# A Characterization of Strong Equivalence for Logic Programs with Variables

Vladimir Lifschitz[1]*, David Pearce[2]**, and Agustín Valverde[3]***

[1] Department of Computer Sciences,
University of Texas at Austin, USA.
vlcs.utexas.edu
[2] Computing Science and Artificial Intelligence,
Univ. Rey Juan Carlos, (Móstoles, Madrid), Spain.
davidandrew.pearceurjc.es
[3] Dept. of Applied Mathematics, Univ. of Málaga, Spain.
a_valverdectima.uma.es

**Abstract.** Two sets of rules are said to be strongly equivalent to each other if replacing one by the other within any logic program preserves the program's stable models. The familiar characterization of strong equivalence of grounded programs in terms of the propositional logic of here-and-there is extended in this paper to a large class of logic programs with variables. This class includes, in particular, programs with conditional literals and cardinality constraints. The first-order version of the logic of here-and-there required for this purpose involves two additional non-intuitionistic axiom schemas.

## 1 Introduction

The concept of a stable model was originally defined in [6] for sets of rules of a very special syntactic form. Later it was extended to arbitrary propositional formulas [13, 5] and to arbitrary first-order sentences [10, 11, 4]. The extension to formulas with quantifiers is important, in particular, in view of its close relation to conditional literals—an LPARSE construct widely used in answer set programming [14]. For instance, according to [4], the choice rule

$$\{q(x) : p(x)\}$$

can be viewed as shorthand for the first-order formula

$$\forall x(p(x) \to (q(x) \lor \neg q(x))).$$

* Partially supported by the National Science Foundation under Grant IIS-0412907.

** Partially supported by CICyT projects TIC-2003-9001-C02 and TIN2006-15455-CO3.

*** Partially supported by CICyT project TIC-2003-9001-C01, TIN2006-15455-CO1 and Junta de Andalucia project TIC-115

Similarly, the LPARSE rule

$$2\,\{q(x)\ :\ p(x)\}$$

can be thought of as shorthand for the formula

$$\begin{array}{l}\forall x(p(x) \to (q(x) \vee \neg q(x)))\wedge \\ \quad \exists xy(p(x) \wedge q(x) \wedge p(y) \wedge q(y) \wedge x \neq y).\end{array}$$

In this paper we extend the main theorem of [8] to stable models of first-order sentences. That theorem relates strong equivalence of propositional (grounded) logic programs to the propositional logic of here-and-there. Recall that two sets of rules are said to be strongly equivalent to each other if replacing one by the other within any logic program preserves the program's stable models; the propositional logic of here-and-there is the extension of propositional intuitionistic logic obtained by adding the axiom schema

HOS $\quad F \vee (F \to G) \vee \neg G.$

This is a simplified form of an axiom from [7], proposed in [2]. It is weaker than the law of the excluded middle, but stronger than the weak law of the excluded middle

WEM $\neg F \vee \neg\neg F.$

(To derive WEM from HOS, take $G$ to be $\neg F$.)

Such characterizations of strong equivalence are interesting because they tell us which transformations can be used to simplify rules, or groups of rules, in a logic program. For instance, if we replace the pair of rules

$$\begin{array}{l} q \leftarrow \mathit{not}\ p \\ q \leftarrow \{\mathit{not}\ p\}\,0 \end{array}$$

in a logic program with the fact $q$ then the stable models of the program will remain the same. Indeed, the formula

$$(\neg p \to q) \wedge (\neg\neg p \to q)$$

is equivalent to $q$ in the propositional logic of here-and-there. (Proof: use WEM with $p$ as $F$.)

There are several natural extensions of the logic of here-and-there to first-order formulas; all of them include the axioms and inference rules of intuitionistic predicate logic, axiom schema HOS, and some other axioms. Our goal here is to determine which of these extensions corresponds to the strong equivalence of first-order sentences in the sense of [4].

The next section is a review the definition of a stable model from [4]. In Section 3 we state our main theorem, which characterizes strong equivalence in terms of a first-order version of the logic of here-and-there, and give examples of the use of that logic for establishing the strong equivalence of formulas and corresponding programs in the language of LPARSE. Section 4 describes a characterization of our first-order logic of here-and-there in terms of Kripke models; the soundness and completeness theorem stated in that section is a key element of the proof of the main theorem. Proofs are outlined in Sections 5 and 6. In Section 7 the theorem on strong equivalence is extended from formulas to theories—sets of formulas, possibly infinite. Related work is discussed in Section 8.

## 2 Stable Models of a First-Order Sentence

If $p$ and $q$ are predicate constants of the same arity then $p = q$ stands for the formula

$$\forall \mathbf{x}(p(\mathbf{x}) \leftrightarrow q(\mathbf{x})),$$

and $p \leq q$ stands for

$$\forall \mathbf{x}(p(\mathbf{x}) \rightarrow q(\mathbf{x})),$$

where $\mathbf{x}$ is a tuple of distinct object variables. If $\mathbf{p}$ and $\mathbf{q}$ are tuples $p_1, \dots, p_n$ and $q_1, \dots, q_n$ of predicate constants then $\mathbf{p} = \mathbf{q}$ stands for the conjunction

$$p_1 = q_1 \wedge \cdots \wedge p_n = q_n,$$

and $\mathbf{p} \leq \mathbf{q}$ for

$$p_1 \leq q_1 \wedge \cdots \wedge p_n \leq q_n.$$

Finally, $\mathbf{p} < \mathbf{q}$ is an abbreviation for $\mathbf{p} \leq \mathbf{q} \wedge \neg(\mathbf{p} = \mathbf{q})$. In second-order logic, we will apply the same notation to tuples of predicate variables.

According to [4], for any first-order sentence (closed formula) $F$, SM$[F]$ stands for the second-order sentence

$$F \wedge \neg \exists \mathbf{u}((\mathbf{u} < \mathbf{p}) \wedge F^*(\mathbf{u})),$$

where $\mathbf{p}$ is the list of all predicate constants $p_1, \dots, p_n$ occurring in $F$, $\mathbf{u}$ is a list of $n$ distinct predicate variables $u_1, \dots, u_n$, and $F^*(\mathbf{u})$ is defined recursively, as follows:

- $p_i(t_1, \dots, t_m)^* = u_i(t_1, \dots, t_m)$;
- $(t_1 = t_2)^* = (t_1 = t_2)$;
- $\bot^* = \bot$;
- $(F \odot G)^* = F^* \odot G^*$, where $\odot \in \{\wedge, \vee\}$;
- $(F \rightarrow G)^* = (F^* \rightarrow G^*) \wedge (F \rightarrow G)$;
- $(QxF)^* = QxF^*$, where $Q \in \{\forall, \exists\}$.

(There is no clause for negation here, because we treat $\neg F$ as shorthand for $F \rightarrow \bot$.) A model of $F$ is *stable* if it satisfies SM$[F]$.

This definition looks very different from the original definition of a stable model from [6], but it is actually a generalization of that definition, in the following sense. Let $F$ be (the sentence corresponding to) a finite set of rules of the form

$$A_0 \leftarrow A_1, \dots, A_m, \textit{not } A_{m+1}, \dots, \textit{not } A_n,$$

where $A_0, \dots, A_n$ are atomic formulas not containing equality. According to Proposition 1 from [4], the Herbrand stable models of $F$ in the sense of the definition above are identical to the stable models of $F$ in the sense of the original definition. For instance, the sentence

$$p(a) \wedge q(b) \wedge \forall x((p(x) \wedge \neg q(x)) \rightarrow r(x)), \tag{1}$$

representing the logic program

$$\begin{array}{l} p(a), \\ q(b), \\ r(x) \leftarrow p(x), not\ q(x), \end{array} \tag{2}$$

has a unique Herbrand stable model

$$\{p(a), q(b), r(a)\},$$

which is the stable model of (2) in the sense of the 1988 definition.

Here is an example illustrating the relationship between the definition above and the semantics of programs with conditional literals and choice rules proposed in [14]. The sentence

$$p(a) \wedge p(b) \wedge \forall x(p(x) \rightarrow (q(x) \vee \neg q(x))),$$

representing the program

$$\begin{array}{l} p(a), \\ p(b), \\ \{q(x)\ :\ p(x)\}, \end{array} \tag{3}$$

has 4 Herbrand stable models

$$\begin{array}{l} \{p(a),\ p(b)\}, \\ \{p(a),\ p(b),\ q(a)\}, \\ \{p(a),\ p(b),\ q(b)\}, \\ \{p(a),\ p(b),\ q(a),\ q(b)\}, \end{array}$$

which are identical to the stable models of (3) in the sense of [14].

## 3 Theorem on Strong Equivalence

About first-order sentences $F$ and $G$ we say that $F$ is *strongly equivalent* to $G$ if, for every first-order sentence $H$ (possibly of a larger signature), $F \wedge H$ has the same stable models as $G \wedge H$ [4].

By $\mathbf{INT}^=$ we denote first-order intuitionistic logic with the usual axioms for equality:

$$x = x$$

and

$$x = y \rightarrow (F(x) \rightarrow F(y))$$

for every formula $F(x)$ such that $y$ is substitutable for $x$ in $F(x)$.

Our characterization of strong equivalence refers to the axiom schema

SQHT $\quad \exists x(F(x) \rightarrow \forall x F(x))$.

The notation SQHT stands for "static quantified here-and-there"; see Section 4 below for an explanation. We also need the "decidable equality" axiom

DE $\quad x = y \vee x \neq y$.

**Theorem on Strong Equivalence** *A sentence $F$ is strongly equivalent to a sentence $G$ iff the equivalence $F \leftrightarrow G$ is provable in*

$$\mathbf{INT}^{=} + \text{HOS} + \text{SQHT} + \text{DE}. \tag{4}$$

We will denote system (4) by $\mathbf{SQHT}^{=}$.

**Example 1** In any program containing the rules

$$\begin{array}{l} \{q(x)\ :\ p(x)\} \\ p(a) \\ q(a) \end{array}$$

replacing the fact $q(a)$ with the constraint

$$\leftarrow \textit{not}\ q(a)$$

would not change the program's stable models. Indeed, the formula

$$\forall x(p(x) \rightarrow (q(x) \vee \neg q(x))) \wedge p(a) \wedge q(a) \tag{5}$$

is intuitionistically equivalent to

$$\forall x(p(x) \rightarrow (q(x) \vee \neg q(x))) \wedge p(a) \wedge \neg\neg q(a).$$

Proof: The first two conjunctive terms of (5) imply $q(a) \vee \neg q(a)$, and consequently $\neg\neg q(a) \leftrightarrow q(a)$.

Replacing a fact by a constraint can be viewed as a simplification, because the effect of adding a constraint to a program on its stable models is easy to describe. For instance, adding the constraint $\leftarrow \textit{not}\ q(a)$ to a program eliminates its stable models that do not satisfy $q(a)$. Adding the fact $q(a)$ to a program may affect its stable models, generally, in a very complicated way.

**Example 2** Dropping $x \neq y$ from the body of the rule

$$p(y) \leftarrow p(x), q(x,y), x \neq y$$

would not change a program's stable models, because the formula

$$\forall xy(p(x) \wedge q(x,y) \wedge x \neq y \rightarrow p(y)) \tag{6}$$

is equivalent to

$$\forall xy(p(x) \land q(x,y) \rightarrow p(y)) \tag{7}$$

in $\mathbf{INT}^=$ + DE. Proof: By DE, (7) is equivalent to the conjunction of (6) and

$$\forall xy(p(x) \land q(x,y) \land x = y \rightarrow p(y)).$$

The last formula is provable in $\mathbf{INT}^=$.

**Example 3** A different characterization of strong equivalence is used in [4, Section 4] to show that $\neg\forall x F(x)$ is strongly equivalent to $\exists x \neg F(x)$. To prove this fact using the theorem above, observe that the equivalence

$$\neg\forall x F(x) \leftrightarrow \exists x \neg F(x) \tag{8}$$

is provable in **INT** + SQHT. (Proof: the implication right-to-left is provable intuitionistically; the implication left-to-right is an intutionistic consequence of SQHT.) Furthemore, $\neg\neg\exists x F(x)$ is strongly equivalent to $\exists x \neg\neg F(x)$. (Proof: the formula

$$\neg\neg\exists x F(x) \leftrightarrow \exists x \neg\neg F(x). \tag{9}$$

is intuitionistically equivalent to the instance of (8) in which $\neg F(x)$ is taken as $F(x)$.)

## 4 Kripke Models

Our proof of the theorem on strong equivalence refers to the class of Kripke models introduced in [4]. In this section we discuss two reasons why this class of models is relevant. On the one hand, system $\mathbf{SQHT}^=$, introduced above in connection with the problem of strong equivalence, turns out to be a sound and complete axiomatization of this class of models. On the other hand, according to Proposition 4 from [4], the first-order equilibrium logic based on this class of models provides a characterization of the concept of a stable model that we are interested in.

The definition of this class of models uses the following notation. If $I$ is an interpretation of a signature $\sigma$ (in the sense of classical logic) then by $\sigma^I$ we denote the extension of $\sigma$ obtained by adding pairwise distinct symbols $\xi^*$, called *names*, for all elements $\xi$ of the universe of $I$ as object constants. We will identify $I$ with its extension to $\sigma^I$ defined by $I(\xi^*) = \xi$. The value that $I$ assigns to a ground term $t$ of signature $\sigma^I$ will be denoted by $t^I$. By $\sigma_f$ we denote the part of $\sigma$ consisting of its function constants (including object constants, which are viewed as function constants of arity 0).

An *HT-interpretation* of $\sigma$ is a triple $\langle I^{\mathrm{f}}, I^{\mathrm{h}}, I^{\mathrm{t}} \rangle$, where

- $I^{\mathrm{f}}$ is an interpretation of $\sigma_f$, and
- $I^{\mathrm{h}}$, $I^{\mathrm{t}}$ are sets of atomic formulas formed using predicate constants from $\sigma$ and object constants $\xi^*$ for arbitrary elements $\xi$ of the universe of $I^{\mathrm{f}}$, such that $I^{\mathrm{h}} \subseteq I^{\mathrm{t}}$.

The symbols h, t are called *worlds*; they are ordered by the relation h<t.

Note that according to this definition the two worlds share a common universe. In this sense, our Kripke models are static; this explains the use of the word "static" in the name of the axiom SQHT (Section 3). The worlds share also a common equality relation: in both of them, equality is understood as identity.

The *satisfaction* relation between an HT-interpretation $I = \langle I^{\mathrm{f}}, I^{\mathrm{h}}, I^{\mathrm{t}} \rangle$, a world $w$ and a sentence $F$ of the signature $\sigma^U$, where $U$ is the universe of $I^{\mathrm{f}}$, is defined recursively:

- $I, w \models p(t_1, \dots)$ if $p((t_1^I)^*, \dots) \in I^w$,
- $I, w \models t_1 = t_2$ if $t_1^I = t_2^I$,
- $I, w \not\models \bot$,
- $I, w \models F \wedge G$ if $I, w \models F$ and $I, w \models G$,
- $I, w \models F \vee G$ if $I, w \models F$ or $I, w \models G$,
- $I, w \models F \to G$ if, for every world $w'$ such that $w \leq w'$, $I, w' \not\models F$ or $I, w' \models G$,
- $I, w \models \forall x F(x)$ if, for each $\xi$ from the universe of $I^{\mathrm{f}}$, $I, w \models F(\xi^*)$,
- $I, w \models \exists x F(x)$ if, for some $\xi$ from the universe of $I^{\mathrm{f}}$, $I, w \models F(\xi^*)$.

We write $I \models F$ if $I, \mathrm{h} \models F$.[1]

For any set $\Gamma$ of sentences and any sentence $F$, we write $\Gamma \models F$ if every HT-interpretation satisfying all formulas in $\Gamma$ satisfies $F$ also. We write $\Gamma \vdash F$ if $F$ is derivable from $\Gamma$ in $\mathbf{SQHT}^=$.

**Soundness and Completeness Theorem** *$\Gamma \models F$ iff $\Gamma \vdash F$.*

In the next section we outline a proof of the more difficult part of this claim, the implication left-to-right.

The corresponding concept of an equilibrium model is defined as follows. An HT-interpretation $\langle I^{\mathrm{f}}, I^{\mathrm{h}}, I^{\mathrm{t}} \rangle$ is *total* if $I^{\mathrm{h}} = I^{\mathrm{t}}$. A total HT-interpretation $\langle I, J, J \rangle$ is an *equilibrium model* of $F$ if

(i) $\langle I, J, J \rangle \models F$, and
(ii) for any proper subset $J'$ of $J$, $\langle I, J', J \rangle \not\models F$.

We can represent an interpretation $I$ of $\sigma$ in the sense of classical logic as the pair $\langle I|_{\sigma^{\mathrm{f}}}, I' \rangle$, where $I'$ is the set of all atomic formulas, formed using predicate constants from $\sigma$ and names $\xi^*$, which are satisfied by $I$. According to Proposition 4 from [4], an interpretation $\langle I, J \rangle$ is a stable model of a sentence $F$ iff $\langle I, J, J \rangle$ is an equilibrium model of $F$. Thus stable models of a sentence are essentially identical to its equilibrium models.

[1] This definition looks different from the definition of satisfaction proposed in [4], but it easy to check that they are equivalent to each other.

## 5 Proof of Completeness

Assume that $\Gamma \nvdash F$. We will define an HT-interpretation $I$ that satisfies all formulas in $\Gamma$ but does not satisfy $F$.

There exists a signature $\sigma'$, obtained from $\sigma$ by adding new object constants, and a set $\Gamma_{\mathrm{h}}$ of sentences of this signature, satisfying the following conditions:

(i) $\Gamma \subseteq \Gamma_{\mathrm{h}}$,
(ii) $F \notin \Gamma_{\mathrm{h}}$,
(iii) $\Gamma_{\mathrm{h}}$ is closed under $\vdash$,
(iv) for any sentence of the form $G \vee H$ in $\Gamma_{\mathrm{h}}$, $G \in \Gamma_{\mathrm{h}}$ or $H \in \Gamma_{\mathrm{h}}$,
(v) for any sentence of the form $\exists x F(x)$ in $\Gamma_{\mathrm{h}}$ there exists an object constant $c$ in $\sigma'$ such that $F(c) \in \Gamma_{\mathrm{h}}$.

(Conditions (iii)–(v) can be expressed by saying that $\Gamma_{\mathrm{h}}$ is *prime*.) The proof is the same as in Henkin's proof of completeness of intuitionistic logic [1, Section 3]. For any ground terms $t_1$ and $t_2$ of the signature $\sigma'$, we write $t_1 \approx t_2$ if the formula $t_1 = t_2$ belongs to $\Gamma_{\mathrm{h}}$. Let $\Gamma_{\mathrm{t}}$ be a maximal superset of $\Gamma_{\mathrm{h}}$ that is consistent and closed under classical logic; such a superset exists by the Lindenbaum lemma. Now $I = \langle I^{\mathrm{f}}, I^{\mathrm{h}}, I^{\mathrm{t}} \rangle$ is defined as follows:

(i) the universe of $I^{\mathrm{f}}$ is the set of equivalence classes of the relation $\approx$;
(ii) for each object constant $c$ from $\sigma$, $I^{\mathrm{f}}[c]$ is the equivalence class of $\approx$ that contains $c$,
(iii) for each function constant $g$ from $\sigma$ of arity $n > 0$, $I^{\mathrm{f}}[g](\xi_1, \dots, \xi_n)$ is the equivalence class of $\approx$ that contains $g(t_1, \dots, t_n)$ for all terms $t_1 \in \xi_1, \dots, t_n \in \xi_n$.
(iv) for each world $w$, $I^w$ is the set of formulas of the form $p(\xi_1^*, \dots, \xi_n^*)$ such that $\Gamma_w$ contains $p(t_1, \dots, t_n)$ for all terms $t_1 \in \xi_1, \dots, t_n \in \xi_n$.

Note that $I$ can be viewed as an HT-interpretation of the extended signature $\sigma'$ if we extend clause (ii) to all objects constants from $\sigma'$. In the rest of this section, terms and formulas are understand as terms and formulas of the extended signature.

**Lemma 1** *For any ground terms $t_1$, $t_2$, $(t_1 = t_2) \in \Gamma_{\mathrm{t}}$ iff $(t_1 = t_2) \in \Gamma_{\mathrm{h}}$.*

**Proof** The if part follows from the fact that $\Gamma_{\mathrm{h}} \subseteq \Gamma_{\mathrm{t}}$. Only if: Assume that $(t_1 = t_2) \notin \Gamma_{\mathrm{h}}$. Since $\Gamma_{\mathrm{h}}$ contains the instance $t_1 = t_2 \vee t_1 \neq t_2$ of DE and is prime, it follows that $(t_1 \neq t_2) \in \Gamma_{\mathrm{h}}$. Since $\Gamma_{\mathrm{t}}$ is a consistent superset of $\Gamma_{\mathrm{h}}$, we can conclude that $(t_1 = t_2) \notin \Gamma_{\mathrm{t}}$.

**Lemma 2** *For any sentence of the form $\exists x G(x)$ there exists an object constant $c$ such that the formula*

$$\exists x G(x) \rightarrow G(c) \tag{10}$$

*belongs to $\Gamma_{\mathrm{t}}$.*

**Proof** *Case 1:* $\exists x G(x) \in \Gamma_{\mathrm{t}}$. Since $\Gamma_{\mathrm{h}}$ contains the instance

$$\neg \exists x G(x) \vee \neg\neg \exists x G(x)$$

of WEM and is prime, $\Gamma_{\mathrm{h}}$ contains one of its disjunctive terms. But the first disjunctive term cannot belong to $\Gamma_{\mathrm{h}}$ because the consistent superset $\Gamma_{\mathrm{t}}$ of $\Gamma_{\mathrm{h}}$ contains $\exists x G(x)$. Consequently $\neg\neg\exists x G(x) \in \Gamma_{\mathrm{h}}$. Since equivalence (9) belongs to $\Gamma_{\mathrm{h}}$, it follows that $\exists x \neg\neg G(x) \in \Gamma_{\mathrm{h}}$. Since $\Gamma_{\mathrm{h}}$ is prime, it follows that there exists an object constant $c$ such that $\neg\neg G(c) \in \Gamma_{\mathrm{h}}$. Since $\Gamma_{\mathrm{h}} \subseteq \Gamma_{\mathrm{t}}$ and (10) is a classical consequence of $\neg\neg G(c)$, it follows that (10) belongs to $\Gamma_{\mathrm{t}}$. *Case 2:* $\exists x G(x) \notin \Gamma_t$. Since $\Gamma_{\mathrm{t}}$ is maximally consistent, it follows that $\neg\exists x G(x) \in \Gamma_{\mathrm{t}}$. Since (10) is a classical consequence of $\neg\exists x G(x)$, it follows that (10) belongs to $\Gamma_{\mathrm{t}}$.

Recall that our goal is to prove two properties of the interpretation $I$: it satisfies all formulas in $\Gamma$ but does not satisfy $F$. By the choice of $\Gamma_{\mathrm{h}}$, this is immediate from the following lemma:

**Lemma 3** *For any sentence $G$ of signature $\sigma'$ and any world $w$,*

$$I, w \models G \ \textit{ iff } \ G \in \Gamma_w.$$

**Proof:** by induction on $G$. We will consider the three cases where reasoning is different than in the similar proof for intuitionistic logic [1, Section 3]: $t_1 = t_2$, $G \rightarrow H$, and $\forall x G(x)$.

**1.** To check that

$$I, w \models t_1 = t_2 \text{ iff } t_1 = t_2 \in \Gamma_w$$

we show that each side is equivalent to $t_1 \approx t_2$. For the left-hand side, this follows from the fact that for every ground term $t$, $t^{I_f}$ is the equivalence class of $\approx$ that contains $t$ (by induction on $t$). For the right-hand side, if $w = \mathrm{h}$ then this is immediate from the definition of $\approx$; if $w = \mathrm{t}$ then use Lemma 1.

**2.** Assume that

$$I, w \models G \ \text{ iff } \ G \in \Gamma_w$$

and

$$I, w \models H \ \text{ iff } \ H \in \Gamma_w;$$

we want to show that

$$I, w \models G \rightarrow H \ \text{ iff } \ G \rightarrow H \in \Gamma_w.$$

The if part follows from the induction hypothesis and the clause for $\rightarrow$ in the definition of satisfaction. To prove the only if part for $w = \mathrm{t}$, use the induction hypothesis and the fact that, by the maximal consistency of $\Gamma_{\mathrm{t}}$, this set contains either $G$ or $\neg G$. For $w = \mathrm{h}$, we conclude from the induction hypothesis and the assumption $I, \mathrm{h} \models G \rightarrow H$ that

$$G \notin \Gamma_{\mathrm{h}} \text{ or } H \in \Gamma_{\mathrm{h}} \tag{11}$$

and

$$G \notin \Gamma_{\mathrm{t}} \text{ or } H \in \Gamma_{\mathrm{t}}. \tag{12}$$

*Case 1:* $G \in \Gamma_{\mathrm{h}}$. Then, by (11), $H \in \Gamma_{\mathrm{h}}$ and consequently $G \to H \in \Gamma_{\mathrm{h}}$. *Case 2:* $\neg G \in \Gamma_{\mathrm{h}}$. Since $\neg G \vdash G \to H$, $G \to H \in \Gamma_{\mathrm{h}}$. *Case 3:* $G, \neg G \notin \Gamma_{\mathrm{h}}$. Since $\Gamma_{\mathrm{h}}$ contains the instance $\neg G \vee \neg\neg G$ of WEM and is prime, it follows that $\neg\neg G \in \Gamma_{\mathrm{h}} \subseteq \Gamma_{\mathrm{t}}$. Then $G \in \Gamma_{\mathrm{t}}$ and, by (12), $H \in \Gamma_{\mathrm{t}}$. Since $\Gamma_{\mathrm{t}}$ is consistent and contains $\Gamma_{\mathrm{h}}$, $\neg H \notin \Gamma_{\mathrm{h}}$. On the other hand, $\Gamma_{\mathrm{h}}$ contains the instance $G \vee (G \to H) \vee \neg H$ of HOS and is prime; since neither $G$ nor $\neg H$ belongs to $\Gamma_{\mathrm{h}}$, $G \to H \in \Gamma_{\mathrm{h}}$.

**3.** Assume that for every object constant $c$

$$I, w \models G(c) \text{ iff } G(c) \in \Gamma_w;$$

we need to show that

$$I, w \models \forall x G(x) \text{ iff } \forall x G(x) \in \Gamma_w.$$

The if part follows from the induction hypothesis and the clause for $\forall$ in the definition of satisfaction. To prove the only if part for $w = \mathrm{t}$, take an object constant $c$ such that the formula

$$\exists x \neg G(x) \to \neg G(c) \tag{13}$$

belongs to $\Gamma_{\mathrm{t}}$ (Lemma 2). It follows from the induction hypothesis and the clause for $\forall$ in the definition of satisfaction that $G(c)$ belongs to $\Gamma_{\mathrm{t}}$ too; $\forall x G(x)$ is a classical consequence of (13) and $G(c)$. To prove the only if part for $w = \mathrm{h}$, consider the instance

$$\exists x (G(x) \to \forall x G(x))$$

of SQHT. Since $\Gamma_{\mathrm{h}}$ is prime, there exists an object constant $c$ such that $\Gamma_{\mathrm{h}}$ contains the formula

$$G(c) \to \forall x G(x). \tag{14}$$

By the induction hypothesis and the clause for $\forall$ in the definition of satisfaction, $G(c) \in \Gamma_{\mathrm{h}}$; $\forall x G(x)$ is an intuitionistic consequence of $G(c)$ and (14).

## 6 Proof of the Strong Equivalence Theorem

In view of the soundness and completeness theorem, the theorem on strong equivalence can be rewritten as: a sentence $F$ is strongly equivalent to a sentence $G$ iff $F$ and $G$ are satisfied by the same HT-interpretations.

If $F$ and $G$ are satisfied by the same HT-interpretations then $F \wedge H$ and $G \wedge H$ are satisfied by the same HT-interpretations; then $F \wedge H$ and $G \wedge H$ have the same equilibrium models, and consequently the same stable models.

For the converse, let us assume that $F \wedge H$ and $G \wedge H$ have the same stable models for every first order sentence $H$. Then these formulas have the same equilibrium models. For any predicate constant $P$, let $C(P)$ stand for the sentence

$$\forall \mathbf{x} (\neg\neg P(\mathbf{x}) \to P(\mathbf{x})).$$

Note first that $F$ and $G$ have the same total models. Indeed, let $H_0$ be the conjunction of sentences $C(P)$ for all predicate constants $P$ occurring in $F$ or $G$; the total models of $F$ can be characterized as the equilibrium models of $F \wedge H_0$, and the total models of $G$ can be characterized as the equilibrium models of $G \wedge H_0$.

Assume that $\langle I^{\mathrm{f}}, I^{\mathrm{h}}, I^{\mathrm{t}}\rangle$ satisfies $F$ but not $G$, and consider the total HT-interpretation $\langle I^{\mathrm{f}}, I^{\mathrm{t}}, I^{\mathrm{t}}\rangle$. It is clear that the total interpretation $\langle I^{\mathrm{f}}, I^{\mathrm{t}}, I^{\mathrm{t}}\rangle$ satisfies $F$, and consequently $G$. Let $H_0$ be the conjunction of sentences $C(P)$ for all predicate constants $P$ occurring in $G$, and let $H_1$ be the implication $G \rightarrow H_0$. The HT-interpretation $\langle I^{\mathrm{f}}, I^{\mathrm{h}}, I^{\mathrm{t}}\rangle$ satisfies $H_1$. Indeed, it does not satisfy its antecedent $G$, and $I^t$ satisfies its consequent $H_0$. Therefore $\langle I^{\mathrm{f}}, I^{\mathrm{t}}, I^{\mathrm{t}}\rangle$ is not an equilibrium model of $F \wedge H_1$. Then $\langle I^{\mathrm{f}}, I^{\mathrm{t}}, I^{\mathrm{t}}\rangle$ is not an equilibrium model of $G \wedge H_1$ either. But this is impossible, because $G \wedge H_1$ implies $H_0$, so that all models of that set are total.

## 7 Strong Equivalence for Theories

The definition of a stable model from [4], reproduced here in Section 2, can be extended to finite sets of first-order sentences: a model of such a set $\Gamma$ is called stable if it satisfies $\mathrm{SM}\left[\bigwedge_{F\in\Gamma} F\right]$. The relationship between stable models and equilibrium models, discussed at the end of Section 4, suggests a way to further extend this definition to "theories"—arbitrary sets of first-order sentences, possibly infinite. We say that a total HT-interpretation $\langle I, J, J\rangle$ is an *equilibrium model* of a set $\Gamma$ of first order sentences if

(i) $\langle I, J, J\rangle \models \Gamma$, and
(ii) for any proper subset $J'$ of $J$, $\langle I, J', J\rangle \not\models \Gamma$.

An interpretation $\langle I, J\rangle$ is a *stable model* of $\Gamma$ iff $\langle I, J, J\rangle$ is an equilibrium model of $\Gamma$.

About sets $\Gamma$, $\Delta$ of first-order sentences we say that $\Gamma$ is *strongly equivalent* to $\Delta$ if, for any set $\Sigma$ of first-order sentences (possibly of a larger signature), $\Gamma \cup \Sigma$ has the same stable models as $\Delta \cup \Sigma$. This relation between sets of formulas can be characterized in the same way as the strong equivalence relation between formulas:

**Theorem on Strong Equivalence for Theories** *A set $\Gamma$ of sentences is strongly equivalent to a set $\Delta$ of sentences iff $\Gamma$ is equivalent to $\Delta$ in* $\mathbf{SQHT}^{=}$.

This theorem, in combination with the theorem on strong equivalence for formulas, shows that the new definition of strong equivalence is a generalization of the definition from Section 3: $F$ and $G$ are strongly equivalent to each other as sentences iff $\{F\}$ and $\{G\}$ are strongly equivalent to each other as theories.

The proof is similar to that in the previous section, but considering the (possibly infinite) set $\Sigma_0$ of formulas $C(P)$ instead of the conjunction $H_0$ of these formulas, and the set of implications $G \rightarrow C(P)$ instead of $H_1$.

## 8 Related Work

An alternative proof that the logic $\mathbf{SQHT}^=$ captures strong equivalence for theories in the sense of the previous section can be found in [12]. The two proofs highlight different properties. In each case it is shown that when two theories $\Gamma$ and $\Delta$ are not equivalent in $\mathbf{SQHT}^=$, an extension $\Sigma$ can be constructed such that the equilibrium models of $\Gamma \cup \Sigma$ and $\Delta \cup \Sigma$ differ. From the proof given in Section 7 above it is clear that $\Sigma$ can be constructed in the same signature, without additional constants. From the proof given in [12] the extension may use object constants not appearing in $\Gamma$ or $\Delta$. However, $\Sigma$ is shown there to have a very simple form: its elements are "unary" formulas, that is, ground atomic formulas and implications $F \to G$ where $F$ and $G$ are ground atomic formulas. From this observation it follows that if $\Gamma$ and $\Delta$ consist of logic program rules and are not strongly equivalent then there exists a set of program rules $\Sigma$ (of a simple form) such that $\Gamma \cup \Sigma$ and $\Delta \cup \Sigma$ have different equilibrium or stable models.

Strong equivalence for non-ground logic programs under the answer set semantics has also been defined and studied in [9, 3]. In the case of [3] the concept is similar to the one presented in the previous section, except that the equivalence is defined with respect to a somewhat different notion of stable model than the one used here, and equality is not explicitly treated. In general the two concepts are different since stable or equilibrium models as defined here are not required to satisfy the unique name assumption. As a consequence not every equilibrium model need be a stable model or answer set in the sense of [3]. However for the safe programs without equality studied in [3] we can establish a simple characterisation of strong equivalence. Let us denote by **SQHT** the logic $\mathbf{SQHT}^=$ without an equality predicate and axioms for equality:

$$\mathbf{SQHT} = \mathbf{INT} + \mathrm{HOS} + \mathrm{SQHT} + \mathrm{DE}.$$

Disjunctive logic programs are defined in the usual way, and rules where each variable appears in at least one positive body atom are called *safe*; a program is *safe* if all its rules are safe. According to a theorem from [12], two safe disjunctive programs are strongly equivalent in the sense of [3] if and only if they are equivalent in the logic **SQHT**. The proof makes use of the fact that if $\Gamma$ and $\Delta$ are non-strongly equivalent safe programs then there exists a set of $\Sigma$ of unary program rules such that $\Gamma \cup \Sigma$ and $\Delta \cup \Sigma$, which are both safe, have different stable models. For safe programs, this theorem also encompasses the notion of strong equivalence found in [9].

## 9 Conclusion

In this paper we understand logic programs with variables in a very general way, as arbitrary first-order formulas with equality. The logic $\mathbf{SQHT}^=$ is the first-order version of the logic of here-and-there that characterizes strong equivalence for such programs. This logic is an extension of the intuitionistic first-order logic with equality. One of its three additional postulates is the axiom schema HOS, familiar from the propositional logic of here-and-there. Another is the well-known decidable equality axiom DE. The third, SQHT, is apparently new; it can be thought of as the result the first step towards converting the trivial implication $\forall x F(x) \rightarrow \forall x F(x)$ to prenex form. Studying properties of this intermediate logic is a topic for future work.

## References

1. Dirk van Dalen. Intuitionistic logic. In Dov Gabbay and Franz Guenther, editors, *Handbook of Philosophical Logic, Volume III: Alternatives in Classical Logic*, Dordrecht, 1986. D. Reidel Publishing Co.
2. Dick De Jongh and Lex Hendriks. Characterization of strongly equivalent logic programs in intermediate logics. *Theory and Practice of Logic Programming*, 3:250–270, 2003.
3. T. Eiter, M. Fink, H. Tompits, and S. Woltran Strong and Uniform Equivalence in Answer-Set Programming: Characterizations and Complexity Results for the Non-Ground Case. KR 2005, AAAI, 2005.
4. Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz. A new perspective on stable models. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 372–379, 2007.
5. Paolo Ferraris. Answer sets for propositional theories. In *Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pages 119–131, 2005.
6. Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Robert Kowalski and Kenneth Bowen, editors, *Proceedings of International Logic Programming Conference and Symposium*, pages 1070–1080, 1988.
7. T. Hosoi. The axiomatization of the intermediate propositional systems $s_n$ of gödel. *Journal of the Faculty of Science of the University of Tokyo*, 13:183–187, 1996.
8. Vladimir Lifschitz, David Pearce, and Agustin Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2:526–541, 2001.
9. F. Lin. Reducing Strong Equivalence of Logic Programs to Entailment in Classical Propositional Logic. In *Proc. KR'02*, 170-176.
10. David Pearce and Agustin Valverde. Towards a first order equilibrium logic for nonmonotonic reasoning. In *Proceedings of European Conference on Logics in Artificial Intelligence (JELIA)*, pages 147–160, 2004.
11. David Pearce and Agustin Valverde. A first order nonmonotonic extension of constructive logic. *Studia Logica*, 80:323–348, 2005.
12. David Pearce and Agustin Valverde. Quantified Equilibrium Logic and the First Order Logic of Here-and-There. Technical Report, Univ. Rey Juan Carlos, 2006, available at http://www.satd.uma.es/matap/investig/tr/ma06_02.pdf.
13. David Pearce. A new logical characterization of stable models and answer sets. In Jürgen Dix, Luis Pereira, and Teodor Przymusinski, editors, *Non-Monotonic Extensions of Logic Programming (Lecture Notes in Artificial Intelligence 1216)*, pages 57–70. Springer-Verlag, 1997.
14. Patrik Simons, Ilkka Niemelä, and Timo Soininen. Extending and implementing the stable model semantics. *Artificial Intelligence*, 138:181–234, 2002.