}{=} & p & \qquad\text{for any atom } p \in At \\
(F \wedge G)^T & \stackrel{\text{def}}{=} & F^T \wedge G^T & \\
(F \vee G)^T & \stackrel{\text{def}}{=} & F^T \vee G^T & \\
(\sim F)^T & \stackrel{\text{def}}{=} & \sim(F^T) & \\
(\neg F)^T & \stackrel{\text{def}}{=} & \begin{cases} \bot & \text{if } T \models F \\ \top & \text{otherwise} \end{cases} &
\end{array}
$$

The *reduct* of a program Π with respect to T corresponds to the explicit program:
$\Pi^T \stackrel{\text{def}}{=} \{\, (F^T \rightarrow G^T) \mid (F \rightarrow G) \in \Pi \,\}$. □

Proposition 1
For any consistent set of literals T and any nested formula F:

- $T \models F$ iff $T \models F^T$;
- $T \mathrel{=\!\!|} F$ iff $T \mathrel{=\!\!|} F^T$. □

Definition 2 (*answer set*)
A consistent set of literals T is an *answer set* of a program Π if it is a $\subseteq$-minimal model of the reduct Π^T. □

Notice that the definitions of reduct and answer set for the case of regular programs directly coincide with the standard definitions in ASP without nested expressions (Gelfond and Lifschitz 1991). They also coincide with (Lifschitz et al. 1999), defined on the case of programs with nested expressions where '$\sim$' is only in front of atoms.

Example 1

Take the program consisting of the single rule (1). For $At = \{p\}$, we have three possible interpretations $T_1 = \{p\}$, $T_2 = \{\sim p\}$ and $T_3 = \emptyset$. This yields two possible reducts $\Pi^{T_1} = \{\sim\bot \to p\}$ and $\Pi^{T_2} = \Pi^{T_3} = \{\sim\top \to p\}$. It is easy to see that their corresponding minimal models are T_1 and T_3 which constitute the two answer sets of Π. □

Example 2

Take the program consisting of the single rule:

$$\neg(\mathit{bird} \wedge \sim \mathit{flies}) \to \sim(\mathit{bird} \wedge \sim \mathit{flies}) \tag{2}$$

capturing the idea that "being a bird that does not fly" should be false by default. If we choose any interpretation T such that $T \models \mathit{bird} \wedge \sim \mathit{flies}$ then the reduct will have a single rule with $\bot$ in the body and the minimal model will be $\emptyset$ which does not satisfy $\mathit{bird} \wedge \sim \mathit{flies}$. If $T \not\models \mathit{bird} \wedge \sim \mathit{flies}$ instead, the reduct becomes $\top \to \sim(\mathit{bird} \wedge \sim \mathit{flies})$ and the minimal models of this program are $\{\sim \mathit{bird}\}$ and $\{\mathit{flies}\}$ that, as they are both compatible with the assumption for T, they become the two answer sets of (2).

Suppose we extend now (2) with the fact *bird*. Doing so, it is easy to see that the only answer set becomes $\{\mathit{flies}\}$. Analogously, if we take (2) plus the fact $\sim\mathit{flies}$ the only answer set becomes $\{\sim \mathit{bird}\}$. Finally, if we add the facts *bird* and $\sim\mathit{flies}$ to (2), the default is deactivated and we get the unique answer set $\{\mathit{bird}, \sim\mathit{flies}\}$. □

3 Equilibrium logic with explicit negation

We start defining the monotonic logic of *Here-and-There with explicit negation*, $\mathcal{X}_5$. Let At be a set of atoms. A *formula* φ is an expression built with the grammar:

$$\varphi ::= p \mid \bot \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \sim\varphi$$

for any atom $p \in At$. We also use the abbreviations:

$$\neg\varphi \stackrel{\text{def}}{=} (\varphi \to \bot) \qquad \varphi \leftrightarrow \psi \stackrel{\text{def}}{=} (\varphi \to \psi) \wedge (\psi \to \varphi)$$

$$\top \stackrel{\text{def}}{=} \neg\bot \qquad \varphi \Leftrightarrow \psi \stackrel{\text{def}}{=} (\varphi \leftrightarrow \psi) \wedge (\sim\varphi \leftrightarrow \sim\psi)$$

Given a pair of formulas φ and α, we write $\varphi[\alpha/p]$ to denote the uniform substitution of all occurrences of atom p in φ by α. As usual, a *theory* is a set of formulas. We sometimes understand finite theories (or subtheories) as the conjunction of their formulas. Notice that programs with nested expressions are also theories under this definition.

An $\mathcal{X}_5$-*interpretation* is a pair $\langle H, T\rangle$ of consistent sets of explicit literals (respectively standing for "here" and "there") satisfying $H \subseteq T$. We say that the interpretation is *total* when $H = T$.

Definition 3 ($\mathcal{X}_5$ *Satisfaction/falsification*)

We say that $\langle H,T\rangle$ *satisfies* (resp. *falsifies*) a formula φ, written $\langle H,T\rangle \models \varphi$ (resp. $\langle H,T\rangle \Dashv \varphi$), when the following recursive conditions hold:

$$
\begin{array}{lll|lll}
\langle H,T\rangle \models \top & & & \langle H,T\rangle \not\Dashv \top & & \\
\langle H,T\rangle \not\models \bot & & & \langle H,T\rangle \Dashv \bot & & \\
\langle H,T\rangle \models p & \text{if } p \in H & & \langle H,T\rangle \Dashv p & \text{if } {\sim} p \in H & \\
\langle H,T\rangle \models \varphi \wedge \psi & \text{if } \langle H,T\rangle \models \varphi \text{ and } \langle H,T\rangle \models \psi & & \langle H,T\rangle \Dashv \varphi \wedge \psi & \text{if } \langle H,T\rangle \Dashv \varphi \text{ or } \langle H,T\rangle \Dashv \psi & \\
\langle H,T\rangle \models \varphi \vee \psi & \text{if } \langle H,T\rangle \models \varphi \text{ or } \langle H,T\rangle \models \psi & & \langle H,T\rangle \Dashv \varphi \vee \psi & \text{if } \langle H,T\rangle \Dashv \varphi \text{ and } \langle H,T\rangle \Dashv \psi & \\
\langle H,T\rangle \models {\sim}\varphi & \text{if } \langle H,T\rangle \Dashv \varphi & & \langle H,T\rangle \Dashv {\sim}\varphi & \text{if } \langle H,T\rangle \models \varphi & \\
\langle H,T\rangle \models \varphi \to \psi & \text{if both} & & \langle H,T\rangle \Dashv \varphi \to \psi & \text{if } \langle T,T\rangle \models \varphi \text{ and } \langle H,T\rangle \Dashv \psi & \\
 & (i)\langle H,T\rangle \not\models \varphi \text{ or } \langle H,T\rangle \models \psi & & & & \\
 & (ii)\langle T,T\rangle \not\models \varphi \text{ or } \langle T,T\rangle \models \psi & & & & \square
\end{array}
$$

A formula φ is a *tautology* (or is *valid*), written $\models \varphi$, if it is satisfied by every possible interpretation. We say that an $\mathcal{X}_5$-interpretation $\langle H,T\rangle$ is a *model* of a theory Γ, written $\langle H,T\rangle \models \Gamma$, if $\langle H,T\rangle \models \varphi$ for all $\varphi \in \Gamma$. The next observation about Definition 3 connects satisfaction '$\models$' with standard HT.

Observation 1

The satisfaction relation '$\models$' (left column in Def. 3) of any formula corresponds to regular HT satisfaction up to the first occurrence of '$\sim$', where the falsification '$\Dashv$' comes into play. $\square$

As a result, any tautology from HT can be shifted to $\mathcal{X}_5$, even if its atoms are uniformly replaced by subformulas containing explicit negation.

Theorem 1

If formula φ is HT valid (and so, it does not contain $\sim$) then $\varphi[\alpha/p]$ is also $\mathcal{X}_5$ valid, for any formula α and any atom p. $\square$

If we choose any p not occurring in φ, then $\varphi[\alpha/p] = \varphi$ and the theorem above is just saying that $\mathcal{X}_5$ is a conservative extension of HT. But it can also be exploited further by replacing, in the HT tautology, any atom by an arbitrary formula containing negation. For instance, if explicit negation only occurs in front of atoms, we essentially get HT with explicit literals playing the role of atoms (disregarding inconsistent models). However, when we combine explicit negation in an arbitrary way, some usual properties of HT need to be checked in the new context.

Lemma 1

Let T be a consistent set of literals and F a nested expression. Then:

- $\langle T,T\rangle \models F$ iff $T \models F$;
- $\langle T,T\rangle \Dashv F$ iff $T \Dashv F$. $\square$

Theorem 2 (*Persistence*)

For any $\mathcal{X}_5$-interpretation $\langle H,T\rangle$ and any formula φ then both:

(i) $\langle H,T\rangle \models \varphi$ implies $\langle T,T\rangle \models \varphi$;
(ii) $\langle H,T\rangle \Dashv \varphi$ implies $\langle T,T\rangle \Dashv \varphi$. $\square$

Proposition 2
For any $\mathcal{X}_5$-interpretation $\langle H,T\rangle$, any formula φ:

- $\langle H,T\rangle \models \neg\varphi$ iff $\langle T,T\rangle \not\models \varphi$;
- $\langle H,T\rangle \;=\!\mid\; \neg\varphi$ iff $\langle T,T\rangle \models \varphi$. $\square$

The following results establish a connection between $\mathcal{X}_5$ and the reduct of a nested expression or a program.

Lemma 2
Let $\langle H,T\rangle$ be an $\mathcal{X}_5$-interpretation and F a nested expression. Then:

- $\langle H,T\rangle \models F$ iff $H \models F^T$;
- $\langle H,T\rangle \;=\!\mid\; F$ iff $H \;=\!\mid\; F^T$. $\square$

Corollary 1
For any consistent set of literals T and any program Π: $\langle T,T\rangle \models \Pi$ iff $T \models \Pi$. $\square$

Proposition 3
For any $\mathcal{X}_5$-intepretation $\langle H,T\rangle$ and any program Π:

$$\langle H,T\rangle \models \Pi \text{ iff } H \text{ is a model of } \Pi^T \text{ and } T \text{ is a model of } \Pi. \qquad \square$$

Definition 4 (*Equilibrium model*)
A total $\mathcal{X}_5$-interpretation $\langle T,T\rangle$ is an *equilibrium model* of a theory Γ if $\langle T,T\rangle$ is a model of Γ and there is no other model $\langle H,T\rangle$ of Γ with $H \subset T$. $\square$

Equilibrium logic (with explicit negation) is the non-monotonic logic induced by equilibrium models. The following theorem guarantees that equilibrium models and answer sets coincide for the syntax of programs with nested expressions.

Theorem 3
An interpretation T is an answer set of a program Π iff $\langle T,T\rangle$ is an equilibrium model of Π. $\square$

To conclude this section, we provide an alternative reduct definition for arbitrary formulas (and not just nested expressions) obtained as a generalisation of Ferraris' reduct (Ferraris 2005). This generalisation introduces a main feature[3] with respect to (Ferraris 2005): it actually uses two dual transformations, φ^T_+ and φ^T_-, to obtain a symmetric behaviour depending on the number of explicit negations in the scope.

[3] We also provide a translation for implications $\alpha \rightarrow \beta$ but this is not strictly necessary: for computing the reduct, they can be previously replaced by $\neg\alpha \vee \beta$.

Definition 5
Given a formula φ and an interpretation T (a consistent set of explicit literals) we define the following pair of mutually recursive transformations:

$$\varphi_+^T \stackrel{\text{def}}{=} \begin{cases} \bot & \text{if } T \not\models \varphi \\ p & \text{if } \varphi = p \in At, p \in T \\ \alpha_+^T \otimes \beta_+^T & \text{if } T \models \varphi, \varphi = \alpha \otimes \beta, \\ & \text{ for } \otimes \in \{\vee, \wedge\} \\ \neg(\alpha_+^T) \vee \beta_+^T & \text{if } T \models \varphi, \varphi = \alpha \rightarrow \beta \\ \neg(\alpha_+^T) & \text{if } T \models \varphi, \varphi = \neg\alpha, \\ \sim(\alpha_-^T) & \text{if } T \models \varphi, \varphi = \sim\alpha \end{cases} \qquad \varphi_-^T \stackrel{\text{def}}{=} \begin{cases} \top & \text{if } T \not\mathrel{=\!\!|} \varphi \\ p & \text{if } \varphi = p \in At, \sim p \in T \\ \alpha_-^T \otimes \beta_-^T & \text{if } T \mathrel{=\!\!|} \varphi, \varphi = \alpha \otimes \beta, \\ & \text{ for } \otimes \in \{\vee, \wedge\} \\ \beta_-^T & \text{if } T \mathrel{=\!\!|} \varphi, \varphi = \alpha \rightarrow \beta \\ \bot & \text{if } T \mathrel{=\!\!|} \varphi, \varphi = \neg\alpha \\ \sim(\alpha_+^T) & \text{if } T \mathrel{=\!\!|} \varphi, \varphi = \sim\alpha \end{cases}$$

The reduct Γ_+^T of a theory Γ is just defined as the set $\{\varphi_+^T \mid \varphi \in \Gamma\}$. □

For instance, given $\varphi = (2)$ and $T = \{\sim bird\}$, the reader can check that the application of the definition above eventually produces the formula $\varphi_+^T = \neg\neg\bot \vee \sim(bird \wedge \top)$ which is equivalent to $\sim bird$. If we take $T = \{flies\}$ instead, the result is $\varphi_+^T = \neg\neg\bot \vee \sim(\top \wedge \sim flies)$ that is equivalent to *flies*. As a third example, if we take $T = \{bird\}$ then we directly get $\varphi_+^T = \bot$.

Theorem 4
For any formula φ and any pair of interpretations $H \subseteq T$:

(i) $H \models \varphi_+^T$ iff $\langle H, T\rangle \models \varphi$;
(ii) $H \mathrel{=\!\!|} \varphi_-^T$ iff $\langle H, T\rangle \mathrel{=\!\!|} \varphi$. □

From Lemma 2 and Theorem 4 we immediately conclude:

Corollary 2
For any nested expression F and any pair of interpretations $H \subseteq T$:

(i) $H \models F^T$ iff $T \models F$ and $H \models F_+^T$;
(ii) $H \mathrel{=\!\!|} F^T$ iff $T \mathrel{=\!\!|} F$ and $H \mathrel{=\!\!|} F_-^T$. □

Corollary 3
$\langle T, T\rangle$ is an equilibrium model of Γ iff T is a minimal model of Γ_+^T. □

Back to the example formula φ =(2), taking $T = \{\sim bird\}$ we saw that φ_+^T is equivalent to $\sim bird$ whose minimal model is obviously T. Therefore, $\langle T, T\rangle$ is an equilibrium model.

4 Multivalued characterisation and equivalence relations

An alternative way of characterising $\mathcal{X}_5$ is as a five-valued logic defined as follows. Given any $\mathcal{X}_5$-interpretation $M = \langle H, T\rangle$ we define its corresponding 5-valued mapping $M : At \rightarrow \{-2, -1, 0, 1, 2\}$ so that, for any atom $p \in At$:

$$M(p) \stackrel{\text{def}}{=} \begin{cases} 2 & \text{if } p \in H \\ -2 & \text{if } \sim p \in H \\ 1 & \text{if } p \in T \setminus H \\ -1 & \text{if } \sim p \in T \setminus H \\ 0 & \text{otherwise, i.e., } p \notin T, \sim p \notin T \end{cases}$$

We can read these five values as follows: 2 = *proved to be true*; -2 = *proved to be false*; 1 = *true by default*; -1 = *false by default*; and 0 = *undefined*. Notice that values 1 and -1 are used for explicit literals in $T \setminus H$. As a consequence:

Proposition 4
An $\mathcal{X}_5$-interpretation $M = \langle H, T \rangle$ is total (i.e. $H = T$) iff $M(p) \in \{-2, 0, 2\}$ for all $p \in At$. □

Definition 6 (*Valuation of formulas*)
This 5-valuation can be extended to arbitrary formulas in the following way:

$$
\begin{aligned}
M(\bot) &\stackrel{\text{def}}{=} -2 \\
M(\top) &\stackrel{\text{def}}{=} 2 \\
M(\varphi \wedge \psi) &\stackrel{\text{def}}{=} \min(M(\varphi), M(\psi)) \\
M(\varphi \vee \psi) &\stackrel{\text{def}}{=} \max(M(\varphi), M(\psi)) \\
M(\varphi \to \psi) &\stackrel{\text{def}}{=} \begin{cases} 2 & \text{if } M(\varphi) \le \max(M(\psi), 0) \\ M(\psi) & \text{otherwise} \end{cases} \\
M(\sim\varphi) &\stackrel{\text{def}}{=} -M(\varphi)
\end{aligned}
$$
□

The designated value is 2, that is, we will understand that a formula is satisfied when $M(\varphi) = 2$. Moreover, a complete correspondence with the satisfaction/falsification of formulas given in the previous section is fixed by the following theorem:

Theorem 5
For any $\mathcal{X}_5$-interpretation $M = \langle H, T \rangle$ and any formula φ:

- $\langle H, T \rangle \models \varphi$ iff $M(\varphi) = 2$;
- $\langle T, T \rangle \models \varphi$ iff $M(\varphi) > 0$;
- $\langle H, T \rangle \mathrel{=\!\!\mid} \varphi$ iff $M(\varphi) = -2$;
- $\langle T, T \rangle \mathrel{=\!\!\mid} \varphi$ iff $M(\varphi) < 0$. □

The equilibrium condition given in Definition 4 can be rephrased in 5-valued terms as follows. Given two $\mathcal{X}_5$-interpretations $M = \langle H, T \rangle$ and $M' = \langle H', T' \rangle$ we say that M is *smaller* than M', written $M \le M'$, when $T = T'$ and $H \subseteq H'$.

Proposition 5
Let M and M' be a pair of $\mathcal{X}_5$-interpretations. Then $M \le M'$ iff, for any atom $p \in At$, the following three conditions hold:

1. $M(p) = 0$ iff $M'(p) = 0$;
2. If $M(p) > 0$, then $M(p) \le M'(p)$;
3. If $M(p) < 0$, then $M'(p) \le M(p)$. □

Theorem 6
A total interpretation $M = \langle T, T \rangle$ is an equilibrium model of a theory Γ iff $M(\varphi) = 2$ for all $\varphi \in \Gamma$ and there is no $M' < M$ such that $M'(\varphi) = 2$ for all $\varphi \in \Gamma$.

Proof
It follows from Theorem 5 and the definition of $\le$ relation. □

The truth tables derived from Definition 6 are depicted in Figure 1, including the tables for derived operators '$\neg$', '$\leftrightarrow$' and '$\Leftrightarrow$'. Note that the table for $\neg\varphi = (\varphi \to \bot)$ is just the first column of the table for '$\to$' since the evaluation of '$\bot$' is fixed to -2. It is easy to check, for instance, that the following implication is valid:

$$\sim\varphi \to \neg\varphi \tag{3}$$

$\wedge$	-2	-1	0	1	2
-2	-2	-2	-2	-2	-2
-1	-2	-1	-1	-1	-1
0	-2	-1	0	0	0
1	-2	-1	0	1	1
2	-2	-1	0	1	2

$\vee$	-2	-1	0	1	2
-2	-2	-1	0	1	2
-1	-1	-1	0	1	2
0	0	0	0	1	2
1	1	1	1	1	2
2	2	2	2	2	2

$\to$	-2	-1	0	1	2
-2	2	2	2	2	2
-1	2	2	2	2	2
0	2	2	2	2	2
1	-2	-1	0	2	2
2	-2	-1	0	1	2

φ	$\sim\varphi$
-2	2
-1	1
0	0
1	-1
2	-2

φ	$\neg\varphi$
-2	2
-1	2
0	2
1	-2
2	-2

$\leftrightarrow$	-2	-1	0	1	2
-2	2	2	2	-2	-2
-1	2	2	2	-1	-1
0	2	2	2	0	0
1	-2	-1	0	2	1
2	-2	-1	0	1	2

$\Leftrightarrow$	-2	-1	0	1	2
-2	2	1	0	-2	-2
-1	1	2	0	-1	-2
0	0	0	2	0	0
1	-2	-1	0	2	1
2	-2	-2	0	1	2

Fig. 1. Truth tables for $\mathcal{X}_5$.

expressing that explicit negation is stronger than default negation[4]. Moreover, default negation is definable in terms of implication and explicit negation (without resorting to $\bot$) since, with some effort, it can be checked that the table for $\neg\varphi$ can be equally obtained through the expression:

$$\sim((\varphi \to \sim\varphi) \to \sim(\varphi \to \sim\varphi))$$

An important remark regarding equivalence is that to express that this (or any) pair of formulas are equivalent, double implication does not suffice. This is because, as we can see in the tables, $M(\varphi \leftrightarrow \psi) = 2$ does not imply that $M(\varphi) = M(\psi)$. To get such a correspondence, we must resort instead to the stronger '$\Leftrightarrow$' for which $M(\varphi \Leftrightarrow \psi) = 2$ holds if and only if $M(\varphi) = M(\psi)$. This lack of the '$\leftrightarrow$' equivalence (we call it *weak* equivalence) has an important consequence: it does not define a congruence relation since $\models \alpha \leftrightarrow \beta$ no longer implies that we can freely replace subformula α by β in any arbitrary context: it may be the case that $\not\models \sim\alpha \leftrightarrow \sim\beta$. For instance, we can easily check that $\models p \wedge \neg p \leftrightarrow \bot$ because $\min(M(p), M(\neg p)) \leq 0$ and $M(\bot) = -2$, so $M(p \wedge \neg p \leftrightarrow \bot) = 2$ for any M. However, we cannot replace $p \wedge \neg p$ by $\bot$ in any context. Take the program Π consisting of the unique rule

$$\sim(p \wedge \neg p) \tag{4}$$

with empty body. Interpretation $T = \{\sim p\}$ is an answer set because $\Pi^T = \{\sim(p \wedge \top)\}$ has $\{\sim p\}$ as minimal model (in fact, it is the unique answer set) but if we replace $p \wedge \neg p$ by $\bot$ in Π we get

[4] This property is called the *coherence* principle in (Pereira and Alferes 1992).

the trivial program $\{\sim\bot\}$ whose unique answer set is $\emptyset$. Although weak equivalence does not guarantee arbitrary replacements, it can be used to replace formulas in a theory, as stated below:

Proposition 6
Let α, β be a pair of formulas such that $\models \alpha \leftrightarrow \beta$. Then, $M \models \Gamma \cup \{\alpha\}$ iff $M \models \Gamma \cup \{\beta\}$ for any theory Γ and $\mathcal{X}_5$-interpretation M. $\square$

As we mentioned before, for obtaining a congruence relation we can use validity of '$\Leftrightarrow$' instead, which guarantees the following substitution theorem.

Theorem 7 (*Substitution*)
Let α, β be a pair of formulas satisfying $\models \alpha \Leftrightarrow \beta$. Then, for any formula φ, we also obtain $\models \varphi[\alpha/p] \Leftrightarrow \varphi[\beta/p]$. $\square$

Still, there are some cases in which $\leftrightarrow$ can be used for substitution, provided that the replaced formulas are not in the scope of explicit negation.

Theorem 8
Let φ be a formula where atom p only occurs outside the scope of explicit negation, and let α, β be two formulas satisfying $\models \alpha \leftrightarrow \beta$. Then, $\models \varphi[\alpha/p] \leftrightarrow \varphi[\beta/p]$. $\square$

An important property of ASP related to HT equivalence is *strong equivalence*. We say that two programs (resp. theories) Γ and Γ' are *strongly equivalent* iff $\Gamma \cup \Delta$ and $\Gamma' \cup \Delta$ have the same answer sets (resp. equilibrium models), for any additional program (resp. theory) Δ. When we talk about strong equivalence of formulas α and β we assume they correspond to the singleton theories $\{\alpha\}$ and $\{\beta\}$. As shown in (Lifschitz et al. 2001) (for the case without explicit negation), two programs or theories are strongly equivalent if and only if they are HT equivalent. Since the '$\leftrightarrow$' relation in HT is congruent, there is no difference between strong equivalence (replacing formulas in a theory) and substitution (replacing subformulas in a formula). However, as explained in (Ortiz and Osorio 2007), once congruence is lost, we can further refine strong equivalence in the following way.

Definition 7 (*Strong equivalence on substitution*)
We say that two formulas α and β are *strongly equivalent on substitutions* if $\Delta \cup \{\varphi[\alpha/p]\}$ and $\Delta \cup \{\varphi[\beta/p]\}$ have the same equilibrium models, for any formula φ and theory Δ.

The proof of the next lemma can be obtained following similar steps to the proof of the main theorem in (Lifschitz et al. 2001) replacing atoms in that case by explicit literals in ours.

Lemma 3
Let α and β be two formulas and be an interpretation such that $\langle H,T\rangle \models \alpha$ but $\langle H,T\rangle \not\models \beta$. Then, there is a finite theory Δ such that $\langle T,T\rangle$ is an equilibrium model of one of $\Delta \cup \{\beta\}$, $\Delta \cup \{\alpha\}$ but not of both. $\square$

Theorem 9
Formulas α and β are strongly equivalent iff $\models \alpha \leftrightarrow \beta$. $\square$

Theorem 10
Formulas α and β are strongly equivalent on substitutions iff $\models \alpha \Leftrightarrow \beta$. □

The following set of valid equivalences allow us reducing any nested expression with explicit negation to an *explicit negation normal form* (NNF) where $\sim$ is only applied on atoms.

$$\begin{aligned} \sim\top &\Leftrightarrow \bot && (5)\\ \sim\bot &\Leftrightarrow \top && (6)\\ \sim(\varphi \wedge \psi) &\Leftrightarrow \sim\varphi \vee \sim\psi && (7)\\ \sim(\varphi \vee \psi) &\Leftrightarrow \sim\varphi \wedge \sim\psi && (8)\\ \sim\sim\varphi &\Leftrightarrow \varphi && (9)\\ \sim\neg\varphi &\Leftrightarrow \neg\neg\varphi && (10) \end{aligned}$$

For instance, we can reduce the nested expression (4) to NNF as follows:

$$\begin{aligned} \sim(p \wedge \neg p) &\Leftrightarrow \sim p \vee \sim\neg p && \text{by (7)}\\ &\Leftrightarrow \sim p \vee \neg\neg p && \text{by (10)} \end{aligned}$$

Programs in NNF correspond to the original syntax in (Lifschitz et al. 1999). That paper provided several transformations that allowed reducing any program in NNF to a regular program. These transformations included commutativity and associativity of conjunction and disjunction (which are obviously satisfied in $\mathcal{X}_5$) plus the equivalences in the following proposition.

Proposition 7
The following formulas are $\mathcal{X}_5$ tautologies:

$$\begin{aligned} \varphi \wedge (\psi \vee \gamma) \Leftrightarrow (\varphi \wedge \psi) \vee (\varphi \wedge \gamma) \quad & \varphi \vee (\psi \wedge \gamma) \Leftrightarrow (\varphi \vee \psi) \wedge (\varphi \vee \gamma) && (11)\\ \varphi \wedge \bot \Leftrightarrow \bot \quad & \varphi \vee \top \Leftrightarrow \top && (12)\\ \varphi \wedge \top \Leftrightarrow \varphi \quad & \varphi \vee \bot \Leftrightarrow \varphi && (13)\\ \neg(\varphi \wedge \psi) \Leftrightarrow \neg\varphi \vee \neg\psi \quad & \neg(\varphi \vee \psi) \Leftrightarrow \neg\varphi \wedge \neg\psi && (14)\\ \neg\top \Leftrightarrow \bot \quad & \neg\bot \Leftrightarrow \top && (15) \end{aligned}$$

$$\begin{aligned} \neg\neg\neg\varphi &\Leftrightarrow \neg\varphi && (16)\\ \varphi \rightarrow \psi \wedge \gamma &\Leftrightarrow (\varphi \rightarrow \psi) \wedge (\varphi \rightarrow \gamma) && (17)\\ \varphi \vee \psi \rightarrow \gamma &\Leftrightarrow (\varphi \rightarrow \gamma) \wedge (\psi \rightarrow \gamma) && (18)\\ \varphi \wedge \neg\neg\psi \rightarrow \gamma &\Leftrightarrow \varphi \rightarrow \gamma \vee \neg\psi && (19)\\ \varphi \rightarrow \gamma \vee \neg\neg\psi &\Leftrightarrow \varphi \wedge \neg\psi \rightarrow \gamma && (20) \end{aligned}$$

and correspond to the transformations in (Lifschitz et al. 1999). □

For instance, as we saw, (4) was equivalent to $\sim p \vee \neg\neg p$ but this can be further transformed into the regular rule $\neg p \rightarrow \sim p$ commonly used to assign falsity of p by default.

Example 3 (*Example 2 continued*)
Rule (2) can be transformed as follows:

$$\begin{aligned} (2) \quad &\Leftrightarrow \neg bird \vee \neg\sim flies \rightarrow \sim(bird \wedge \sim flies) && \text{by (14)}\\ &\Leftrightarrow \neg bird \vee \neg\sim flies \rightarrow \sim bird \vee \sim\sim flies && \text{by (7)}\\ &\Leftrightarrow \neg bird \vee \neg\sim flies \rightarrow \sim bird \vee flies && \text{by (9)}\\ &\Leftrightarrow (\neg bird \rightarrow \sim bird \vee flies)\\ &\quad \wedge(\neg\sim flies \rightarrow \sim bird \vee flies) && \text{by (18)} \end{aligned}$$

and the last step is a conjunction of two regular rules as in standard ASP solvers. □

Reduction to NNF is also possible on arbitrary formulas. For that purpose, we can combine (5)-(10) with the following valid (weak) equivalence:

$$\sim(\varphi \to \psi) \quad \leftrightarrow \quad \neg\neg\varphi \wedge \sim\psi \tag{21}$$

However, the reduction must be done with some care, because this last equivalence cannot be shifted to $\Leftrightarrow$. Indeed, the left and right expressions have different valuations when $M(\varphi) = M(\psi) = 1$, obtaining $M(\sim(\varphi \to \psi)) = -2 \neq -1 = M(\neg\neg\varphi \wedge \sim\psi)$. Fortunately, Theorem 8 allows us applying (21) from the outermost occurrence of $\sim$ and then recursively combining with (5)-(10) until $\sim$ is only applied to atoms.

Theorem 11
For any formula φ there exists a formula ψ in NNF such that $\models \varphi \leftrightarrow \psi$. □

For instance, we can reduce the following formula into NNF as follows:

$$\begin{aligned} \sim(a \to \sim b \wedge (c \to d)) \quad & \leftrightarrow \quad \neg\neg a \wedge \sim(\sim b \wedge (c \to d)) \\ & \leftrightarrow \quad \neg\neg a \wedge (\sim\sim b \vee \sim(c \to d)) \\ & \leftrightarrow \quad \neg\neg a \wedge (b \vee \neg\neg c \wedge \sim d) \end{aligned}$$

However, we cannot apply (21) making a replacement in the scope of explicit negation. A clear counterexample is the formula $\sim\sim(p \to q)$ that, due to (9), is strongly equivalent to $p \to q$, but applying (21) inside would incorrectly lead to the nested expression $\sim(\neg\neg p \wedge \sim q)$ that can be transformed into the strongly equivalent expression $\neg p \vee q$, different from $p \to q$ in ASP.

5 Related work

As explained in the introduction, this work is obviously related to the characterisation of '$\sim$' as Nelson's *strong negation* (Nelson 1949) for intermediate logics. In particular, the addition of strong negation to HT produces the five-valued logic $\mathcal{N}_5$ already present in the original definition of Equilibrium Logic (Pearce 1997). In fact, the interpretations and the truth values we have chosen for $\mathcal{X}_5$ coincide with those for $\mathcal{N}_5$, and their evaluation of (non-derived) connectives $\top, \wedge, \vee$ and $\to$ from Figure 1 also coincide in both logics, except for one difference in the table of implication: the value for $M(\varphi) = 1$ and $M(\psi) = -2$ changes from -2 to -1 in $\mathcal{N}_5$. This change and its result on derived operators is shown in Figure 2 where the different values are framed in rectangles. As a result, $\mathcal{N}_5$ ceases to satisfy (10) and (21) whose role in the reduction to NNF is respectively replaced by the $\mathcal{N}_5$-valid weak equivalences:

$$\sim\neg\varphi \quad \leftrightarrow \quad \varphi \tag{22}$$

$$\sim(\varphi \to \psi) \quad \leftrightarrow \quad \varphi \wedge \sim\psi \tag{23}$$

The difference between (21) and (23) also reveals the effect on falsification of implication in both logics. While $\langle H, T\rangle \Dashv \varphi \to \psi$ requires $\langle T, T\rangle \models \varphi$ in $\mathcal{X}_5$, this is replaced by condition $\langle H, T\rangle \models \varphi$ in $\mathcal{N}_5$. Curiously, although these two logics provide a different behaviour for $\sim$ as strong versus explicit negation, they actually have the same evaluation for that connective, while their real technical difference lies on falsity of implication.

The reason why $\mathcal{N}_5$ does not capture the extended reduct for nested expressions proposed

$\rightarrow$	-2	-1	0	1	2
-2	2	2	2	2	2
-1	2	2	2	2	2
0	2	2	2	2	2
1	$\boxed{-1}$	-1	0	2	2
2	-2	-1	0	1	2

φ	$\neg\varphi$
-2	2
-1	2
0	2
1	$\boxed{-1}$
2	-2

$\leftrightarrow$	-2	-1	0	1	2
-2	2	2	2	$\boxed{-1}$	-2
-1	2	2	2	-1	-1
0	2	2	2	0	0
1	$\boxed{-1}$	-1	0	2	1
2	-2	-1	0	1	2

$\Leftrightarrow$	-2	-1	0	1	2
-2	2	1	0	$\boxed{-1}$	-2
-1	1	2	0	-1	-2
0	0	0	2	0	0
1	$\boxed{-1}$	-1	0	2	1
2	-2	-2	0	1	2

Fig. 2. Truth tables for $\mathcal{N}_5$ that differ from $\mathcal{X}_5$.

in this paper is that (16) is not valid in that logic. This is because, when $M(\varphi) = 1$, we get $M(\neg\varphi) = -1 \neq -2 = M(\neg\neg\neg\varphi)$. It is still possible to define $\mathcal{N}_5$ operators in $\mathcal{X}_5$ as follows:

$$\begin{aligned} \varphi \overset{\mathcal{N}_5}{\rightarrow} \psi &\overset{\text{def}}{=} \varphi \rightarrow \sim\varphi \vee \psi \\ \overset{\mathcal{N}_5}{\neg}\, \varphi &\overset{\text{def}}{=} \varphi \rightarrow \sim\varphi \end{aligned}$$

using here the $\mathcal{X}_5$ interpretation for implication. Analogously, we can also define the $\mathcal{X}_5$ operators in $\mathcal{N}_5$ in the following way:

$$\begin{aligned} \varphi \overset{\mathcal{X}_5}{\rightarrow} \psi &\overset{\text{def}}{=} (\varphi \rightarrow \psi) \wedge (\sim\psi \rightarrow \neg\neg\neg\varphi) \\ \overset{\mathcal{X}_5}{\neg}\, \varphi &\overset{\text{def}}{=} \neg\neg\neg\varphi \end{aligned}$$

assuming that we interpret implication and $\neg$ under $\mathcal{N}_5$ instead.

An interesting connection between both variants is that the addition of the excluded middle axiom schemata $\varphi \vee \neg\varphi$ imposes the restriction of total models $\langle T, T\rangle$ both in $\mathcal{X}_5$ and in $\mathcal{N}_5$. This means that all atoms and formulas are evaluated in the set $\{-2, 0, 2\}$, for which the truth tables coincide in these two logics and actually collapse to classical logic with strong negation (Vakarelov 1977) introduced in Section 2. This coincidence is important since equilibrium models (and so, answer sets) are total models.

To conclude the section on related work, another possibility for interpreting a second negation '$\sim$' inside intuitionistic logic was provided by (Fariñas del Cerro and Herzig 1996) using a *classical* negation interpretation. Although the idea seems closer to Gelfond and Lifschitz' original terminology for a second negation, it actually provides undesired effects from an ASP point of view. Classical negation in HT means keeping only the satisfaction relation '$\models$' in Definition 3 (falsification '$=\!\mid$' is not needed) but replacing the condition for '$\sim$' so that $\langle H, T\rangle \models \sim\varphi$ if $\langle H, T\rangle \not\models \varphi$. One important effect of this change is that HT with classical negation ceases to satisfy the persistence property (Theorem 2). But perhaps a more important problem from the ASP perspective is that $\neg p$ implies $\sim p$ for any atom p. Thus, the rule $\neg p \rightarrow \sim p$ becomes a tautology in this context, whereas it is normally used in ASP to conclude that p is explicitly false by default.

6 Conclusions

We have introduced a variant of constructive negation in Equilibrium Logic (and its monotonic basis, HT) we called *explicit negation*. This variant shares some similarities with the previous formalisation based on Nelson's strong negation, but changes the interpretation for falsity of implication. We have also introduced a reduct-based definition of answer sets for programs with nested expressions extended with explicit negation, proving the correspondence with equilibrium models.

For future work, we will study a possible axiomatisation. To this aim, it is interesting to observe that the formulas (7)-(9) (in their weak equivalence versions) plus (22) and (23) actually correspond to Vorob'ev axiomatisation (Vorob'ev 1952a; Vorob'ev 1952b) of strong negation in intuitionistic logic. As we saw, the role of (22) and (23) in $\mathcal{N}_5$ is replaced in $\mathcal{X}_5$ by (9) and (21), so an interesting question is whether this replacement may become a complete axiomatisation for explicit negation in $\mathcal{X}_5$ or intuitionistic logic in the general case. We also plan to explore the effect of explicit negation on extensions of equilibrium logic, revisiting the use of strong negation in paraconsistent (Odintsov and Pearce 2005) and partial (Cabalar et al. 2006) equilibrium logic, or considering its combination with partial functions (Cabalar 2011; Cabalar et al. 2014), and temporal (Aguado et al. 2013) or epistemic (Fariñas del Cerro et al. 2015; Cabalar et al. 2019) reasoning.

References

AGUADO, F., CABALAR, P., DIÉGUEZ, M., PÉREZ, G., AND VIDAL, C. 2013. Temporal equilibrium logic: a survey. *Journal of Applied Non-Classical Logics 23,* 1-2, 2–24.

CABALAR, P. 2011. Functional answer set programming. *Theory and Practice of Logic Programming 11,* 2-3, 203–233.

CABALAR, P., FANDINNO, J., AND FARIÑAS DEL CERRO, L. 2019. Founded world views with autoepistemic equilibrium logic. In *Proc. of the 15th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR'19)* (June 3-7).

CABALAR, P., FARIÑAS DEL CERRO, L., PEARCE, D., AND VALVERDE, A. 2014. A free logic for stable models with partial intensional functions. In *Proc. of the 14th European Conf. on Logics in Artificial Intelligence, JELIA 2014, Funchal, Madeira, Portugal, September 24-26*, E. Fermé and J. Leite, Eds. Lecture Notes in Computer Science, vol. 8761. Springer, 340–354.

CABALAR, P., ODINTSOV, S., AND PEARCE, D. 2006. Strong negation in well-founded and partial stable semantics for logic programs. In *Proc. of the 10th Ibero-American Artificial Intelligence Conf. (IBERAMIA'06), Ribeiro Preto, Brazil*. Lecture Notes in Artificial Intelligence, vol. 4140. 592–601.

FARIÑAS DEL CERRO, L. AND HERZIG, A. 1996. Combining classical and intuitionistic logic, or: Intuitionistic implication as a conditional. In *Proc. of the 1st Intl. Workshop on Frontiers of Combining Systems, Munich, Germany, March 26-29*, F. Baader and K. U. Schulz, Eds. Applied Logic Series, vol. 3. Kluwer Academic Publishers, 93–102.

FARIÑAS DEL CERRO, L., HERZIG, A., AND SU, E. I. 2015. Epistemic equilibrium logic. In *Proc. of the Intl. Joint Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, 2964–2970.

FERRARIS, P. 2005. Answer sets for propositional theories. In *Proc. of the 8th Intl. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR'05)*, C. Baral, G. Greco, N. Leone, and G. Terracina, Eds. Lecture Notes in Computer Science, vol. 3662. Springer, 119–131.

GELFOND, M. AND LIFSCHITZ, V. 1988. The stable model semantics for logic programming. In *Proc. of the 5th Intl. Conference on Logic Programming (ICLP'88)*. 1070–1080.

GELFOND, M. AND LIFSCHITZ, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Comput. 9,* 3/4, 365–386.

GELFOND, M. AND LIFSCHITZ, V. 1993. Representing action and change by logic programs. *Journal of Logic Programming 17,* 2/3&4, 301–321.

HEYTING, A. 1930. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse*, 42–56.

LIFSCHITZ, V., PEARCE, D., AND VALVERDE, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic 2,* 4, 526–541.

LIFSCHITZ, V., TANG, L. R., AND TURNER, H. 1999. Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence 25,* 3–4, 369–389.

MAREK, V. AND TRUSZCZYŃSKI, M. 1999. *Stable models and an alternative logic programming paradigm*. Springer-Verlag, 169–181.

NELSON, D. 1949. Constructible falsity. *Journal of Symbolic Logic 14*, 1626.

NIEMELÄ, I. 1999. Logic programs with stable model semantics as a constraint programming paradigm. *AMAI 25*, 241–273.

ODINTSOV, S. P. AND PEARCE, D. 2005. Routley semantics for answer sets. In *Proc. of the 8th Logic Programming and Nonmonotonic Reasoning, LPNMR 2005, Diamante, Italy*, C. Baral, G. Greco, N. Leone, and G. Terracina, Eds. Lecture Notes in Computer Science, vol. 3662. Springer, 343–355.

ORTIZ, M. AND OSORIO, M. 2007. Strong negation and equivalence in the safe belief semantics. *Journal of Logic and Computation 17*, 499–515.

PEARCE, D. 1997. A new logical characterisation of stable models and answer sets. In *NMELP*. Lecture Notes in Computer Science, vol. 1216. Springer, 57–70.

PEREIRA, L. M. AND ALFERES, J. J. 1992. Well founded semantics for logic programs with explicit negation. In *Proceedings of the European Conference on Artificial Intelligence (ECAI'92)*. John Wiley & Sons, Montreal, Canada, 102–106.

VAKARELOV, D. 1977. Notes on N-lattices and constructive logic with strong negation. *Studia logica 36,* 1-2, 109–125.

VOROB'EV, N. 1952a. A constructive propositional calculus with strong negation (in Russian). *Doklady Akademii Nauk SSR 85*, 465–468.

VOROB'EV, N. 1952b. The problem of deducibility in constructive propositional calculus with strong negation (in russian). *Doklady Akademii Nauk SSR 85*, 689–692.

Appendix A. Proofs

Proof of Proposition 1.

We use structural induction on F. Notice that, when $F = \neg G$ and T is a consistent set of literals, we have both:

$$\begin{array}{rll} T \models \neg G & \text{iff} & T \not\models G \\ & \text{iff} & (\neg G)^T = \top \\ & \text{iff} & T \models (\neg G)^T \end{array} \qquad \begin{array}{rll} T =\!\mid \neg G & \text{iff} & T \models G \\ & \text{iff} & (\neg G)^T = \bot \\ & \text{iff} & T =\!\mid (\neg G)^T \end{array}$$

□

Proof of Theorem 1.

Let At be the set of atoms occurring in φ or in α. Suppose φ is HT valid. Without loss of generality, we can assume that α belongs to signature $At \setminus \{p\}$ because, being φ HT-valid, we can always replace any atom p by a fresh one p' and restate the theorem for the latter. Now, suppose $\varphi[\alpha/p]$ is not $\mathcal{X}_5$ valid. Then, there is some $\mathcal{X}_5$ interpretation $M = \langle H, T\rangle$ for signature $At \setminus \{p\}$ such that $M \not\models \varphi[\alpha/p]$ in $\mathcal{X}_5$. But then, we can construct an HT interpretation $\langle H', T'\rangle$ for signature At that assigns atom p the same behaviour as α. Formally:

$$H' = \begin{cases} H & \text{if } \langle H,T\rangle \not\models \alpha \\ H \cup \{p\} & \text{if } \langle H,T\rangle \models \alpha \end{cases} \qquad T' = \begin{cases} T & \text{if } \langle T,T\rangle \not\models \alpha \\ T \cup \{p\} & \text{if } \langle T,T\rangle \models \alpha \end{cases}$$

for p an atom and α a formula. It can be easily proved by induction on φ that:

$$\langle H,T\rangle \models \varphi[\alpha/p] \text{ iff } \langle H',T'\rangle \models \varphi$$

Since we had $\langle H,T\rangle \not\models \varphi[\alpha/p]$, then $\langle H',T'\rangle \not\models \varphi$, which contradicts that φ is HT valid. □

Proof of Lemma 1.

Use induction on F and definitions of $T \models F$ from Section 2 and $\langle T,T\rangle \models F$ from Definition 3. □

Proof of Theorem 2.

Take $M = \langle H,T\rangle$ an $\mathcal{X}_5$-interpretation and φ a formula. We are going to prove both (i) and (ii) by structural induction on φ.

- If $\varphi = p$ and $\langle H,T\rangle \models p$, then $p \in H \subseteq T$ which implies $\langle T,T\rangle \models p$. In case $\langle H,T\rangle =\!\mid p$, we have that $\sim p \in H \subseteq T$ and $\langle T,T\rangle =\!\mid p$.
- If $\varphi = \alpha \wedge \beta$ and $\langle H,T\rangle \models \varphi$, then both $\langle H,T\rangle \models \alpha$ and $\langle H,T\rangle \models \beta$. By induction, $\langle T,T\rangle \models \alpha$ and $\langle T,T\rangle \models \beta$ or, equivalently $\langle T,T\rangle \models \alpha \wedge \beta$. Now suppose that $\langle H,T\rangle =\!\mid \varphi$, then $\langle H,T\rangle =\!\mid \alpha$ or $\langle H,T\rangle =\!\mid \beta$. By induction, $\langle T,T\rangle =\!\mid \alpha$ or $\langle T,T\rangle =\!\mid \beta$ which means that $\langle T,T\rangle =\!\mid \alpha \wedge \beta$.
- If $\varphi = \alpha \vee \beta$, the proof is similar to the previous case.
- If $\varphi = \sim\alpha$ and $\langle H,T\rangle \models \varphi$, then $\langle H,T\rangle =\!\mid \alpha$ and, by induction $\langle T,T\rangle =\!\mid \alpha$ or $\langle T,T\rangle \models \varphi$. If $\langle H,T\rangle =\!\mid \varphi$, then $\langle H,T\rangle \models \alpha$ and, by induction $\langle T,T\rangle \models \alpha$ or $\langle T,T\rangle =\!\mid \varphi$.
- If $\varphi = \alpha \to \beta$ and $\langle H,T\rangle \models \varphi$, then $[\langle H,T\rangle \not\models \alpha$ or $\langle H,T\rangle \models \beta]$ and $[\langle T,T\rangle \not\models \alpha$ or $\langle T,T\rangle \models \beta]$ which implies that $\langle T,T\rangle \models \varphi$ by definition. Now suppose that $\langle H,T\rangle =\!\mid \varphi$. Then, $\langle T,T\rangle \models \alpha$ and $\langle H,T\rangle =\!\mid \beta$. By induction $\langle T,T\rangle =\!\mid \beta$, so $\langle T,T\rangle =\!\mid \varphi$.

$\square$

Proof of Proposition 2.

By applying Theorem 2, we know that $\langle T,T\rangle \not\models \varphi$ implies $\langle H,T\rangle \not\models \varphi$. This proves that $\langle H,T\rangle \models \neg\varphi : \varphi \to \bot$ iff $\langle T,T\rangle \not\models \varphi$. On the other hand, $\langle H,T\rangle \mathrel{=\!\!|} \neg\varphi$ iff $\langle T,T\rangle \models \varphi$ and $\langle H,T\rangle \mathrel{=\!\!|} \bot$ or, equivalently, $\langle T,T\rangle \models \varphi$. $\square$

Proof of Lemma 2.

We proceed by structural induction on F. If $F = p$, take into account that $p^T = p$. First of all, $\langle H,T\rangle \models p$ iff $p \in H$. On the other hand, $\langle H,T\rangle \mathrel{=\!\!|} p$ iff $H \models\sim p$, or equivalently $H \mathrel{=\!\!|} p$
Now suppose that $F = F_1 \wedge F_2$. Then, it follows:

$$
\begin{aligned}
\langle H,T\rangle \models F \quad & \text{iff} \quad \langle H,T\rangle \models F_1 \text{ and } \langle H,T\rangle \models F_2 \\
& \text{iff } H \models (F_1)^T \text{ and } H \models (F_2)^T \\
& \text{iff } H \models (F_1)^T \wedge (F_2)^T \\
& \text{iff } H \models (F_1 \wedge F_2)^T
\end{aligned}
$$

Also:

$$
\begin{aligned}
\langle H,T\rangle \mathrel{=\!\!|} F \quad & \text{iff} \quad \langle H,T\rangle \mathrel{=\!\!|} F_1 \text{ or } \langle H,T\rangle \mathrel{=\!\!|} F_2 \\
& \text{iff } H \mathrel{=\!\!|} (F_1)^T \text{ or } H \mathrel{=\!\!|} (F_2)^T \\
& \text{iff } H \mathrel{=\!\!|} (F_1)^T \wedge (F_2)^T \\
& \text{iff } H \mathrel{=\!\!|} (F_1 \wedge F_2)^T
\end{aligned}
$$

The proof of the case $F = F_1 \vee F_2$ is similar.
We can also use Lemma 1 and Proposition 2 to prove that

$$
\begin{aligned}
\langle H,T\rangle \models \neg F \ & \text{iff } \langle T,T\rangle \not\models F \\
& \text{iff } T \not\models F \\
& \text{iff } (\neg F)^T = \top \\
& \text{iff } H \models (\neg F)^T
\end{aligned}
$$

and

$$
\begin{aligned}
\langle H,T\rangle \mathrel{=\!\!|} \neg F \ & \text{iff } \langle T,T\rangle \models F \\
& \text{iff } T \models F \\
& \text{iff } (\neg F)^T = \bot \\
& \text{iff } H \mathrel{=\!\!|} (\neg F)^T
\end{aligned}
$$

Finally:

$$
\begin{aligned}
\langle H,T\rangle \models\sim F \quad & \text{iff} \quad \langle H,T\rangle \mathrel{=\!\!|} F \\
& \text{iff } H \mathrel{=\!\!|} F^T \\
& \text{iff } H \models\sim \left(F^T\right) \\
& \text{iff } H \models (\sim F)^T
\end{aligned}
$$

and

$$
\begin{aligned}
\langle H,T\rangle \mathrel{=\!\!|} {\sim}F \quad \text{iff} \quad & \langle H,T\rangle \models F\\
& \text{iff } H \models F^T\\
& \text{iff } H \mathrel{=\!\!|} {\sim}(F^T)\\
& \text{iff } H \mathrel{=\!\!|} ({\sim}F)^T
\end{aligned}
$$

□

Proof of Corollary 1.

Notice that, when F is any nested expression, we have that:

$$\langle T,T\rangle \models F \text{ iff } T \models F \text{ iff } T \models F^T$$

by applying Lemma 1 and Proposition 1. □

Proof of Proposition 3.

Take $r : F \to G$ any rule of Π. First of all, suppose that $\langle H,T\rangle \models r$. By persistence, we know that $\langle T,T\rangle \models r$, so T is a model of r because of Corollary 1. Moreover:

$$
\begin{aligned}
&\langle H,T\rangle \not\models F \text{ or } \langle H,T\rangle \models G, \text{ and}\\
&\langle T,T\rangle \not\models F \text{ or } \langle T,T\rangle \models G
\end{aligned}
$$

By applying Lemma 2, we can say that $H \not\models F^T$ or $H \models G^T$, so $H \models F^T \to G^T$.
For the other direction, if H is a model of $r^T : F^T \to G^T$ and T is a model of r, we have that $H \not\models F^T$ or $H \models G^T$. Using Lemma 2, we can say that $\langle H,T\rangle \not\models F$ or $\langle H,T\rangle \models G$. The fact that T is a model of r allow us to conclude that $\langle H,T\rangle \models r$. □

Proof of Theorem 3.

Suppose that T is an answer set of a program Π. Then $\langle T,T\rangle \models \Pi$ by Corollary 1. If $H \subseteq T$ is such that $\langle H,T\rangle \models \Pi$, we know by Proposition 3 that $H \models \Pi^T$. By minimality of T, we conclude that $H = T$. The proof of the other direction is similar. □

Proof of Theorem 4.

For proving (i), if $T \not\models \varphi$ then both $\langle H,T\rangle \not\models \varphi$ (by persistence) and $\varphi^T_+ = \bot$ (by definition) and the result trivially follows. Similarly, for (ii), when $T \not\mathrel{=\!\!|} \varphi$ both $\langle H,T\rangle \not\mathrel{=\!\!|} \varphi$ (by persistence) and $\varphi^T_- = \top$. So, we assume $T \models \varphi$ for (i) and $T \mathrel{=\!\!|} \varphi$ for (ii). We are going to prove both (i) and (ii) by structural induction on φ.

- If $\varphi = p$ and $T \models p$, we know that $p \in H$ iff $\langle H,T\rangle \models p$ iff $H \models p = p^T$. In case that $T \mathrel{=\!\!|} p$, we have that $\langle H,T\rangle \mathrel{=\!\!|} p$ iff ${\sim}p \in H$ iff $H \mathrel{=\!\!|} p = p^T$.
- Suppose that $\varphi = \alpha \wedge \beta$ and $T \models \varphi$. Then both $T \models \alpha$ and $T \models \beta$. Notice that $\langle H,T\rangle \models \varphi$ iff $\langle H,T\rangle \models \alpha$ and $\langle H,T\rangle \models \beta$. By induction, this is equivalent to $H \models \alpha^T_+$ and $H \models \beta^T_+$, that is, $H \models \varphi^T_+ = \alpha^T_+ \wedge \beta^T_+$. On the other hand, $\langle H,T\rangle \mathrel{=\!\!|} \varphi$ iff $\langle H,T\rangle \mathrel{=\!\!|} \alpha$ or $\langle H,T\rangle \mathrel{=\!\!|} \beta$. By induction, this is equivalent to $H \mathrel{=\!\!|} \alpha^T_-$ or $H \mathrel{=\!\!|} \beta^T_-$, that is, $H \mathrel{=\!\!|} \varphi^T_- = \alpha^T_- \wedge \beta^T_-$.
- If $\varphi = \alpha \vee \beta$, the proof is similar to the previous case.
- If $\varphi = \alpha \to \beta$ and $T \models \varphi$. First of all, suppose that $T \models \alpha$ and $H \models \varphi^T_+ = \neg(\alpha^T_+) \vee \beta^T_+$. We want to prove that $\langle H,T\rangle \models \varphi$. We can consider that $\langle H,T\rangle \models \alpha$, which is the same, by

induction, that $H \models \alpha^T_+$. This implies that $H \models \beta^T_+$. Finally, applying induction and taking into account that $T \models \beta$, we can conclude that $\langle H,T\rangle \models \beta$. For the other direction, we proceed in a similar way.
In the case that $T \not\models \alpha$, we have that $\alpha^T_+ = \bot$. So, for any, $H \subseteq T$, it holds that $\langle H,T\rangle \not\models \alpha$ and $H \models \neg(\alpha^T_+)$.
On the other hand, when $T \,=\!\!\mid\, \varphi$, we know that $T \models \alpha$ and $T \,=\!\!\mid\, \beta$, and we have to show that $H \,=\!\!\mid\, \varphi^T_- = \beta^T_-$ iff $\langle H,T\rangle \,=\!\!\mid\, \varphi$. This latter is equivalent to $\langle H,T\rangle \,=\!\!\mid\, \beta$ or, by induction, to $H \,=\!\!\mid\, \beta^T_-$.

- When $\varphi = \neg\alpha$ and $T \models \varphi$ then $T \not\models \alpha$. This implies two things: first, by Proposition 2 we conclude $\langle H,T\rangle \models \neg\alpha = \varphi$ and second, by persistence, $\langle H,T\rangle \not\models \alpha$. By induction on the latter, $H \not\models \alpha^T_+$ and thus $H \models \neg(\alpha^T_+) = \varphi^T_+$. For proving (ii), if $T \,=\!\!\mid\, \varphi$ then $T \models \alpha$. By Proposition 2, this implies $\langle H,T\rangle \,=\!\!\mid\, \neg\alpha = \varphi$ but we also have $\varphi^T_- = \bot$ and $H \,=\!\!\mid\, \bot$ trivially.
- When $\varphi = \sim\alpha$ and $T \models \varphi$, we know that $T \,=\!\!\mid\, \alpha$. Then, $\langle H,T\rangle \models \varphi$ iff $\langle H,T\rangle \,=\!\!\mid\, \alpha$. Using induction, we can say that this is equivalent to $H \,=\!\!\mid\, \alpha^T_-$ which coincides, by definition, with $H \models \sim(\alpha^T_-) = \varphi^T_+$. In the case that $T \,=\!\!\mid\, \varphi$, we have that $T \models \alpha$. So by induction, we can assure that $\langle H,T\rangle \models \alpha$ iff $H \models \alpha^T_+$. Using that $\langle H,T\rangle \models \alpha$ iff $\langle H,T\rangle \,=\!\!\mid\, \sim\alpha$ and $H \models \alpha^T_+$ iff $H \,=\!\!\mid\, \sim(\alpha^T_+)$, we can conclude the result.

□

Proof of Theorem 5.

- When $\varphi = p$, we know that $\langle H,T\rangle \models p$ iff $p \in H$ or $M(p) = 2$. Moreover $\langle T,T\rangle \models p$ iff $M(p) = 1$ or $M(p) = 2$, that is, $M(p) > 0$. On the other hand, $\langle H,T\rangle \,=\!\!\mid\, p$ iff $\sim p \in H$ iff $M(p) = -2$. Finally, $\langle T,T\rangle \,=\!\!\mid\, p$ iff $\sim p \in T$ iff $M(p) = -1$ or $M(p) = -2$, that is $M(p) < 0$.
- Suppose that $\varphi = \varphi_1 \wedge \varphi_2$. It follows that:

$$\begin{array}{rll} \langle H,T\rangle \models \varphi_1 \wedge \varphi_2 & \text{iff} & \langle H,T\rangle \models \varphi_1 \text{ and } \langle H,T\rangle \models \varphi_2 \\ & \text{iff} & M(\varphi_1) = 2 \text{ and } M(\varphi_2) = 2 \\ & \text{iff} & min(M(\varphi_1), M(\varphi_2)) = 2 \\ & \text{iff} & M(\varphi_1 \wedge \varphi_2) = 2 \end{array}$$

and

$$\begin{array}{rll} \langle H,T\rangle \,=\!\!\mid\, \varphi_1 \wedge \varphi_2 & \text{iff} & \langle H,T\rangle \,=\!\!\mid\, \varphi_1 \text{ or } \langle H,T\rangle \,=\!\!\mid\, \varphi_2 \\ & \text{iff} & M(\varphi_1) = -2 \text{ or } M(\varphi_2) = -2 \\ & \text{iff} & min(M(\varphi_1), M(\varphi_2)) = -2 \\ & \text{iff} & M(\varphi_1 \wedge \varphi_2) = -2 \end{array}$$

We can also prove that:

$$\begin{array}{rll} \langle T,T\rangle \models \varphi_1 \wedge \varphi_2 & \text{iff} & \langle T,T\rangle \models \varphi_1 \text{ and } \langle T,T\rangle \models \varphi_2 \\ & \text{iff} & M(\varphi_1) > 0 \text{ and } M(\varphi_2) > 0 \\ & \text{iff} & min(M(\varphi_1), M(\varphi_2)) > 0 \\ & \text{iff} & M(\varphi_1 \wedge \varphi_2) > 0 \end{array}$$

and

$$\begin{array}{rll} \langle T,T\rangle \models\!\!\!\mid \varphi_1 \wedge \varphi_2 & \text{iff} & \langle T,T\rangle \models\!\!\!\mid \varphi_1 \text{ or } \langle T,T\rangle \models\!\!\!\mid \varphi_2 \\ & \text{iff} & M(\varphi_1) < 0 \text{ or } M(\varphi_2) < 0 \\ & \text{iff} & min(M(\varphi_1), M(\varphi_2)) < 0 \\ & \text{iff} & M(\varphi_1 \wedge \varphi_2) < 0 \end{array}$$

- The proof of the case $\varphi = \varphi_1 \vee \varphi_2$ is similar.
- Suppose that $\varphi = \varphi_1 \to \varphi_2$. It follows that:

$$\langle H,T\rangle \models \varphi_1 \to \varphi_2 \quad \text{iff} \quad \left\{ \begin{array}{c} \langle H,T\rangle \not\models \varphi_1 \text{ or } \langle H,T\rangle \models \varphi_2 \\ \text{and} \\ \langle T,T\rangle \not\models \varphi_1 \text{ or } \langle T,T\rangle \models \varphi_2 \end{array} \right. \quad \text{iff} \quad \left\{ \begin{array}{c} M(\varphi_1) \neq 2 \text{ or } M(\varphi_2) = 2 \\ \text{and} \\ M(\varphi_1) \leq 0 \text{ or } M(\varphi_2) > 0 \end{array} \right.$$

$$\text{iff} \quad \left\{ \begin{array}{c} M(\varphi_1) \leq 0 \\ \text{or} \\ M(\varphi_1) \neq 2 \text{ and } M(\varphi_2) > 0 \\ \text{or} \\ M(\varphi_2) = 2 \end{array} \right. \quad \text{iff} \quad M(\varphi_1 \to \varphi_2) = 2$$

and

$$\begin{array}{rll} \langle H,T\rangle \models\!\!\!\mid \varphi_1 \to \varphi_2 & \text{iff} & \langle T,T\rangle \models \varphi_1 \text{ and } \langle H,T\rangle \models\!\!\!\mid \varphi_2 \\ & \text{iff} & M(\varphi_1) > 0 \text{ and } M(\varphi_2) = -2 \\ & \text{iff} & M(\varphi_1 \to \varphi_2) = -2 \end{array}$$

We can also show that:

$$\begin{array}{rll} \langle T,T\rangle \models \varphi_1 \to \varphi_2 & \text{iff} & \langle T,T\rangle \not\models \varphi_1 \text{ or } \langle T,T\rangle \models \varphi_2 \\ & \text{iff} & M(\varphi_1) \leq 0 \text{ and } M(\varphi_2) > 0 \\ & \text{iff} & M(\varphi_1 \to \varphi_2) > 0 \end{array}$$

and

$$\begin{array}{rll} \langle T,T\rangle \models\!\!\!\mid \varphi_1 \to \varphi_2 & \text{iff} & \langle T,T\rangle \models \varphi_1 \text{ and } \langle T,T\rangle \models\!\!\!\mid \varphi_2 \\ & \text{iff} & M(\varphi_1) > 0 \text{ and } M(\varphi_2) < 0 \\ & \text{iff} & M(\varphi_1 \to \varphi_2) < 0 \end{array}$$

- Since $\langle H,T\rangle \models \sim\varphi$ (resp. $\langle H,T\rangle \models\!\!\!\mid \sim\varphi$) iff $\langle H,T\rangle \models\!\!\!\mid \varphi$ (resp. $\langle H,T\rangle \models \varphi$) and the fact that $M(\sim\varphi) = -M(\varphi)$ complete the proof for $\sim\varphi$.

$\square$

Proof of Proposition 5.

For the left to right direction, if $M \leq M'$, we have $M = \langle H,T\rangle$ and $M' = \langle H',T\rangle$ with $H \subseteq H'$. Then, case 1 holds since, for any atom p, $M(p) = 0$ iff $\{p, \sim p\} \cap T = \emptyset$ iff $M'(p) = 0$. For case 2, $M(p) \in \{1,2\}$. Take $M(p) = 2$, then $p \in H \subseteq H'$ and $M'(p) = 2$. If, instead, $M(p) = 1$, we conclude $p \in T \setminus H$ and, as $p \in T$, it means $M'(p) \geq 1 = M(p)$. Case 3 with $M(p) < 0$ is proved in an analogous way. For the right to left direction, suppose cases 1, 2 and 3 hold and let $M = \langle H,T\rangle$ and $M' = \langle H',T'\rangle$. We prove first that $T \subseteq T'$. For any $p \in T$ we get $0 < M(p) \leq M'(p)$ which means $p \in T'$. Analogously, when $\sim p \in T$ we obtain $0 > M(p) \geq M'(p)$ that implies $\sim p \in T'$. Direction $T' \subseteq T$ is analogous, and we conclude $T = T'$. We prove now that $H \subseteq H'$. Any $p \in H$ satisfies $M(p) = 2$, and so, $M'(p) \geq M(p) = 2$ which implies that $M'(p) = 2$, that is $p \in H'$.

Similarly, for any $\sim p \in H$ we have $M(p) = -2$, so $M'(p) \leq M(p) = -2$ which implies that $M'(p) = -2$, that is $\sim p \in H'$. $\square$

Proof of Proposition 6.

We just prove the left to right direction (the other one is symmetric). Suppose that $M \models \Gamma \cup \{\alpha\}$. Since $M \models \alpha$, we conclude $M(\alpha) = 2$. On the other hand, as $\alpha \leftrightarrow \beta$ is a tautology, $M(\alpha \leftrightarrow \beta) = 2$ and, with $M(\alpha) = 2$ the only possibility is $M(\beta) = 2$, that is, $M \models \beta$. Therefore, we immediately get $M \models \Gamma \cup \{\beta\}$. $\square$

Proof of Theorem 7.

As we saw in Figure 1, $M(\alpha \Leftrightarrow \beta) = 2$ iff $M(\alpha) = M(\beta)$. Since $\models \alpha \Leftrightarrow \beta$, this means that α and β have the same valuation for any M. Since valuation of formulas is compositional, $M(\varphi[\alpha/p]) = M(\varphi[\beta/p])$ and so $M(\varphi[\alpha/p] \Leftrightarrow \varphi[\beta/p]) = 2$ for any M. $\square$

Proof of Theorem 8.

We can suppose that $p \in At(\varphi)$ because, otherwise, $\varphi[\alpha/p] = \varphi[\beta/p] = \varphi$. Since p does not occur in the scope of explicit negation, we can also suppose that $\varphi \neq \sim\psi$. The proof follows by structural induction on φ. We are going to suppose that $\langle H,T\rangle \models \psi[\alpha/p]$ iff $\langle H,T\rangle \models \psi[\beta/p]$ for any subformula ψ of φ and any $\mathcal{X}_5$-interpretation $\langle H,T\rangle$.

- If $\varphi = p$, then $\varphi[\alpha/p] = \alpha$ and $\varphi[\beta/p] = \beta$ and we have $\models \alpha \leftrightarrow \beta$.
- If $\varphi = \varphi_1 \wedge \varphi_2$ and $\langle H,T\rangle \models \varphi[\alpha/p]$, then both $\langle H,T\rangle \models \varphi_1[\alpha/p]$ and $\langle H,T\rangle \models \varphi_2[\alpha/p]$. By induction, $\langle H,T\rangle \models \varphi_i[\beta/p]$, for $i = 1,2$, so $\langle H,T\rangle \models \varphi[\beta/p]$.
- If $\varphi = \varphi_1 \vee \varphi_2$, the proof is similar to the previous case.
- If $\varphi = \varphi_1 \to \varphi_2$ and $\langle H,T\rangle \models \varphi[\alpha/p]$, then $\big[\langle H,T\rangle \not\models \varphi_1[\alpha/p]$ or $\langle H,T\rangle \models \varphi_2[\alpha/p]\big]$ and $\big[\langle T,T\rangle \not\models \varphi_1[\alpha/p]$ or $\langle T,T\rangle \models \varphi_2[\alpha/p]\big]$ which implies, by induction, that $\big[\langle H,T\rangle \not\models \varphi_1[\beta/p]$ or $\langle H,T\rangle \models \varphi_2[\beta/p]\big]$ and $\big[\langle T,T\rangle \not\models \varphi_1[\beta/p]$ or $\langle T,T\rangle \models \varphi_2[\beta/p]\big]$. This means that $\langle H,T\rangle \models \varphi[\beta/p]$.

$\square$

Proof of Lemma 3.

First note that $\langle H,T\rangle \models \alpha$ implies that $\langle H \cap At(\alpha), T \cap At(\alpha)\rangle \models \alpha$ and, since α is finite we have that $At(\alpha)$ is finite. Hence, we assume without loss of generality that $\langle H,T\rangle$ is finite. Let $M = \langle H,T\rangle$ and $M^t = \langle T,T\rangle$. Note that $M \models \alpha$ implies $M^t \models \alpha$ (Theorem 2). In case that $M^t \not\models \beta$, let $\Delta = T$. Then, M^t is an equilibrium model of $\Delta \cup \{\alpha\}$ but not of $\Delta \cup \{\beta\}$. Otherwise, $H \neq T$ and we define:

$$\Delta = H \cup \{\, l_1 \to l_2 \mid l_1, l_2 \in T \setminus H \,\}.$$

We can prove that M is an equilibrium model of $\Delta \cup \{\beta\}$ but not of $\Delta \cup \{\alpha\}$. In fact, if $M' = \langle H',T\rangle$ is a model of $\Delta \cup \{\beta\}$, then $H \subseteq H'$ but $H \neq H'$ because $\langle H,T\rangle \not\models \beta$. If $H' \neq T$, we can find $l_1 \in H' \setminus H$ and $l_2 \in T \setminus H'$. But, then $l_1 \to l_2 \in \Delta$ and $\langle H',T\rangle \not\models l_1 \to l_2$. This proves that M is an equilibrium model of $\Delta \cup \{\beta\}$. The fact that $\langle H,T\rangle \models \Delta \cup \{\alpha\}$ and $H \neq T$ implies that M is not an equilibrium model of $\Delta \cup \{\alpha\}$. $\square$

Proof of Theorem 9.

For the "if" direction, the result is straightforward: when $\models \alpha \leftrightarrow \beta$, α and β have the same models, and so, also $\Delta \cup \{\alpha\}$ and $\Delta \cup \{\beta\}$ have the same models, for any theory Δ, so they also have the same equilibrium models. For the "only if" direction we will proceed by contraposition, i.e., we will prove that $\not\models \alpha \leftrightarrow \beta$ implies that α and β are not strongly equivalent. If $\not\models \alpha \leftrightarrow \beta$ there is some model $\langle H, T \rangle$ of one of the formulas that is not model of the other. Without loss of generality, suppose $\langle H, T \rangle \models \alpha$ and $\langle H, T \rangle \not\models \beta$. Then, by Lemma 3, α and β are not strongly equivalent. $\square$

Proof of Theorem 10.

For the "if" direction, if $\models \alpha \Leftrightarrow \beta$ we conclude $\models \varphi[\alpha/p] \Leftrightarrow \varphi[\beta/p]$ by Theorem 7. But then, $\varphi[\alpha/p]$ and $\varphi[\beta/p]$ share the same models, and thus, $\Delta \cup \{\varphi[\alpha/p]\}$ and $\Delta \cup \{\varphi[\alpha/p]\}$ also have the same models for any Δ, so they also share the same equilibrium models. For the "only if" direction, note that if α and β are strongly equivalent on substitutions, they are also strongly equivalent and, from Theorem 9, this implies that $\models \alpha \leftrightarrow \beta$. Furthermore, if α and β are strongly equivalent on substitutions, we also get that $\sim\alpha$ and $\sim\beta$ are strongly equivalent. From Theorem 9, this implies that $\models \sim\alpha \leftrightarrow \sim\beta$. Hence, we obtain that $\models \alpha \Leftrightarrow \beta$. $\square$