A Functional Account of Strong Negation in Answer Set Programming

Michael Bartholomew and Joohyung Lee

School of Computing, Informatics and Decision Systems Engineering
Arizona State University, Tempe, USA

Abstract. We present an alternative account of strong negation in answer set programming by identifying literals possibly containing strong negation with Boolean functions under the recent stable model semantics proposed by Bartholomew and Lee. Under a complete interpretation, we show that minimizing both positive and negative literals in the traditional answer set semantics has the same effect as ensuring the uniqueness of function values under the recent stable model semantics. Our work clarifies the relation among strong negation, default negation, and choice rules, and provides insights into the relation between different versions of the stable model semantics. Based on the study we present a new, simple method of representing transition systems in answer set programming. We also relate the functional stable model semantics to Lifschitz's two-valued logic programs, viewing the latter as a special case of the former.

1 Introduction

The distinction between default negation and strong negation has been useful in answer set programming. For instance, the interplay between the two different negations leads to an elegant solution to the frame problem. The fact that block b stays in the same location l by inertia can be described by the rule

$$\mathit{On}(b,l,t{+}1) \;\leftarrow\; \mathit{On}(b,l,t),\; \mathit{not}\;\; {\sim}\mathit{On}(b,l,t+1) \tag{1}$$

along with the rule that describes the uniqueness of location values [Lifschitz, 2002]

$$\sim\!\mathit{On}(b,l_1,t) \;\leftarrow\; \mathit{On}(b,l,t),\; l \neq l_1 \;. \tag{2}$$

Here '$\sim$' is the symbol for strong negation that represents explicit falsity while '*not*' is the symbol for negation as failure (default negation). Rule (1) states that without explicit evidence to the contrary, block b remains in location l. If we are given explicit conflicting information about the location of b at time $t{+}1$ then this conclusion will be defeated by rule (2), which asserts the uniqueness of location.

An alternative representation of inertia, which does not involve strong negation, was presented by Bartholomew and Lee [2012]. They replace rule (1) by a choice rule

$$\{\mathit{On}(b,l,t{+}1)\} \;\leftarrow\; \mathit{On}(b,l,t)\;, \tag{3}$$

which states that "if b is in l at time t, then decide arbitrarily whether to assert that b is in l at time $t{+}1$," and replace rule (2) by constraints (rules with the empty head) which describe that every block is in only one location.

$$\begin{array}{l} \leftarrow\ 2\{\mathit{On}(b,l,t) : \mathit{Location}(l)\} \\ \leftarrow\ \{\mathit{On}(b,l,t) : \mathit{Location}(l)\}0\ . \end{array} \tag{4}$$

In the absence of additional information about the location of block b at time $t{+}1$, asserting $\mathit{On}(b,l,t{+}1)$ is the only option, in view of the existence of location constraint. But if we are given conflicting information about the location of b at time $t+1$ then not asserting $\mathit{On}(b,l,t{+}1)$ is the only option, in view of the uniqueness of location constraint.

The presence of rules (4) ensures that *On* is a function that maps blocks b to locations l. Thus rule (3) essentially represents the inertia involving function values. In fact, Bartholomew and Lee's representation above was obtained from the following "choice" rule that describes defaults involving function values:

$$\{\mathit{Loc}(b,t{+}1) = l\}\ \leftarrow\ \mathit{Loc}(b,t) = l\ . \tag{5}$$

The two solutions to the frame problem do not result in the same answer sets. In the solution using strong negation, an answer set contains only one atom of the form $\mathit{On}(b,l,t)$ for each block b and each time t; for all other location l', negative literals $\sim\mathit{On}(b,l',t)$ belong to the answer set. On the other hand, such negative literals do not occur in answer sets of a program following the solution using choice rules. This observation is related to the symmetric and the asymmetric views of predicates that Lifschitz described in his message to Texas Action Group [1]:

> The way I see it, in ASP programs we use predicates of two kinds, let's call them "symmetric" and "asymmetric." The fact that an object a does not have a property p is reflected by the presence of $\sim p(a)$ in the answer set if p is "symmetric," and by the absence of $p(a)$ if p is "asymmetric." In the second case, the [strong] negation of p is not used in the program at all.

Another instance of this discussion is the related to the closed world assumption. For instance, in program

$$\begin{array}{lll} \mathit{Person}(\mathit{john}) & \mathit{Person}(\mathit{james}) & \mathit{Professor}(\mathit{james}) \\ \sim\mathit{Professor}(x)\ \leftarrow\ \mathit{not}\ \mathit{Professor}(x) & & \end{array} \tag{6}$$

the last rule represents the explicit closed world assumption, which completes the description. The rule allows us to derive $\sim\mathit{Professor}(\mathit{john})$ under the symmetric view. On the other hand, we can view the program without the last rule as a complete description under the implicit closed world assumption. In this case we understand the absence of $\mathit{Professor}(\mathit{john})$ in the answer set as the falsity of the atom.

In this paper, we present an alternative account of strong negation by identifying literals possibly containing strong negation with Boolean functions under the recent functional stable model semantics by Bartholomew and Lee [2012]. Under a complete interpretation, we show that minimizing both positive and negative literals in the traditional answer set semantics has the same effect as ensuring the uniqueness of function values under the recent stable model semantics, which provides insights into the relationship between the different versions of the stable model semantics. Related to this, we show conditions under which

[1] http://www.cs.utexas.edu/users/vl/tag/choice_discussion

strong negation can be replaced by default negation and choice rules, which tells us when the symmetric and the asymmetric views are interchangeable. As a consequence of this, we present a new, simple representation of transition systems in answer set programming that does not involve strong negation.

Related to the topics of this paper, Lifschitz [2012] recently proposed "two-valued logic programs," which modifies the traditional stable model semantics so that it can represent complete information without distinguishing between strong negation and default negation. We show that two-valued logic programs are in fact a special case of multi-valued propositional formulas under the stable model semantics.

A long version with complete proofs is available at http://peace.eas.asu.edu/joolee/papers/sneg-long.pdf.

2 Preliminaries

2.1 Review: First-Order Stable Model Semantics and Strong Negation

This review follows [Ferraris *et al.*, 2011]. A *signature* is defined as in first-order logic. It consists of *function constants* and *predicate constants*. Function constants of arity 0 are also called *object constants*. We assume the following set of primitive propositional connectives and quantifiers: $\bot$ (falsity), $\land$, $\lor$, $\rightarrow$, $\forall$, $\exists$. The syntax of a formula is as defined as in first-order logic. We understand $\neg F$ as an abbreviation of $F \rightarrow \bot$; symbol $\top$ stands for $\bot \rightarrow \bot$, and $F \leftrightarrow G$ stands for $(F \rightarrow G) \land (G \rightarrow F)$.

The stable models of a sentence F relative to a list of predicates $\mathbf{p} = (p_1, \dots, p_n)$ are defined via the *stable model operator with the intensional predicates* $\mathbf{p}$, denoted by SM$[F; \mathbf{p}]$. Let $\mathbf{u}$ be a list of distinct predicate variables $u_1, \dots, u_n$ of the same length as $\mathbf{p}$. By $\mathbf{u} = \mathbf{p}$ we denote the conjunction of the formulas $\forall \mathbf{x}(u_i(\mathbf{x}) \leftrightarrow p_i(\mathbf{x}))$, where $\mathbf{x}$ is a list of distinct object variables of the same length as the arity of p_i, for all $i = 1, \dots, n$. By $\mathbf{u} \leq \mathbf{p}$ we denote the conjunction of the formulas $\forall \mathbf{x}(u_i(\mathbf{x}) \rightarrow p_i(\mathbf{x}))$ for all $i = 1, \dots, n$, and $\mathbf{u} < \mathbf{p}$ stands for $(\mathbf{u} \leq \mathbf{p}) \land \neg(\mathbf{u} = \mathbf{p})$. For any first-order sentence F, expression SM$[F; \mathbf{p}]$ stands for the second-order sentence

$$F \land \neg \exists \mathbf{u}((\mathbf{u} < \mathbf{p}) \land F^*(\mathbf{u})), \tag{7}$$

where $F^*(\mathbf{u})$ is defined recursively:

- $p_i(\mathbf{t})^* = u_i(\mathbf{t})$ for any list $\mathbf{t}$ of terms;
- $F^* = F$ for any atomic formula F (including $\bot$ and equality) that does not contain members of $\mathbf{p}$;
- $(F \land G)^* = F^* \land G^*$; $(F \lor G)^* = F^* \lor G^*$;
- $(F \rightarrow G)^* = (F^* \rightarrow G^*) \land (F \rightarrow G)$;
- $(\forall x F)^* = \forall x F^*$; $(\exists x F)^* = \exists x F^*$.

A model of a sentence F (in the sense of first-order logic) is called $\mathbf{p}$-*stable* if it satisfies SM$[F; \mathbf{p}]$. We will often simply write SM$[F]$ instead of SM$[F; \mathbf{p}]$ when $\mathbf{p}$ is the list of all predicate constants occurring in F, and call a model of SM$[F]$ simply a *stable* model of F. The definition of a stable model is straightforwardly extended to the case when F is a many-sorted first-order formula.

The traditional stable models of a logic program Π are identical to the Herbrand stable models of the *FOL-representation* of Π (i.e., the conjunction of the universal closures of implications corresponding to the rules).

Ferraris *et al.* [2011] incorporate strong negation into the stable model semantics by distinguishing between intensional predicates of two kinds, *positive* and *negative*. Each negative intensional predicate has the form $\sim\!p$, where p is a positive intensional predicate and '$\sim$' is a symbol for strong negation. Syntactically $\sim$is not a logical connective, as it can appear only as a part of a predicate constant. An interpretation of the underlying signature is *coherent* if it satisfies the formula $\neg\exists\mathbf{x}(p(\mathbf{x}) \land \sim\!p(\mathbf{x}))$, where $\mathbf{x}$ is a list of distinct object variables, for each negative predicate $\sim\!p$. We consider coherent interpretations only.

Example 1 *The following is a representation of the Blocks World in the syntax of logic programs:*

$$\begin{array}{rcll}
\bot & \leftarrow & \mathit{On}(b_1,b,t), \mathit{On}(b_2,b,t) & (b_1 \neq b_2) \\
\mathit{On}(b,l,t+1) & \leftarrow & \mathit{Move}(b,l,t) & \\
\bot & \leftarrow & \mathit{Move}(b,l,t), \mathit{On}(b_1,b,t) & \\
\bot & \leftarrow & \mathit{Move}(b,b_1,t), \mathit{Move}(b_1,l,t) & \\
\mathit{On}(b,l,0) & \leftarrow & \mathit{not}\ \sim\!\mathit{On}(b,l,0) & \\
\sim\!\mathit{On}(b,l,0) & \leftarrow & \mathit{not}\ \mathit{On}(b,l,0) & \\
\mathit{Move}(b,l,t) & \leftarrow & \mathit{not}\ \sim\!\mathit{Move}(b,l,t) & \\
\sim\!\mathit{Move}(b,l,t) & \leftarrow & \mathit{not}\ \mathit{Move}(b,l,t) & \\
\mathit{On}(b,l,t+1) & \leftarrow & \mathit{On}(b,l,t), \mathit{not}\ \sim\!\mathit{On}(b,l,t+1) & \\
\sim\!\mathit{On}(b,l,t) & \leftarrow & \mathit{On}(b,l_1,t) & (l \neq l_1)
\end{array} \qquad (8)$$

where On and Move are predicate constants, b, b_1, b_2 are variables ranging over the blocks, l, l_1 are variables ranging over the locations (blocks and the table), and t is a variable ranging over the timepoints. The first rule states that only one block can be on a block. The next three rules describe the effect and preconditions of action Move. The next four rules describe that the initial value of On is exogenous, and action Move is exogenous at each time. The next rule describes the inertia, and the last rule asserts that a block can be in only one location.

2.2 Review: Stable Models of Multi-Valued Propositional Formulas

The following is a review of the stable model semantics of multi-valued propositional formula from [Bartholomew and Lee, 2012], which is a special case of the semantics that we review in the next section.

A *(multi-valued propositional) signature* is a set σ of symbols called *constants*, along with a nonempty finite set $\mathit{Dom}(c)$ of symbols, disjoint from σ, assigned to each constant c. We call $\mathit{Dom}(c)$ the *domain* of c. A *Boolean* constant is one whose domain is the set $\{\text{TRUE}, \text{FALSE}\}$. An *atom* of a signature σ is an expression of the form $c\!=\!v$ ("the value of c is v") where $c \in \sigma$ and $v \in \mathit{Dom}(c)$. A *multi-valued propositional formula* of σ is a propositional combination of atoms.

A *(multi-valued propositional) interpretation* of σ is a function that maps every element of σ to an element of its domain. An interpretation I *satisfies* an atom $c\!=\!v$ (symbolically, $I \models c\!=\!v$) if $I(c) = v$. The satisfaction relation is extended from atoms to arbitrary formulas according to the usual truth tables for the propositional connectives.

The reduct F^I of a multi-valued propositional formula F relative to a multi-valued propositional interpretation I is the formula obtained from F by replacing each maximal subformula that is not satisfied by I with $\bot$. Interpretation I is a *stable model* of F if I is the unique model of F^I.

Example 2 *Consider a multi-valued propositional signature $\sigma = \{ClrBlue, ClrRed, TapeClr\}$, where $Dom(ClrBlue) = Dom(ClrRed) = \{\text{TRUE}, \text{FALSE}\}$ and $Dom(TapeClr) = \{Red, Blue, Green\}$. The following is a multi-valued propositional formula F:*

$$(ClrBlue = \text{TRUE} \lor ClrBlue = \text{FALSE}) \land\ (ClrRed = \text{TRUE} \lor ClrRed = \text{FALSE})$$
$$\land\ (ClrBlue = \text{TRUE} \rightarrow TapeClr = Blue) \land\ (ClrRed = \text{TRUE} \rightarrow TapeClr = Red)\ .$$

Consider an interpretation I such that $I(ClrBlue) = \text{FALSE}$, $I(ClrRed) = \text{TRUE}$ and $I(TapeClr) = Red$. The reduct F^I is

$$(\bot \lor ClrBlue{=}\text{FALSE}) \land (ClrRed{=}\text{TRUE} \lor \bot)$$
$$\land\ (\bot \rightarrow \bot) \land (ClrRed{=}\text{TRUE} \rightarrow TapeClr{=}Red),$$

and I is the only interpretation of σ that satisfies F^I.

2.3 Review: Functional Stable Model Semantics

For predicate symbols (constants or variables) u and c, we define $u \leq c$ as $\forall \mathbf{x}(u(\mathbf{x}) \rightarrow c(\mathbf{x}))$. We define $u = c$ as $\forall \mathbf{x}(u(\mathbf{x}) \leftrightarrow c(\mathbf{x}))$ if u and c are predicate symbols, and $\forall \mathbf{x}(u(\mathbf{x}) = c(\mathbf{x}))$ if they are function symbols.

Let $\mathbf{c}$ be a list of distinct predicate and function constants and let $\widehat{\mathbf{c}}$ be a list of distinct predicate and function variables corresponding to $\mathbf{c}$. We call members of $\mathbf{c}$ *intensional* constants. By $\mathbf{c}^{pred}$ we mean the list of the predicate constants in $\mathbf{c}$, and by $\widehat{\mathbf{c}}^{pred}$ the list of the corresponding predicate variables in $\widehat{\mathbf{c}}$. We define $\widehat{\mathbf{c}} < \mathbf{c}$ as $(\widehat{\mathbf{c}}^{pred} \leq \mathbf{c}^{pred}) \land \neg(\widehat{\mathbf{c}} = \mathbf{c})$ and $\text{SM}[F;\ \mathbf{c}]$ as

$$F \land \neg \exists \widehat{\mathbf{c}}(\widehat{\mathbf{c}} < \mathbf{c} \land F^*(\widehat{\mathbf{c}})),$$

where $F^*(\widehat{\mathbf{c}})$ is defined the same as the one in Section 2.1 except for the base case:

- When F is an atomic formula, F^* is $F' \land F$, where F' is obtained from F by replacing all intensional (function and predicate) constants in it with the corresponding (function and predicate) variables.[2]

If $\mathbf{c}$ contains predicate constants only, this definition of a stable model reduces to the one in [Ferraris *et al.*, 2011], also reviewed in Section 2.1.

We abbreviate the formula $F \lor \neg F$ ("the law of excluded middle") as $\{F\}$. A formula $\{\mathbf{t} = \mathbf{t}'\}$, where $\mathbf{t}$ contains an intensional function constant and $\mathbf{t}'$ does not, represents that $\mathbf{t}$ takes the value $\mathbf{t}'$ by default. For example, the f-stable models of $\{f = 1\}$ maps f to 1. On the other hand, the default is defeated when we conjoin the formula with $(f = 2)$: the f-stable models of $\{f = 1\} \land (f = 2)$ maps f to 2, not to 1.

[2] If an atomic formula F contains no intensional function constants, then F^* can be defined as F', as in [Ferraris *et al.*, 2011].

Example 3 *The Blocks World can be described in this language as follows. For readability, we write in a logic program like syntax:*

$$
\begin{array}{rcl}
\bot & \leftarrow & \mathit{Loc}(b_1,t)\!=\!b \wedge \mathit{Loc}(b_2,t)\!=\!b \wedge (b_1 \neq b_2)\\
\mathit{Loc}(b,t{+}1)\!=\!l & \leftarrow & \mathit{Move}(b,l,t)\\
\bot & \leftarrow & \mathit{Move}(b,l,t) \wedge \mathit{Loc}(b_1,t)\!=\!b\\
\bot & \leftarrow & \mathit{Move}(b,b_1,t) \wedge \mathit{Move}(b_1,l,t)\\
\{\mathit{Loc}(b,0)\!=\!l\} & & \\
\{\mathit{Move}(b,l,t)\} & & \\
\{\mathit{Loc}(b,t{+}1)\!=\!l\} & \leftarrow & \mathit{Loc}(b,t)\!=\!l
\end{array}
$$

where Loc is a function constant. The last rule is a default formula that describes the commonsense law of inertia. The stable models of this program are the models of $\mathrm{SM}[F;\ \mathit{Loc},\mathit{Move}]$*, where* F *is the FOL-representation of the program.*

3 Representing Strong Negation by Boolean Functions

3.1 Representing Strong Negation in Multi-Valued Propositional Formulas

Given a traditional propositional logic program Π of a signature σ, we identify σ with the multi-valued propositional signature whose constants are the same symbols from σ and every constant is Boolean. By Π^{mv} we mean the multi-valued propositional formula that is obtained from Π by replacing negative literals of the form $\sim p$ with $p = \text{FALSE}$ and positive literals of the form p with $p = \text{TRUE}$.

We say that a set X of literals from σ is *complete* if, for each atom $a \in \sigma$, either a or $\sim a$ is in X. We identify a complete set of literals from σ with the corresponding multi-valued propositional interpretation.

Theorem 1 *A complete set of literals is an answer set of* Π *in the sense of [Gelfond and Lifschitz, 1991] iff it is a stable model of* Π^{mv} *in the sense of [Bartholomew and Lee, 2012].*

The theorem tells us that checking the minimality of positive and negative literals under the traditional stable model semantics is essentially the same as checking the uniqueness of corresponding function values under the stable model semantics from [Bartholomew and Lee, 2012].

Example 4 *Consider a simple transition system consisting of two states depending on whether fluent p is true or false, and an action that makes p true.*

$$
\begin{array}{rclcrcl}
p_0 & \leftarrow & \mathit{not}\ \sim\!p_0 & \quad & p_1 & \leftarrow & a\\
\sim\!p_0 & \leftarrow & \mathit{not}\ p_0 & & & & \\
 & & & & p_1 & \leftarrow & p_0, \mathit{not}\ \sim\!p_1\\
a & \leftarrow & \mathit{not}\ \sim\!a & & \sim\!p_1 & \leftarrow & \sim\!p_0, \mathit{not}\ p_1\ .\\
\sim\!a & \leftarrow & \mathit{not}\ a & & & &
\end{array}
\tag{9}
$$

The program has four answer sets, each of which corresponds to the edges of the transition system. For instance, $\{\sim\!p_0, a, p_1\}$ *is an answer set. This program can be encoded in the input language of* CLINGO *or* DLV*. In the input language of a system like* DLV *that allows disjunctions in the head, the first four rules can be succinctly replaced by*

$$p_0 \vee {\sim}p_0 \qquad\qquad a \vee {\sim}a\ .$$

According to Theorem 1, the stable models of this program are the same as the stable models of the following multi-valued propositional formula (written in a logic program style; '$\neg$' represents default negation):

$$\begin{array}{rcl@{\qquad}rcl}
p_0 = \mathrm{TRUE} & \leftarrow & \neg(p_0 = \mathrm{FALSE}) & p_1 = \mathrm{TRUE} & \leftarrow & a = \mathrm{TRUE} \\
p_0 = \mathrm{FALSE} & \leftarrow & \neg(p_0 = \mathrm{TRUE}) & & & \\
 & & & p_1 = \mathrm{TRUE} & \leftarrow & p_0 = \mathrm{TRUE} \wedge \neg(p_1 = \mathrm{FALSE}) \\
a = \mathrm{TRUE} & \leftarrow & \neg(a = \mathrm{FALSE}) & p_1 = \mathrm{FALSE} & \leftarrow & p_0 = \mathrm{FALSE} \wedge \neg(p_1 = \mathrm{TRUE}) \\
a = \mathrm{FALSE} & \leftarrow & \neg(a = \mathrm{TRUE}) & & &
\end{array}$$

3.2 Relation among Strong Negation, Default Negation, Choice Rules and Boolean Functions

The following theorem presents conditions under which equivalent transformations in classical logic preserve stable models.

Theorem 2 *Let F be a sentence, let* $\mathbf{c}$ *be a list of predicate and function constants, and let I be a (coherent) interpretation. Let F' be a formula obtained from F by replacing a subformula $\neg H$ with $\neg H'$ such that $I \models \widetilde{\forall}(H \leftrightarrow H')$. Then*

$$I \models \mathrm{SM}[F;\mathbf{c}] \text{ iff } I \models \mathrm{SM}[F';\mathbf{c}]\ .$$

In particular, the theorem allows us to exchange default negation and strong negation occurring in H if we consider complete interpretations only.

Example 4 continued *Each answer set of the first program in Example 4 is complete. In view of Theorem 2, the first two rules can be rewritten as $p_0 \leftarrow$ not not p_0 and ${\sim}p_0 \leftarrow$ not not ${\sim}p_0$, which can be further abbreviated as choice rules $\{p_0\}$ and $\{{\sim}p_0\}$. Consequently, the whole program can be rewritten using choice rules as*

$$\begin{array}{c@{\qquad\qquad}rcl}
\{p_0\} & p_1 & \leftarrow & a \\
\{{\sim}p_0\} & & & \\
 & \{p_1\} & \leftarrow & p_0 \\
\{a\} & \{{\sim}p_1\} & \leftarrow & {\sim}p_0 \\
\{{\sim}a\} & & &
\end{array}$$

Similarly, in view of Theorem 2, the first rule of the second program in Example 4 can be rewritten as $p_0 = \mathrm{TRUE} \leftarrow \neg\neg(p_0 = \mathrm{TRUE})$ and further as $\{p_0 = \mathrm{TRUE}\}$. This transformation allows us to rewrite the whole program using choice rules as

$$\begin{array}{c@{\qquad\qquad}rcl}
\{p_0 = B\} & p_1 = \mathrm{TRUE} & \leftarrow & a = \mathrm{TRUE} \\
\{a = B\} & \{p_1 = B\} & \leftarrow & p_0 = B
\end{array}$$

where B belongs to $\{\mathrm{TRUE}, \mathrm{FALSE}\}$.

3.3 Representing Strong Negation by Boolean Functions in the First-Order Case

Theorem 1 can be extended to the first-order case as follows.

Let f be a function constant. A first-order formula is called *f-plain* if each atomic formula

- does not contain f, or
- is of the form $f(\mathbf{t}) = u$ where $\mathbf{t}$ is a tuple of terms not containing f, and u is a term not containing f.

For a list $\mathbf{c}$ of predicate and function constants, we say that F is $\mathbf{c}$-plain if F is f-plain for each function constant f in $\mathbf{c}$.

Let F be a formula with strong negation. Formula $F^{\sim pp}_{b}$ is obtained from F as follows:

- in the signature of F, replace p and $\sim p$ with a new intensional function constant b of arity n, where n is the arity of p (or $\sim p$), and add two non-intensional object constants TRUE and FALSE;
- replace every occurrence of $\sim p(\mathbf{t})$, where $\mathbf{t}$ is a list of terms, with $b(\mathbf{t}) = \text{FALSE}$, and then replace every occurrence of $p(\mathbf{t})$ with $b(\mathbf{t}) = \text{TRUE}$.

By *FC_b* ("Functional Constraint on b") we denote the conjunction of the following formulas, which enforces b to behave like predicates:

$$\text{FALSE} \neq \text{TRUE}, \tag{10}$$

$$\neg\neg\forall\mathbf{x}(b(\mathbf{x}) = \text{FALSE} \lor b(\mathbf{x}) = \text{TRUE}). \tag{11}$$

where $\mathbf{x}$ is a list of distinct object variables.

Theorem 3 *Let $\mathbf{c}$ be a set of predicate and function constants, and let F be a $\mathbf{c}$-plain formula. Formulas*

$$\forall\mathbf{x}((p(\mathbf{x}) \leftrightarrow b(\mathbf{x}){=}\text{TRUE}) \land (\sim p(\mathbf{x}) \leftrightarrow b(\mathbf{x}){=}\text{FALSE})), \tag{12}$$

and FC_b entail

$$\text{SM}[F;\ \sim p\, p\, \mathbf{c}] \leftrightarrow \text{SM}[F^{\sim pp}_{b};\ b\, \mathbf{c}]\ .$$

If we drop the requirement that F be $\mathbf{c}$-plain, the result does not hold as in the following example.

Example 5 *Take $\mathbf{c}$ as fg and let F be $p(f)\land \sim p(g)$. Consider the interpretation I whose universe is $\{1, 2\}$ such that I contains $p(1), \sim p(2)$ and with the mappings $b^I(1) = \text{TRUE}, b^I(2) = \text{FALSE}, f^I = 1, g^I = 2$. I certainly satisfies FC_b and (12). I also satisfies $\text{SM}[F;\ \sim ppfg]$ but it does not satisfy $\text{SM}[F^{p\sim p}_{b};\ bfg]$; we can take $\widehat{b}^I(1) = \text{FALSE}, \widehat{b}^I(2) = \text{TRUE}, \widehat{f}^I = 2, \widehat{g}^I = 1$ that satisfies $\widehat{b}\widehat{f}\widehat{g} < bfg$ and $(F^{p\sim p}_{b})^*(\widehat{b}\widehat{f}\widehat{g})$ which is*

$$b(f) = \text{TRUE} \land \widehat{b}(\widehat{f}) = \text{TRUE} \land b(g) = \text{FALSE} \land \widehat{b}(\widehat{g}) = \text{FALSE}.$$

We say that an interpretation is *complete* on a predicate p if it satisfies

$$\forall\mathbf{x}(p(\mathbf{x})\lor \sim p(\mathbf{x}))\ .$$

Note that any interpretation that satisfies both (12) and FC_b is complete on p. Theorem 3 tells us that for any interpretation I that is complete on p, minimizing the extents of both p and $\sim p$ has the same effect as ensuring the corresponding Boolean function b to have a unique value.

The following corollary shows that there is a 1–1 correspondence between the stable models of F and the stable models of $F^{\sim pp}_{b}$. For any interpretation I of the signature of F, by $I^{\sim pp}_{b}$ we denote the interpretation of the signature of $F^{\sim pp}_{b}$ obtained from I by replacing the relation p^I with function b^I such that

$$\begin{array}{l} b^I(\xi_1,\dots,\xi_n) = \text{TRUE}^I \;\text{ if }\; p^I(\xi_1,\dots,\xi_n) = \text{TRUE} \\ b^I(\xi_1,\dots,\xi_n) = \text{FALSE}^I \;\text{ if }\; (\sim p)^I(\xi_1,\dots,\xi_n) = \text{TRUE}\ . \end{array}$$

(Note that we overload the symbols TRUE and FALSE.) We also require that $I^{\sim pp}_{b}$ satisfy (10). Consequently, $I^{\sim pp}_{b}$ satisfies FC_b. Since I is complete on p and coherent, b^I is well-defined.

Corollary 1 *Let* $\mathbf{c}$ *be a set of predicate and function constants, and let* F *be a* $\mathbf{c}$*-plain sentence. (a) An interpretation* I *of the signature of* F *that is complete on* p*,* I *is a model of* $\text{SM}[F;\ \sim p\ p\ \mathbf{c}]$ *iff* $I^{\sim pp}_{b}$ *is a model of* $\text{SM}[F^{\sim pp}_{b} \wedge FC_b;\ b\mathbf{c}]$. *(b) An interpretation* J *of the signature of* $F^{\sim pp}_{b}$ *is a model of* $\text{SM}[F^{\sim pp}_{b} \wedge FC_b;\ b\ \mathbf{c}]$ *iff* $J = I^{\sim pp}_{b}$ *for some model* I *of* $\text{SM}[F; \sim p\ p\ \mathbf{c}]$.

4 Representing NonBoolean Functions Using Strong Negation

In this section, we show how to eliminate nonBoolean intensional functions in favor of Boolean intensional functions. Combined with the method in the previous section, it gives us a systematic method of representing nonBoolean functions using strong negation.

4.1 Eliminating nonBoolean Functions in favor of Boolean Functions

Let F be an f-plain formula. Formula F^f_b is obtained from F as follows:

- in the signature of F, replace f with a new boolean intensional function b of arity $n+1$ where n is the arity of f;
- replace each subformula $f(\mathbf{t}) = c$ with $b(\mathbf{t}, c) = \text{TRUE}$.

By UE_b, we denote the following formulas that preserve the functional behavior:

$$\forall \mathbf{x} yz(y \neq z \wedge b(\mathbf{x}, y) = \text{TRUE} \rightarrow b(\mathbf{x}, z) = \text{FALSE}), \tag{13}$$

$$\neg\neg\forall \mathbf{x} \exists y(b(\mathbf{x}, y) = \text{TRUE}), \tag{14}$$

where $\mathbf{x}$ is a n-tuple of variables and all variables in $\mathbf{x}$, y, and z are pairwise distinct.

Theorem 4 *For any* f*-plain formula* F*,*

$$\forall \mathbf{x} y\big((f(\mathbf{x}) = y \leftrightarrow b(\mathbf{x}, y) = \text{TRUE}) \wedge (f(\mathbf{x}) \neq y \leftrightarrow b(\mathbf{x}, y) = \text{FALSE})\big)$$

and $\exists xy(x \neq y)$ *entail*

$$\text{SM}[F; f\mathbf{c}] \;\leftrightarrow\; \text{SM}[F^f_b \wedge UE_b; b\mathbf{c}]\ .$$

By I^f_b, we denote the interpretation of the signature of F^f_b obtained from I by replacing the function f^I with the function b^I such that

$$\begin{array}{ll} b^I(\xi_1,\dots,\xi_n,\xi_{n+1}) = \text{TRUE}^I & \text{if } f^I(\xi_1,\dots,\xi_n) = \xi_{n+1} \\ b^I(\xi_1,\dots,\xi_n,\xi_{n+1}) = \text{FALSE}^I & \text{otherwise.} \end{array}$$

for all $\xi_1,\dots,\xi_n,\xi_{n+1}$ from the universe of I.

Corollary 2 *Let F be an f-plain sentence. (a) An interpretation I of the signature of F that satisfies $\exists xy(x \neq y)$ is a model of* $\text{SM}[F; f\mathbf{c}]$ *iff I^f_b is a model of* $\text{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$. *(b) An interpretation J of the signature of F^f_b that satisfies $\exists xy(x \neq y)$ is a model of* $\text{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$ *iff $J = I^f_b$ for some model I of* $\text{SM}[F; f\mathbf{c}]$.

Example 3 continued *In the program in Example 3, we eliminate nonBoolean function Loc in favor of Boolean function On as follows.*

$$\begin{array}{rcl} \bot & \leftarrow & On(b_1,b,t) = \text{TRUE} \wedge On(b_2,b,t) = \text{TRUE} \wedge b_1 \neq b_2 \\ On(b,l,t+1) = \text{TRUE} & \leftarrow & Move(b,l,t) \\ \bot & \leftarrow & Move(b,l,t) \wedge On(b_1,b,t) = \text{TRUE} \\ \bot & \leftarrow & Move(b,b_1,t) \wedge Move(b_1,l,t) \\ \{On(b,l,0) = \text{TRUE}\} & & \\ \{Move(b,l,t)\} & & \\ \{On(b,l,t+1) = \text{TRUE}\} & \leftarrow & On(b,l,t) = \text{TRUE} \\ On(b,l,t) = \text{FALSE} & \leftarrow & On(b,l_1,t) = \text{TRUE} \wedge l \neq l_1 \ . \end{array}$$

This program can be further turned into a usual logic program by applying Corollary 1.

$$\begin{array}{rcll} \bot & \leftarrow & On(b_1,b,t), On(b_2,b,t) & (b_1 \neq b_2) \\ On(b,l,t+1) & \leftarrow & Move(b,l,t) & \\ \bot & \leftarrow & Move(b,l,t), On(b_1,b,t) & \\ \bot & \leftarrow & Move(b,b_1,t), Move(b_1,l,t) & \\ \{On(b,l,0)\} & & & \\ \{Move(b,l,t)\} & & & \\ \{On(b,l,t+1)\} & \leftarrow & On(b,l,t) & \\ {\sim}On(b,l,t) & \leftarrow & On(b,l_1,t) & (l \neq l_1) \end{array} \tag{15}$$

Let us compare this program with program (8). *Similar to the explanation in Example 4 (continued), the 5th and the 7th rules of* (8) *can be represented using choice rules, which are the same as the 5th and the 6th rules of* (15). *The 6th and the 7th rules of* (8) *is the closed world assumption. We can check that adding these rules to* (15) *extends the answer sets of* (8) *in a conservative way with the definition of the negative literals.*

In general, nonBoolean functions can be represented using strong negation by composing the two methods, first eliminating nonBoolean functions in favor of Boolean functions as in Corollary 2 and then eliminating Boolean functions in favor of predicates as in Corollary 1. In the following we state this translation.

Let F be an f-plain formula where f is an intensional function constant. Formula F^f_p is obtained from F as follows:

- in the signature of F, replace f with two new intensional predicates p and ${\sim}p$ of arity $n+1$ where n is the arity of f;

- replace each subformula $f(\mathbf{t}) = c$ with $p(\mathbf{t}, c)$.

By UE_p, we denote the following formulas that preserve the functional behavior:

$$\forall \mathbf{x}yz(y \neq z \wedge p(\mathbf{x}, y) \rightarrow {\sim}p(\mathbf{x}, z)) \ ,$$

$$\neg\neg\forall \mathbf{x}\exists y(p(\mathbf{x}, y)) \ ,$$

where $\mathbf{x}$ is an n-tuple of variables and all variables in $\mathbf{x}, y, z$ are pairwise distinct.

Theorem 5 *For any $f\mathbf{c}$-plain formula F, formulas*

$$\forall \mathbf{x}y(f(\mathbf{x}) = y \leftrightarrow p(\mathbf{x}, y)), \ \ \forall \mathbf{x}y(f(\mathbf{x}) \neq y \leftrightarrow {\sim}p(\mathbf{x}, y)), \ \ \exists xy(x \neq y)$$

entail

$$\mathrm{SM}[F; f\mathbf{c}] \leftrightarrow \mathrm{SM}[F^f_p \wedge \mathit{UE}_p; {\sim}pp\mathbf{c}] \ .$$

By $I^f_{\sim pp}$, we denote the interpretation of the signature of $F^f_{\sim pp}$ obtained from I by replacing the function f^I with the set p^I that consists of the tuples

$$\langle \xi_1, \dots, \xi_n, f^I(\xi_1, \dots, \xi_n) \rangle$$

for all $\xi_1, \dots, \xi_n$ from the universe of I. We then also add the set $({\sim}p)^I$ that consists of the tuples

$$\langle \xi_1, \dots, \xi_n, \xi_{n+1} \rangle$$

for all $\xi_1, \dots, \xi_n, \xi_{n+1}$ from the universe of I that do not occur in the set p^I.

Corollary 3 *Let F be an $f\mathbf{c}$-plain sentence. (a) An interpretation I of the signature of F that satisfies $\exists xy(x \neq y)$ is a model of* $\mathrm{SM}[F; f\mathbf{c}]$ *iff $I^f_{\sim pp}$ is a model of* $\mathrm{SM}[F^f_p \wedge \mathit{UE}_p; {\sim}pp\mathbf{c}]$. *(b) An interpretation J of the signature of F^f_p that satisfies $\exists xy(x \neq y)$ is a model of* $\mathrm{SM}[F^f_p \wedge \mathit{UE}_p; {\sim}pp\mathbf{c}]$ *iff $J = I^f_{\sim pp}$ for some model I of* $\mathrm{SM}[F; f\mathbf{c}]$.

Theorem 5 and Corollary 3 are similar to Theorem 8 and Corollary 2 from [Bartholomew and Lee, 2012]. The main difference is that the latter statements refer to a constraint called UEC_p that is weaker than UE_p. For instance, the elimination method from [Bartholomew and Lee, 2012] turns the Blocks World in Example 3 into an almost the same program as (15) except that the last rule is turned into a constraint UEC_{On}:

$$\leftarrow \ \mathit{On}(b, l, t) \wedge \mathit{On}(b, l_1, t) \wedge l \neq l_1 \ . \tag{16}$$

It is clear that the stable models of $F^f_p \wedge \mathit{UE}_p$ are under the symmetric view, and the stable models of $F^f_p \wedge \mathit{UEC}_p$ are under the asymmetric view. To see how replacing UE_{On} by UEC_{On} turns the symmetric view to the asymmetric view, first observe that adding (16) to program (15) does not affect the stable models of the program. Let's call this program Π. It is easy to see that Π is a conservative extension of the program that is obtained from Π by deleting the rule with ${\sim}\mathit{On}(b, l, t)$ in the head.

5 Relating to Lifschitz's Two-Valued Logic Programs

Lifschitz [2012] presented a high level logic program that does not contain explicit default negation, but can handle nonmonotonic reasoning in a similar style as in default logic.

In this section we show how his formalism can be viewed as a special case of multi-valued propositional formulas under the stable model semantics in which every function is considered Boolean.

5.1 Review: Two-Valued Logic Programs

Let σ be a signature in propositional logic. A *two-valued rule* is an expression of the form

$$L_0 \leftarrow L_1, \dots, L_n : F \tag{17}$$

where $L_0, \dots, L_n$ are propositional literals formed from σ and F is a propositional formula of signature σ.

A *two-valued program Π* is a set of two-valued rules. An interpretation I is a function from σ to $\{\text{TRUE}, \text{FALSE}\}$. The *reduct* of a program Π relative to an interpretation I, denoted Π^I, is the set of rules $L_0 \leftarrow L_1, \dots, L_n$ corresponding to the rules (17) of Π for which $I \models F$. I is a stable model of Π if it is a minimal model of Π^I.

Example 6

$$a \leftarrow\ : a, \qquad \neg a \leftarrow : \neg a, \qquad b \leftarrow a : \top \tag{18}$$

The reduct of this program relative to $\{a, b\}$ consists of rules a and $b \leftarrow a$. Interpretation $\{a, b\}$ is the minimal model of the reduct, so that it is a stable model of the program.

As described in [Lifschitz, 2012], if F in every rule (17) has the form of conjunctions of literals, then two-valued logic program can be turned into a program with strong negation, where we consider only complete answer sets. For instance, program (18) can be turned into

$$a \leftarrow \textit{not} \sim\!a, \qquad \sim\!a \leftarrow \textit{not}\ a, \qquad b \leftarrow a\ .$$

This program has two answer sets, $\{a, b\}$ and $\sim\!a$, and only the complete answer set $\{a, b\}$ corresponds to the stable model found in Example 6.

5.2 Translation into SM with Boolean Functions

Given a two-valued logic program Π of signature σ, we define the transformation $\textit{tv2sm}(\Pi)$ as the conjunction of

$$\neg\neg \textit{Tr}(F) \wedge \textit{Tr}(L_1) \wedge \dots \wedge \textit{Tr}(L_n) \rightarrow \textit{Tr}(L_0)$$

for each rule (17) in Π, where *Tr* is defined as follows.

- For any literal L,

$$\textit{Tr}(L) = \begin{cases} A = \text{TRUE} & \text{if } L \text{ is a positive literal } A \\ A = \text{FALSE} & \text{if } L \text{ is a negative literal } \sim\!A \end{cases}$$

- For any formula F, $\textit{Tr}(\neg F)$ is defined as $\neg \textit{Tr}(F)$.
- For any formulas F, G and any $\odot \in \{\wedge, \vee, \rightarrow\}$, $\textit{Tr}(F \odot G)$ is defined as $\textit{Tr}(F) \odot \textit{Tr}(G)$.

Given a two-valued logic program Π of signature σ, we identify σ with the multi-valued propositional signature whose constants are from σ and the domain of every constant is Boolean values $\{\text{TRUE}, \text{FALSE}\}$. For any interpretation I of σ, we obtain the multi-valued interpretation I' from I by the following mapping. For each atom A in σ,

$$I'(A) = \begin{cases} \text{TRUE} & \text{if } I \models A \\ \text{FALSE} & \text{if } I \models \neg A \end{cases}$$

Theorem 6 *For any two-valued program Π of signature σ, an interpretation I is a stable model of Π in the sense of Lifschitz iff I' is a stable model of tv2sm(Π) in the sense of [Bartholomew and Lee, 2012].*

Example 6 continued For the program Π in Example 6, *tv2sm*(Π) is the following multi-valued propositional formula:

$$\begin{array}{l}\big(\neg\neg(a{=}\text{TRUE}) \rightarrow a{=}\text{TRUE}\big) \wedge \big(\neg\neg(a{=}\text{FALSE}) \rightarrow a{=}\text{FALSE}\big) \\ \;\wedge\; \big(a{=}\text{TRUE} \rightarrow b{=}\text{TRUE}\big).\end{array}$$

According to [Bartholomew and Lee, 2012], this too has only one stable model in which a and b are both mapped to TRUE, corresponding the unique stable model of Π according to Lifschitz.

Consider now that (17) contains variables. It is not difficult to see that *tv2sm*(Π) can be straightforwardly extended to non-ground program. This accounts for providing the semantics of the first-order extension of two-valued logic programs.

6 Conclusion

The interchange between strong negation and choice rules described here is limited to the case when we are interested in representing complete information in the asymmetric view of predicates. For instance, it does not apply when we omit the last rule from (6) and view that program as an incomplete description, in which case the absence of *Professor*$(john)$ in the answer set is understood as being unknown instead of being false. On the other hand, there are many cases we use answer set programming to describe complete information, such as transition systems where a state is a complete characterization of fluents. Our study contributes to understanding of strong negation in this context.

References

[Bartholomew and Lee, 2012] Michael Bartholomew and Joohyung Lee. Stable models of formulas with intensional functions. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2012. To appear.

[Ferraris *et al.*, 2011] Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz. Stable models and circumscription. *Artificial Intelligence*, 175:236–263, 2011.

[Gelfond and Lifschitz, 1991] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[Gelfond *et al.*, 1991] Michael Gelfond, Vladimir Lifschitz, and Arkady Rabinov. What are the limitations of the situation calculus? In Robert Boyer, editor, *Automated Reasoning: Essays in Honor of Woody Bledsoe*, pages 167–179. Kluwer, 1991.

[Lifschitz, 2002] Vladimir Lifschitz. Answer set programming and plan generation. *Artificial Intelligence*, 138:39–54, 2002.

[Lifschitz, 2012] Vladimir Lifschitz. Two-valued logic programs. Unpublished Draft, 2012.

A Proofs

A.1 Proof of Theorem 1

Theorem 1 *A complete set of literals is an answer set of Π in the sense of [Gelfond* et al.*, 1991] iff it is a stable model of Π^{mv} in the sense of [Bartholomew and Lee, 2012].*

Proof. Let I be the interpretation formed from including all of the literals from X and all the assignments from the multi-valued view of X. Let us denote the set of all predicate symbols from X as $\mathbf{p}$ and their negative counterparts as $\sim\mathbf{p}$ and all of the function symbols from the multi-valued view of X as $\mathbf{b}$. Clearly I satisfies

$$\forall\mathbf{x}((p(\mathbf{x}) \leftrightarrow b(\mathbf{x})=\text{TRUE}) \wedge (\sim p(\mathbf{x}) \leftrightarrow b(\mathbf{x})=\text{FALSE})),$$

for each $p \in \mathbf{p}$ and the corresponding $b \in \mathbf{b}$. From this and since X is complete, it follows that $I \models FC_b$ for each $b \in \mathbf{b}$. Thus, we can apply Theorem 3 (multiple times) to conclude that SM$[\Pi^{FOL};\ \mathbf{p} \sim\mathbf{p}] \leftrightarrow$ SM$[(\Pi^{mv})^{FOL};\ \mathbf{b}]$.

A.2 Proof of Theorem 2

Proposition 1 *If F is negative on* $\mathbf{c}$ *then*

$$(\widehat{\mathbf{c}} \leq \mathbf{c}) \rightarrow (F^*(\widehat{\mathbf{c}}) \leftrightarrow F)$$

is logically valid.

Proof. By induction. ∎

Theorem 2 *Let F be a sentence, let* $\mathbf{c}$ *be a list of predicate and function constants, and let I be a (coherent) interpretation. Let F' be a formula obtained from F by replacing a subformula $\neg H$ with $\neg H'$ such that $I \models \widetilde{\forall}(H \leftrightarrow H')$. Then*

$$I \models \text{SM}[F;\mathbf{c}] \text{ iff } I \models \text{SM}[F';\mathbf{c}]\ .$$

Proof. By Proposition 1, the following formulas are logically valid:

$$\begin{array}{l}(\mathbf{d} \leq \mathbf{c}) \rightarrow ((\neg H)^*(\mathbf{d}) \leftrightarrow (\neg H)) \\ (\mathbf{d} \leq \mathbf{c}) \rightarrow ((\neg H')^*(\mathbf{d}) \leftrightarrow (\neg H')).\end{array}$$

where $\mathbf{d}$ is a list of new constants corresponding to $\mathbf{c}$. Since $I \models \widetilde{\forall}(H \leftrightarrow H')$, we conclude that

$$I \models (\mathbf{d} \leq \mathbf{c}) \rightarrow ((\neg H)^*(\mathbf{d}) \leftrightarrow (\neg H')^*(\mathbf{d}))$$

and consequently

$$I \models (\mathbf{d} \leq \mathbf{c}) \rightarrow ((F)^*(\mathbf{d}) \leftrightarrow (F')^*(\mathbf{d}))$$

and thus

$$I \models \text{SM}[F;\mathbf{c}] \text{ iff } I \models \text{SM}[F';\mathbf{c}]\ .$$

∎

A.3 Proof of Theorem 3 and Corollary 1

Theorem 3 *For any set* $\mathbf{c}$ *of predicate and function constants,*

$$\forall \mathbf{x}((p(\mathbf{x}) \leftrightarrow b(\mathbf{x}) = \text{TRUE}) \wedge ({\sim}p(\mathbf{x}) \leftrightarrow b(\mathbf{x}) = \text{FALSE})),$$

and FC_b *entail*

$$\text{SM}[F;\ p \sim p\ \mathbf{c}] \leftrightarrow \text{SM}[F_b^{p\sim p};\ b\ \mathbf{c}]\ .$$

Proof. For any interpretation $\mathcal{I} = \langle I, X\rangle$ of signature $\sigma \supseteq \{b, p, \mathbf{c}\}$ satisfying (12), it is clear that $\mathcal{I} \models F$ iff $\mathcal{I} \models F_b^{p\sim p}$ since $F_b^{p\sim p}$ is simply the result of replacing all $p(\mathbf{t})$ with $b(\mathbf{t}) = \text{TRUE}$ and all ${\sim}p(\mathbf{t})$ with $b(\mathbf{t}) = \text{FALSE}$. Thus it only remains to be shown that $\mathcal{I} \models \neg\exists\widehat{b}\widehat{\mathbf{c}}((\widehat{b}\widehat{\mathbf{c}} < b\mathbf{c}) \wedge (F_b^{p\sim p})^*(\widehat{b}\widehat{\mathbf{c}}))$ iff $\mathcal{I} \models \neg\exists\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}}((\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}} <\sim pp\mathbf{c}) \wedge F^*(\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}}))$ or equivalently, $\mathcal{I} \models \exists\widehat{b}\widehat{\mathbf{c}}((\widehat{b}\widehat{\mathbf{c}} < b\mathbf{c}) \wedge (F_b^{p\sim p})^*(\widehat{f}\widehat{\mathbf{c}}))$ iff $\mathcal{I} \models \exists\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}}((\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}} <\sim pp\mathbf{c}) \wedge F^*(\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}}))$.

($\Rightarrow$) Assume $\mathcal{I} \models \exists\widehat{b}\widehat{\mathbf{c}}((\widehat{b}\widehat{\mathbf{c}} < b\mathbf{c}) \wedge (F_b^{p\sim p})^*(\widehat{b}, \widehat{\mathbf{c}}))$. We wish to show that $\mathcal{I} \models \exists\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}}((\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}} <\sim pp\mathbf{c}) \wedge F^*(\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}}))$

That is, take any function a of the same arity as b and any list of predicates and functions $\mathbf{d}$ of the same length $\mathbf{c}$. Now let $\mathcal{I}' = \langle I \cup J_{a\mathbf{d}}^{b\mathbf{c}}, X \cup Y_{\mathbf{d}}^{\mathbf{c}}\rangle$ be from an extended signature $\sigma' = \sigma \cup \{a, q, \mathbf{d}\}$ where J is an interpretation of functions from the signature σ and I and J agree on all symbols not occurring in $\{b, \mathbf{c}\}$. $J_{a\mathbf{d}}^{b\mathbf{c}}$ denotes the interpretation from $\sigma_{a\mathbf{d}}^{b\mathbf{c}}$ (the signature obtained from σ by replacing b with a and all elements of $\mathbf{c}$ with all elements of $\mathbf{d}$) obtained from the interpretation J by replacing b with a and the functions in $\mathbf{c}$ with the corresponding functions in $\mathbf{d}$. Similarly, $Y_{\mathbf{d}}^{\mathbf{c}}$ is the interpretation from σ' obtained from the interpretation Y by replacing predicates from $\mathbf{c}$ by the corresponding predicates from $\mathbf{d}$. We assume

$$\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c} \wedge (F_b^{p\sim p})^*(a\mathbf{d}))$$

and wish to show that there are predicates ${\sim}q$, q of the same arity as ${\sim}p$, p such that

$$\mathcal{I}' \models ({\sim}qq\mathbf{d} <\sim pp\mathbf{c} \wedge F^*({\sim}qq\mathbf{d})).$$

We define the new predicates ${\sim}q$, q in terms of b and a as follows:

$$\begin{aligned} {\sim}q(\mathbf{x}) &\leftrightarrow a(\mathbf{x}) = \text{FALSE} \wedge b(\mathbf{x}) = \text{FALSE} \\ q(\mathbf{x}) &\leftrightarrow a(\mathbf{x}) = \text{TRUE} \wedge b(\mathbf{x}) = \text{TRUE} \end{aligned}$$

We first show if $\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c})$ then $\mathcal{I}' \models ({\sim}qq\mathbf{d} <\sim pp\mathbf{c})$:
Observe that from the definition of ${\sim}q$ and q, it follows that $\mathcal{I}' \models \forall\mathbf{x}({\sim}q(\mathbf{x}) \rightarrow b(\mathbf{x}) = \text{FALSE}) \wedge \forall\mathbf{x}(q(\mathbf{x}) \rightarrow b(\mathbf{x}) = \text{TRUE})$ and from (12), this is equivalent to $\mathcal{I}' \models \forall\mathbf{x}(\sim q(\mathbf{x}) \rightarrow\sim p(\mathbf{x})) \wedge \forall\mathbf{x}(q(\mathbf{x}) \rightarrow p(\mathbf{x}))$ or simply $\mathcal{I}' \models\sim qq \leq\sim pp$. Thus, since $\mathcal{I}' \models \mathbf{d}^{pred} \leq \mathbf{c}^{pred}$, $\mathcal{I}' \models\sim qq\mathbf{d}^{pred} \leq\sim pp\mathbf{c}^{pred}$.
Case 1: $\mathcal{I}' \models \forall\mathbf{x}(b(\mathbf{x}) = a(\mathbf{x}))$.
In this case it then must be that $\mathcal{I}' \models \mathbf{d} \neq \mathbf{c}$. Thus it follows that $\mathcal{I}' \models\sim qq\mathbf{d} \neq\sim pp\mathbf{c}$. Consequently we conclude that

$$\mathcal{I}' \models ({\sim}qq\mathbf{d}^{pred} \leq\sim pp\mathbf{c}^{pred}) \wedge \sim qq\mathbf{d} \neq\sim pp\mathbf{c}$$

or simply, $\mathcal{I}' \models (\sim qq\mathbf{d} <\sim pp\mathbf{c})$.

Case 2: $\mathcal{I}' \models \neg\forall\mathbf{x}y(b(\mathbf{x}) = a(\mathbf{x}))$.
That is, since $\mathcal{I}' \models FC_b$, there is some list of object names $\mathbf{t}$ such that either $\mathcal{I}' \models b(\mathbf{t}) = \textsc{false} \land a(\mathbf{t}) \neq \textsc{false}$ or $\mathcal{I}' \models b(\mathbf{t}) = \textsc{true} \land a(\mathbf{t}) \neq \textsc{true}$.
Subcase 1: $\mathcal{I}' \models b(\mathbf{t}) = \textsc{false} \land a(\mathbf{t}) \neq \textsc{false}$
By (12), $\mathcal{I}' \models\sim p(\mathbf{t})$ and by definition of $\sim q$, $\mathcal{I}' \models \neg \sim q(\mathbf{t})$ so $\mathcal{I}' \models\sim q \neq\sim p$.
Subcase 2: $\mathcal{I}' \models b(\mathbf{t}) = \textsc{true} \land a(\mathbf{t}) \neq \textsc{true}$
By (12), $\mathcal{I}' \models p(\mathbf{t})$ and by definition of q, $\mathcal{I}' \models \neg q(\mathbf{t})$ so $\mathcal{I}' \models q \neq p$.
Therefore, no matter which subcase holds, we have $\sim qq \neq\sim pp$ and thus $\sim qq\mathbf{d} \neq\sim pp\mathbf{c}$.
Consequently we conclude

$$\mathcal{I}' \models (\sim qq\mathbf{d}^{pred} \leq\sim pp\mathbf{c}^{pred})\land \sim qq\mathbf{d} \neq\sim pp\mathbf{c}$$

or simply, $\mathcal{I}' \models (\sim qq\mathbf{d} <\sim pp\mathbf{c})$.

We now show by induction that $\mathcal{I}' \models F^*(\sim qq\mathbf{d})$:

Case 1: F is an atomic formula not containing b.
$F_b^{p\sim p}$ is exactly F thus $(F_b^{p\sim p})^*(a\mathbf{d})$ is exactly $F^*(\sim qq\mathbf{d})$ so certainly the claim holds.

Case 2: F is $\sim p(\mathbf{t})$, where $\mathbf{t}$ contains no itensional function constants.
$F^*(\sim qq\mathbf{d})$ is $\sim q(\mathbf{t})$.
$F_b^{p\sim p}$ is $b(\mathbf{t}) = \textsc{false}$.
$(F_b^{p\sim p})^*(a\mathbf{d})$ is $b(\mathbf{t}) = \textsc{false} \land a(\mathbf{t}) = \textsc{false}$.
By the definition of $\sim q$, it is clear that $\mathcal{I}' \models F^*(\sim qq\mathbf{d})$ so certainly the claim holds.

Case 3: F is $p(\mathbf{t})$, where $\mathbf{t}$ contains no itensional function constants.
$F^*(\sim qq\mathbf{d})$ is $q(\mathbf{t})$.
$F_b^{p\sim p}$ is $b(\mathbf{t}) = \textsc{true}$.
$(F_b^{p\sim p})^*(a\mathbf{d})$ is $b(\mathbf{t}) = \textsc{true} \land a(\mathbf{t}) = \textsc{true}$.
By the definition of q, it is clear that $\mathcal{I}' \models F^*(\sim qq\mathbf{d})$ so certainly the claim holds.

Case 4: F is $G \odot H$ where $\odot \in \{\land, \lor\}$.
By I.H. on G and H.

Case 5: F is $G \rightarrow H$.
By I.H. on G and H.

Case 6: F is $Q\mathbf{x}G(\mathbf{x})$ where $Q \in \{\forall, \exists\}$.
By I.H. on G.

($\Leftarrow$) Assume $\mathcal{I} \models \exists\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}}((\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}} <\sim pp\mathbf{c}) \land F^*(\widehat{\sim p}\widehat{p}\widehat{\mathbf{c}}))$. We wish to show that $\mathcal{I} \models \exists\widehat{b}\widehat{\mathbf{c}}((\widehat{b}\widehat{\mathbf{c}} < b\mathbf{c}) \land (F_b^{p\sim p})^*(\widehat{b}\widehat{\mathbf{c}}))$

That is, take any predicates $\sim q$, q of the same arity as $\sim p$, p and any list of predicates and functions $\mathbf{d}$ of the same length as $\mathbf{c}$ and let $\mathcal{I}' = \langle I \cup J^{b\mathbf{c}}_{a\mathbf{d}}, X \cup Y^{\mathbf{c}}_{\mathbf{d}}\rangle$ is defined as before. We assume

$$\mathcal{I}' \models (\sim qq\mathbf{d} <\sim pp\mathbf{c} \land F^*(\sim qq\mathbf{d}))$$

and wish to show that there is a function a of the same arity as b such that

$$\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c} \wedge (F_b^{p\sim p})^*(a\mathbf{d})).$$

We define the new function a in terms of $\sim p$, p, $\sim q$, and q as follows:

$$\mathcal{I}' \models a(\mathbf{x}) = \text{TRUE iff } \mathcal{I}' \models ((p(\mathbf{x}) \wedge q(\mathbf{x})) \vee (\sim p(\mathbf{x}) \wedge \neg \sim q(\mathbf{x})))$$
$$\mathcal{I}' \models a(\mathbf{x}) = \text{FALSE iff } \mathcal{I}' \models ((\sim p(\mathbf{x}) \wedge \sim q(\mathbf{x})) \vee (p(\mathbf{x}) \wedge \neg q(\mathbf{x})))$$

Note that since $\mathcal{I}' \models (12)$, $\mathcal{I}' \models FC_b$ and $\mathcal{I}' \models \sim qq\mathbf{d} < \sim pp\mathbf{c}$ this is a well-defined function. This is because $\mathcal{I}' \models (12)$ and $\mathcal{I}' \models FC_b$ guarantee that $\mathcal{I}'$ is complete on p. In addition to this, $\mathcal{I}' \models \sim qq\mathbf{d} < \sim pp\mathbf{c}$ guarantees that the four cases covered in this definition are the only ones possible; for any given $\mathbf{t}$ exactly one of $p(\mathbf{t})$ and $\sim p(\mathbf{t})$ is true. Wlog, assume $p(\mathbf{t})$ then $\mathcal{I}' \models \sim qq\mathbf{d} < \sim pp\mathbf{c}$ gives us that $\sim q(\mathbf{t})$ must be false and $q(\mathbf{t})$ may be true or false. The other two cases are symmetric by considering when $\sim p(\mathbf{t})$ is true.

We first show if $\mathcal{I}' \models (\sim qq\mathbf{d} < \sim pp\mathbf{c})$ then $\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c})$:
Observe that $\mathcal{I}' \models (\sim qq\mathbf{d} < \sim pp\mathbf{c})$ by definition entails $\mathcal{I}' \models (\sim qq\mathbf{d}^{pred} \leq \sim pp\mathbf{c}^{pred})$ and further by definition, $\mathcal{I}' \models (\mathbf{d}^{pred} \leq \mathbf{c}^{pred})$ and then since b and a are not predicates, $\mathcal{I}' \models ((a\mathbf{d})^{pred} \leq (b\mathbf{c})^{pred})$.

Case 1: $\mathcal{I}' \models \forall\mathbf{x}(p(\mathbf{x}) \leftrightarrow q(\mathbf{x})) \wedge \forall\mathbf{x}(\sim p(\mathbf{x}) \leftrightarrow \sim q(\mathbf{x}))$.
In this case, $\mathcal{I}' \models (\sim pp = \sim qq)$ so for it to be the case that $\mathcal{I}' \models (\sim qq\mathbf{d} < \sim pp\mathbf{c})$, it must be that $\mathcal{I}' \models \neg(\mathbf{c} = \mathbf{d})$. It then follows that $\mathcal{I}' \models \neg(b\mathbf{c} = a\mathbf{d})$. Consequently in this case, $\mathcal{I}' \models ((a\mathbf{d})^{pred} \leq (b\mathbf{c})^{pred}) \wedge \neg(b\mathbf{c} = a\mathbf{d})$ or simply $\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c})$.

Case 2: $\mathcal{I}' \models \neg(\forall\mathbf{x}y(p(\mathbf{x}) \leftrightarrow q(\mathbf{x})) \wedge \forall\mathbf{x}(\sim p(\mathbf{x}) \leftrightarrow \sim q(\mathbf{x})))$.
Since $\mathcal{I}' \models \sim qq < \sim pp$ and $\mathcal{I}' \models (12)$ and since $\mathcal{I}'$ is complete on p, there is some list of object names $\mathbf{t}$ such that either $\mathcal{I}' \models p(\mathbf{t}) \wedge \neg q(\mathbf{t})$ or $\mathcal{I}' \models \sim p(\mathbf{t}) \wedge \neg \sim q(\mathbf{t})$.
Subcase 1: $\mathcal{I}' \models p(\mathbf{t}) \wedge \neg q(\mathbf{t})$.
By (12), $\mathcal{I}' \models b(\mathbf{t}) = \text{TRUE}$ and by definition of a, $\mathcal{I}' \models a(\mathbf{t}) = \text{FALSE}$. Thus, $\mathcal{I}' \models a \neq b$. Consequently, in this case $\mathcal{I}' \models ((a\mathbf{d})^{pred} \leq (b\mathbf{c})^{pred}) \wedge \neg(b\mathbf{c} = a\mathbf{d})$ or simply $\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c})$.
Subcase 2: $\mathcal{I}' \models \sim p(\mathbf{t}) \wedge \neg \sim q(\mathbf{t})$.
By (12), $\mathcal{I}' \models b(\mathbf{t}) = \text{FALSE}$ and by definition of a, $\mathcal{I}' \models a(\mathbf{t}) = \text{TRUE}$. Thus, $\mathcal{I}' \models a \neq b$. Consequently, in this case $\mathcal{I}' \models ((a\mathbf{d})^{pred} \leq (b\mathbf{c})^{pred}) \wedge \neg(b\mathbf{c} = a\mathbf{d})$ or simply $\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c})$.

We now show by induction that $\mathcal{I}' \models (F_b^{p\sim p})^*(a\mathbf{d})$:

Case 1: F is an atomic formula not containing b.
$F_b^{p\sim p}$ is exactly F thus $(F_b^{p\sim p})^*(a\mathbf{d})$ is exactly $F^*(\sim qq\mathbf{d})$ so certainly the claim holds.

Case 2: F is $\sim p(\mathbf{t})$.
$F^*(q\mathbf{d})$ is $\sim q(\mathbf{t})$.
$F_b^{p\sim p}$ is $b(\mathbf{t}) = \text{FALSE}$.
$(F_b^{p\sim p})^*(a\mathbf{d})$ is $b(\mathbf{t}) = \text{FALSE} \wedge a(\mathbf{t}) = \text{FALSE}$.
By (12), $\mathcal{I}' \models b(\mathbf{t}) = \text{FALSE}$. By definition of a, $\mathcal{I}' \models a(\mathbf{t}) = \text{FALSE}$.

Case 3: F is $p(\mathbf{t})$.
$F^*(q\mathbf{d})$ is $q(\mathbf{t})$.
$F^{p\sim p}_b$ is $b(\mathbf{t}) = \textsc{true}$.
$(F^{p\sim p}_b)^*(a\mathbf{d})$ is $b(\mathbf{t}) = \textsc{true} \wedge a(\mathbf{t}) = \textsc{true}$.
By (12), $\mathcal{I}' \models b(\mathbf{t}) = \textsc{true}$. By definition of a, $\mathcal{I}' \models a(\mathbf{t}) = \textsc{true}$.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.
By I.H. on G and H.

Case 5: F is $G \rightarrow H$.
By I.H. on G and H.

Case 6: F is $Q\mathbf{x}G(\mathbf{x})$ where $Q \in \{\forall, \exists\}$.
By I.H. on G. ∎

Corollary 1 *For any formula F and any interpretation I of the signature of F that is complete on p, (a) I is a model of $\text{SM}[F;\ p\ \sim\! p\ \mathbf{c}]$ iff $I^{p\sim p}_{b}$ is a model of $\text{SM}[F^{p\sim p}_{b} \wedge FC_b;\ b\ \mathbf{c}]$. (b) An interpretation J of the signature of $F^{p\sim p}_{b}$ is a model of $\text{SM}[F^{p\sim p}_{b} \wedge FC_b;\ b\ \mathbf{c}]$ iff $J = I^{p\sim p}_{b}$ for some model I of $\text{SM}[F; p\ \sim\! p\ \mathbf{c}]$.*

Proof. For two interpretations I of signature σ_1 and J of signature σ_2, by $I \cup J$ we denote the interpretation of signature $\sigma_1 \cup \sigma_2$ and universe $|I| \cup |J|$ that interprets all symbols occurring only in σ_1 in the same way I does and similarly for σ_2 and J. For symbols appearing in both σ_1 and σ_2, I must interpret these the same as J does, in which case $I \cup J$ also interprets the symbol in this way.

(a⇒) Assume $I \models \textsc{true} \neq \textsc{false}$ and $I^b_p \models \text{SM}[F; p \sim\! p\mathbf{c}]$. Since $I \models \textsc{true} \neq \textsc{false}$, $I \cup I^{p\sim p}_{b} \models \textsc{true} \neq \textsc{false}$ since by definition of $I^{p\sim p}_{b}$, I and $I^{p\sim p}_{b}$ share the same universe. By definition of $I^{p\sim p}_{b}$, $I \cup I^{p\sim p}_{b} \models (12)$. Therefore, since I is complete on p and by (12), $I \cup I^{p\sim p}_{b} \models FC_b$. Thus by Theorem 3, $I \cup I^{p\sim p}_{b} \models \text{SM}[F^{p\sim p}_{b} \wedge FC_b;\ b\,\mathbf{c}] \leftrightarrow \text{SM}[F; p \sim\! p\mathbf{c}]$.

Since we assume $I \models \text{SM}[F; p \sim\! p\mathbf{c}]$, it is the case that $I \cup I^{p\sim p}_{b} \models \text{SM}[F; p \sim\! p\mathbf{c}]$ and thus it must be the case that $I \cup I^{p\sim p}_{b} \models \text{SM}[F^{p\sim p}_{b};\ b\,\mathbf{c}]$. Since $I \cup I^{p\sim p}_{b} \models FC_b$ and FC_b is a constraint, $I \cup I^{p\sim p}_{b} \models \text{SM}[F^{p\sim p}_{b} \wedge FC_b;\ b\,\mathbf{c}]$. Therefore since the signature of I does contain b, we conclude $I^{p\sim p}_{b} \models \text{SM}[F^{p\sim p}_{b} \wedge FC_b;\ b\,\mathbf{c}]$.

(a⇐) Assume $I \models \text{SM}[F^{p\sim p}_{b} \wedge FC_b; b\mathbf{c}] \wedge (\textsc{true} \neq \textsc{false})$. Since $I \models \textsc{true} \neq \textsc{false}$, $I \cup I^{p\sim p}_{b} \models \textsc{true} \neq \textsc{false}$ since by definition of $I^{p\sim p}_{b}$, I and $I^{p\sim p}_{b}$ share the same universe. By definition of $I^{p\sim p}_{b}$, $I \cup I^{p\sim p}_{b} \models (12)$. Since we assume $I \models \text{SM}[F^{p\sim p}_{b} \wedge FC_b; b\mathbf{c}]$, it follows that $I \models FC_b$. Thus by Therorem 3, $I \cup I^{p\sim p}_{b} \models \text{SM}[F^{p\sim p}_{b} \wedge FC_b; b\mathbf{c}] \leftrightarrow \text{SM}[F; p \sim\! p\mathbf{c}]$.

Since we assume $I^{p\sim p}_{b} \models \text{SM}[F^{p\sim p}_{b} \wedge FC_b; b\mathbf{c}]$, it is the case that $I \cup I^b_p \models \text{SM}[F^{p\sim p}_{b} \wedge FC_b; b\mathbf{c}]$ and thus since FC_b is a constraint, it follows that $I \cup I^b_p \models \text{SM}[F^{p\sim p}_{b}; b\mathbf{c}]$. It then follows that $I \cup I^b_p \models \text{SM}[F; p \sim\! p\mathbf{c}]$. However since the signature of $I^{p\sim p}_{b}$ does not contain p, we conclude $I \models \text{SM}[F; p \sim\! p\mathbf{c}]$.

(b⇒) Assume $J \models \textsc{true} \neq \textsc{false}$ and $J \models \text{SM}[F^{p\sim p}_{b} \wedge FC_b;\ b\,\mathbf{c}]$. Let $I = J^{\ b}_{p\sim p}$ where $J^{\ b}_{p\sim p}$ denotes the interpretation of the signature of $F^{p\sim p}_{b} \wedge FC_b$ obtained from J by replacing the boolean function b with the predicate p such that

$I \models p^I(\xi_1, \dots, \xi_k)$ for all tuples such that $b^I(\xi_1, \dots, \xi_k) = \text{TRUE}$ and,
$I \models \sim p^I(\xi_1, \dots, \xi_k)$ for all tuples such that $b^I(\xi_1, \dots, \xi_k) = \text{FALSE}$.
Since $J \models FC_b$, this is a well-defined function.
Clearly, $J = {I^{p}}_{b}^{\sim p}$ so it only remains to be shown that $I \models \text{SM}[F; p \sim p\ \mathbf{c}]$.

Since I and J have the same universe and $J \models \text{TRUE} \neq \text{FALSE}$, it follows that $I \cup J \models \text{TRUE} \neq \text{FALSE}$. Also by the definition of $J_{p \sim p}^{\ b}$ $I \cup J \models (12)$. Also, since $J \models FC_b$, it follows that $I \cup J \models FC_b$.Thus by Theorem 3, $I \cup J \models \text{SM}[{F^{p}}_{b}^{\sim p};\ b\ \mathbf{c}] \leftrightarrow \text{SM}[F; p \sim p\mathbf{c}]$.

Since we assume $J \models \text{SM}[{F^{p}}_{b}^{\sim p} \wedge FC_b;\ b\ \mathbf{c}]$, it is the case that $I \cup J \models \text{SM}[{F^{p}}_{b}^{\sim p} \wedge FC_b;\ b\ \mathbf{c}]$ and since FC_b is a constraint, $I \cup J \models \text{SM}[{F^{p}}_{b}^{\sim p};\ b\ \mathbf{c}]$. Thus it must be the case that $I \cup J \models \text{SM}[F; p \sim p\mathbf{c}]$. Now since the signature of J does not contain p, we conclude $I \models \text{SM}[F; p \sim p\mathbf{c}]$.

(b⇐)Take any I such that $J = {I^{p}}_{b}^{\sim p}$ and $I \models \text{SM}[F; p \sim p\mathbf{c}]$. Since $J \models \text{TRUE} \neq \text{FALSE}$ and I and J share the same universe, $I \cup J \models \text{TRUE} \neq \text{FALSE}$. By definition of $J = {I^{p}}_{b}^{\sim p}$, $I \cup J \models (12)$. Since I is complete on p and $I \cup J \models (12)$, it follows that $I \cup I_p^b \models FC_b$. Thus by Theorem 3, $I \cup J \models \text{SM}[{F^{p}}_{b}^{\sim p};\ b\ \mathbf{c}] \leftrightarrow \text{SM}[F; p \sim p\mathbf{c}]$

Since we assume $I \models \text{SM}[F; p \sim p\mathbf{c}]$, it is the case that $I \cup J \models \text{SM}[F; p \sim p\mathbf{c}]$ and thus it must be the case that $I \cup J \models \text{SM}[{F^{p}}_{b}^{\sim p};\ b\ \mathbf{c}]$. Since FC_b is a constraint, it then follows that $I \cup J \models \text{SM}[{F^{p}}_{b}^{\sim p} \wedge FC_b;\ b\ \mathbf{c}]$. However since the signature of I does not contain b, we conclude $J \models \text{SM}[{F^{p}}_{b}^{\sim p} \wedge FC_b;\ b\ \mathbf{c}]$. ∎

A.4 Proof of Theorem 4 and Corollary 2

Theorem 4 *For any f-plain formula F,*

$$\begin{array}{l}\forall \mathbf{x} y\big((f(\mathbf{x}) = y \leftrightarrow b(\mathbf{x}, y) = \text{TRUE}) \\ \quad \wedge (f(\mathbf{x}) \neq y \leftrightarrow b(\mathbf{x}, y) = \text{FALSE})\big)\end{array} \tag{19}$$

and $\exists xy(x \neq y)$ entail

$$\text{SM}[F; f\mathbf{c}] \ \leftrightarrow\ \text{SM}[F_b^f \wedge UE_b; b\mathbf{c}]\ .$$

Proof. For any interpretation $\mathcal{I} = \langle I, X \rangle$ of signature $\sigma \supseteq \{f, b, \mathbf{c}\}$ satisfying (19), it is clear that $\mathcal{I} \models F$ iff $\mathcal{I} \models F_b^f$ since F_b^f is simply the result of replacing all $f(\mathbf{x}) = y$ with $b(\mathbf{x}, y) = \text{TRUE}$. Thus it only remains to be shown that $\mathcal{I} \models \neg\exists \widehat{f\mathbf{c}}((\widehat{f\mathbf{c}} < f\mathbf{c}) \wedge F^*(\widehat{f\mathbf{c}}))$ iff $\mathcal{I} \models \neg\exists \widehat{b\mathbf{c}}((\widehat{b\mathbf{c}} < b\mathbf{c}) \wedge (F_b^f \wedge UE_b)^*(\widehat{b\mathbf{c}}))$ or equivalently, $\mathcal{I} \models \exists \widehat{f\mathbf{c}}((\widehat{f\mathbf{c}} < f\mathbf{c}) \wedge F^*(\widehat{f\mathbf{c}}))$ iff $\mathcal{I} \models \exists \widehat{b\mathbf{c}}((\widehat{b\mathbf{c}} < b\mathbf{c}) \wedge (F_b^f)^*(\widehat{b\mathbf{c}}) \wedge (UE_b)^*(\widehat{b\mathbf{c}}))$.

(⇒) Assume $\mathcal{I} \models \exists \widehat{f\mathbf{c}}((\widehat{f\mathbf{c}} < f\mathbf{c}) \wedge F^*(\widehat{f}, \widehat{\mathbf{c}}))$. We wish to show that $\mathcal{I} \models \exists \widehat{b\mathbf{c}}((\widehat{b\mathbf{c}} < b\mathbf{c}) \wedge (F_b^f)^*(\widehat{b\mathbf{c}}) \wedge (UE_b)^*(\widehat{b\mathbf{c}}))$

That is, take any function g of the same arity as f and any list of predicates and functions $\mathbf{d}$ of the same length $\mathbf{c}$. Now let $\mathcal{I}' = \langle I \cup J_{ag\mathbf{d}}^{bf\mathbf{c}}, X \cup Y_{\mathbf{d}}^{\mathbf{c}} \rangle$ be from an extended signature $\sigma' = \sigma \cup \{g, a, \mathbf{d}\}$ where J is an interpretation of functions from the signature σ and I and J agree on all symbols not occurring in $\{b, f, \mathbf{c}\}$. $J_{ag\mathbf{d}}^{bf\mathbf{c}}$ denotes the interpretation from $\sigma_{ag\mathbf{d}}^{bf\mathbf{c}}$ (the signature obtained from σ by replacing f with g, b with a, and all elements of $\mathbf{c}$ with all elements of $\mathbf{d}$) obtained from the interpretation J by replacing f with g, b with a, and the functions in $\mathbf{c}$ with the corresponding functions in $\mathbf{d}$. Similarly, $Y^{\mathbf{c}}_{\mathbf{d}}$ is the interpretation from σ' obtained from the interpretation Y by replacing predicates from $\mathbf{c}$ by the corresponding predicates from $\mathbf{d}$. We assume

$$\mathcal{I}' \models (g\mathbf{d} < f\mathbf{c} \wedge F^*(g\mathbf{d}))$$

and wish to show that there is a (boolean) function a of the same arity as b such that

$$\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c} \wedge (F^f_b)^*(a\mathbf{d}) \wedge (UE_b)^*(a\mathbf{d})).$$

We define the new function a in terms of g as follows:

$$a^{\mathcal{I}'}(\boldsymbol{\xi}, \xi') = \begin{cases} \text{TRUE} & \text{if } \mathcal{I}' \models g(\boldsymbol{\xi}) = \xi' \\ \text{FALSE} & \text{otherwise.} \end{cases}$$

We first show $\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c})$:
Case 1: $\mathcal{I}' \models \forall \mathbf{x} y (f(\mathbf{x}) = y \leftrightarrow g(\mathbf{x}) = y)$.
In this case it then must be the case that $\mathcal{I}' \models \mathbf{d} \neq \mathbf{c}$. Thus it follows that $\mathcal{I}' \models a\mathbf{d} \neq b\mathbf{c}$. Consequently we conclude that

$$\mathcal{I}' \models (\mathbf{d}^{pred} \leq \mathbf{c}^{pred}) \wedge a\mathbf{d} \neq b\mathbf{c}$$

or simply, $\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c})$.

Case 2: $\mathcal{I}' \models \neg\forall \mathbf{x} y (f(\mathbf{x}) = y \leftrightarrow g(\mathbf{x}) = y)$.
In this case it then must be the case that for some list of object names $\mathbf{t}$ and some c that $\mathcal{I}' \models f(\mathbf{t}) = c \wedge g(\mathbf{t}) \neq c$. By the definition of a, this means that $a(\mathbf{t}, c)^{\mathcal{I}'} = \text{FALSE}$ but by (19), $b(\mathbf{t}, c)^{\mathcal{I}'} = \text{TRUE}$. Therefore, $\mathcal{I}' \models a \neq b$ and thus $\mathcal{I}' \models a\mathbf{d} \neq b\mathbf{c}$. Consequently we conclude

$$\mathcal{I}' \models (\mathbf{d}^{pred} \leq \mathbf{c}^{pred}) \wedge a\mathbf{d} \neq b\mathbf{c}$$

or simply, $\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c})$.

We now show by induction that $\mathcal{I}' \models (F^f_b)^*(a\mathbf{d})$:

Case 1: F is an atomic formula not containing f.
F^f_b is exactly F thus $F^*(g\mathbf{d})$ is exactly $(F^f_b)^*(a\mathbf{d})$ so certainly the claim holds.

Case 2: F is $f(\mathbf{t}) = c$, where $\mathbf{t}$ contains no intensional functions.
$F^*(g\mathbf{d})$ is $f(\mathbf{t}) = c \wedge g(\mathbf{t}) = c$.
F^f_b is $b(\mathbf{t}, c) = \text{TRUE}$.
$(F^f_b)^*(a\mathbf{d})$ is $b(\mathbf{t}, c) = \text{TRUE} \wedge a(\mathbf{t}, c) = \text{TRUE}$.
Since $\mathcal{I}' \models F^*(g\mathbf{d})$, by the definition of a, $\mathcal{I}' \models a(\mathbf{t}, c) = \text{TRUE}$ and from (19) it follows that $\mathcal{I}' \models b(\mathbf{t}, c) = true$.

Case 3: F is $f(\mathbf{t}) = c$, where $\mathbf{t}$ contains at least one intensional function.
$F^*(g\mathbf{d})$ is $f(\mathbf{t}) = c \wedge g(\mathbf{t}^{f\mathbf{c}}_{g\mathbf{d}}) = c$.

F_b^f is $b(\mathbf{t}, c) = \text{TRUE}$.
$(F_b^f)^*(a\mathbf{d})$ is $b(\mathbf{t}, c) = \text{TRUE} \wedge a(\mathbf{t}_{g\mathbf{d}}^{f\mathbf{c}}, c) = \text{TRUE}$.
Since $\mathcal{I}' \models F^*(g\mathbf{d})$, by the definition of a, $\mathcal{I}' \models a(\mathbf{t}_{g\mathbf{d}}^{f\mathbf{c}}, c) = \text{TRUE}$ and from (19) it follows that $\mathcal{I}' \models b(\mathbf{t}, c) = true$.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.
By I.H. on G and H.

Case 5: F is $G \rightarrow H$.
By I.H. on G and H.

Case 6: F is $Q\mathbf{x}G(\mathbf{x})$ where $Q \in \{\forall, \exists\}$.
By I.H. on G.

We now show that $\mathcal{I}' \models (UE_b)^*(a\mathbf{d})$:
$(UE_b)^*(a\mathbf{d})$ is equivalent to the following 2 formulas:

$$\begin{array}{l}\forall \mathbf{x}yz(y \neq z \wedge (b(\mathbf{x}, y) = \text{TRUE} \rightarrow b(\mathbf{x}, z) = \text{FALSE})\wedge \\ (b(\mathbf{x}, y) = \text{TRUE} \wedge a(\mathbf{x}, y) = \text{TRUE} \rightarrow \\ b(\mathbf{x}, z) = \text{FALSE} \wedge a(\mathbf{x}, z) = \text{FALSE}))\end{array} \qquad (13^*)$$

$$\forall \mathbf{x} \exists y (b(\mathbf{x}, y) = \text{TRUE})(14^*)$$

(14^*) follows from (19). The first implication of (13^*) follows from (19) . All that remains to show is that the second implication of (13^*) is satisfied by $\mathcal{I}'$.

Now, take any vector of terms $\mathbf{t_x}$ the same length as $\mathbf{x}$ and any terms t_y and t_z. If $\mathcal{I}' \not\models b(\mathbf{t_x}, t_y) = \text{TRUE} \wedge a(\mathbf{t_x}, t_y) = \text{TRUE}$, then the second implication of (13^*) is satisfied by $\mathcal{I}'$. If instead $\mathcal{I}' \models b(\mathbf{t_x}, t_y) = \text{TRUE} \wedge a(\mathbf{t_x}, t_y) = \text{TRUE}$ then by (19) , $\mathcal{I}' \models b(\mathbf{t_x}, t_z) = \text{FALSE}$ for $t_y \neq t_z$ and by definition of a, $\mathcal{I}' \models a(\mathbf{t_x}, t_z) = \text{FALSE}$ for $t_y \neq t_z$ so in this case too, the second implication of (13^*) is satisfied by $\mathcal{I}'$.

($\Leftarrow$) Assume $\mathcal{I} \models \exists \widehat{b}\widehat{\mathbf{c}}((\widehat{b}\widehat{\mathbf{c}} < b\mathbf{c}) \wedge (F_b^f)^*(\widehat{b}\widehat{\mathbf{c}}) \wedge (UE_b)^*(\widehat{b}\widehat{\mathbf{c}}))$. We wish to show that $\mathcal{I} \models \exists \widehat{f}\widehat{\mathbf{c}}((\widehat{f}\widehat{\mathbf{c}} < f\mathbf{c}) \wedge F^*(\widehat{f}\widehat{\mathbf{c}}))$

That is, take any (boolean) function a of the same arity as b and any list of predicates and functions $\mathbf{d}$ the same length as $\mathbf{c}$ and let $\mathcal{I}' = \langle I \cup J_{ag\mathbf{d}}^{bf\mathbf{c}}, X \cup Y_{\mathbf{d}}^{\mathbf{c}} \rangle$ be defined as before. We assume

$$\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c} \wedge (F_b^f)^*(a\mathbf{d}) \wedge (UE_b)^*(a\mathbf{d}))$$

and wish to show that there is a function g of the same arity as f such that

$$\mathcal{I}' \models (g\mathbf{d} < f\mathbf{c} \wedge F^*(g\mathbf{d})).$$

We define the new function g in terms of a as follows:

$$g^{\mathcal{I}'}(\boldsymbol{\xi}) = \begin{cases} f(\boldsymbol{\xi}) & \text{if } \mathcal{I}' \models a(\boldsymbol{\xi}, f(\boldsymbol{\xi})) = \text{TRUE} \\ m(f(\boldsymbol{\xi})) & \text{otherwise} \end{cases}$$

where m is a mapping from the universe to itself such that $\forall x(m(x) \neq x)$. Note that the assumption that there are at least two elements in the universe is essential to this definition.

We first show $\mathcal{I}' \models (g\mathbf{d} < f\mathbf{c})$:

Case 1: $\mathcal{I}' \models \forall \mathbf{x} y(b(\mathbf{x}, y) = a(\mathbf{x}, y))$.
In this case, $\mathcal{I}' \models (a = b)$ so for it to be the case that $\mathcal{I}' \models (a\mathbf{d} < b\mathbf{c})$, it must be that $\mathcal{I}' \models \neg(\mathbf{c} = \mathbf{d})$. It then follows that $\mathcal{I}' \models \neg(f\mathbf{c} = g\mathbf{d})$. Consequently in this case, $\mathcal{I}' \models ((g\mathbf{d})^{pred} \leq (f\mathbf{c})^{pred}) \wedge \neg(f\mathbf{c} = g\mathbf{d})$ or simply $\mathcal{I}' \models (g\mathbf{d} < f\mathbf{c})$.

Case 2: $\mathcal{I}' \models \neg\forall \mathbf{x} y(b(\mathbf{x}, y) = a(\mathbf{x}, y))$.
Thus, for some $\boldsymbol{\xi}$, $\mathcal{I}' \models \neg\forall y(b(\boldsymbol{\xi}, y) = a(\boldsymbol{\xi}, y))$. Further, for some ξ', $b(\boldsymbol{\xi}, \xi') \neq a(\boldsymbol{\xi}, \xi')$. Then from $\mathcal{I}' \models UE^*_b(a, \mathbf{d})$, it follows that for some ξ', $b(\boldsymbol{\xi}, \xi') = \text{TRUE} \wedge a(\boldsymbol{\xi}, \xi') = \text{FALSE}$. This is because $\mathcal{I}' \models UE^*_b(a, \mathbf{d})$ means that there must be some ξ' for which $b(\boldsymbol{\xi}, \xi') = \text{TRUE}$ and since if $b(\boldsymbol{\xi}, \xi') = \text{TRUE} \wedge a(\boldsymbol{\xi}, \xi') = \text{TRUE}$, then, $a = b$, which we assume not to be the case. Thus by definition of g, $\mathcal{I}' \models g(\boldsymbol{\xi}) = m(f(\boldsymbol{\xi}))$. And since $m(f(\boldsymbol{\xi}) \neq f(\boldsymbol{\xi}$, we conclude $\mathcal{I}' \models \neg(g = f)$. It then follows that $\mathcal{I}' \models \neg(f\mathbf{c} = g\mathbf{d})$. Consequently in this case, $\mathcal{I}' \models ((g\mathbf{d})^{pred} \leq (f\mathbf{c})^{pred}) \wedge \neg(f\mathbf{c} = g\mathbf{d})$ or simply $\mathcal{I}' \models (g\mathbf{d} < f\mathbf{c})$.

We now show by induction that $\mathcal{I}' \models F^*(g\mathbf{d})$:

Case 1: F is an atomic formula not containing f.
F^f_b is exactly F thus $F^*(g\mathbf{d})$ is exactly $(F^f_b)^*(a\mathbf{d})$ so certainly the claim holds.

Case 2: F is $f(\mathbf{t}) = c$, where $\mathbf{t}$ contains no intensional functions.
$F^*(g\mathbf{d})$ is $f(\mathbf{t}) = c \wedge g(\mathbf{t}) = c$.
F^f_b is $b(\mathbf{t}, c) = \text{TRUE}$.
$(F^f_b)^*(a\mathbf{d})$ is $a(\mathbf{t}, c) = \text{TRUE} \wedge b(\mathbf{t}, c) = \text{TRUE}$.
Since $\mathcal{I}' \models (F^f_b)^*(a\mathbf{d})$, it follows from the definition of g that $g(\mathbf{t}) = f(\mathbf{t})$. From (19), it follows that $\mathcal{I}' \models f(\mathbf{t}) = c$ and so it must be that $\mathcal{I}' \models g(\mathbf{t}) = c$, from which we conclude $\mathcal{I}' \models F^*(g\mathbf{d})$.

Case 3: F is $f(\mathbf{t}) = c$, where $\mathbf{t}$ contains at least one intensional function.
$F^*(g\mathbf{d})$ is $f(\mathbf{t}) = c \wedge g(\mathbf{t}^{f\mathbf{c}}_{g\mathbf{d}}) = c$.
F^f_b is $b(\mathbf{t}, c) = \text{TRUE}$.
$(F^f_b)^*(a\mathbf{d})$ is $a(\mathbf{t}^{f\mathbf{c}}_{g\mathbf{d}}, c) = \text{TRUE} \wedge b(\mathbf{t}, c) = \text{TRUE}$.
Since $\mathcal{I}' \models (F^f_b)^*(a\mathbf{d})$, it follows from the definition of g that $g(\mathbf{t}^{f\mathbf{c}}_{g\mathbf{d}}) = f(\mathbf{t})$. From (19), it follows that $\mathcal{I}' \models f(\mathbf{t}) = c$ and so it must be that $\mathcal{I}' \models g(\mathbf{t}^{f\mathbf{c}}_{g\mathbf{d}}) = c$, from which we conclude $\mathcal{I}' \models F^*(g\mathbf{d})$.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.
By I.H. on G and H.

Case 5: F is $G \rightarrow H$.
By I.H. on G and H.

Case 6: F is $Q\mathbf{x}G(\mathbf{x})$ where $Q \in \{\forall, \exists\}$.
By I.H. on G. ∎

Corollary 2 *Let F be an f-plain sentence. (a) An interpretation I of the signature of F that satisfies $\exists xy(x \neq y)$ is a model of* $\mathrm{SM}[F; f\mathbf{c}]$ *iff* I^f_b *is a model of* $\mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$. *(b) An interpretation J of the signature of F^f_b that satisfies $\exists xy(x \neq y)$ is a model of* $\mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$ *iff $J = I^f_b$ for some model I of* $\mathrm{SM}[F; f\mathbf{c}]$.

Proof. For two interpretations I of signature σ_1 and J of signature σ_2, by $I \cup J$ we denote the interpretation of signature $\sigma_1 \cup \sigma_2$ and universe $|I| \cup |J|$ that interprets all symbols occurring only in σ_1 in the same way I does and similarly for σ_2 and J. For symbols appearing in both σ_1 and σ_2, I must interpret these the same as J does, in which case $I \cup J$ also interprets the symbol in this way.

(a$\Rightarrow$) Assume $I \models \mathrm{SM}[F; f\mathbf{c}] \wedge \exists xy(x \neq y)$. Since $I \models \exists xy(x \neq y)$, $I \cup I^f_b \models \exists xy(x \neq y)$ since by definition of I^f_b, I and I^f_b share the same universe. By definition of I^f_b, $I \cup I^f_b \models$ (19). Thus by Theorem 4, $I \cup I^f_b \models \mathrm{SM}[F; f\mathbf{c}] \leftrightarrow \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$.

Since we assume $I \models \mathrm{SM}[F; f\mathbf{c}]$, it is the case that $I \cup I^f_b \models \mathrm{SM}[F; f\mathbf{c}]$ and thus it must be the case that $I \cup I^f_b \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$. However since the signature of I does not contain b, we conclude $I^f_b \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$.

(a$\Leftarrow$) Assume $I \models \exists xy(x \neq y)$ and $I^f_b \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$. Since $I \models \exists xy(x \neq y)$, $I \cup I^f_b \models \exists xy(x \neq y)$ since by definition of I^f_b, I and I^f_b share the same universe. By definition of I^f_b, $I \cup I^f_b \models$ (19). Thus by Theorem 4, $I \cup I^f_b \models \mathrm{SM}[F; f\mathbf{c}] \leftrightarrow \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$.

Since we assume $I^f_b \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$, it is the case that $I \cup I^f_b \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$ and thus it must be the case that $I \cup I^f_b \models \mathrm{SM}[F; f\mathbf{c}]$. Therefore since the signature of I^f_b does contain f, we conclude $I \models \mathrm{SM}[F; f\mathbf{c}]$.

(b$\Rightarrow$) Assume $J \models \exists xy(x \neq y)$ and $J \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$. Let $I = J^b_f$ where J^b_f denotes the interpretation of the signature of F obtained from J by replacing the boolean function b with the function f such that $f^I(\xi_1, \dots, \xi_k) = \xi_{k+1}$ for all tuples such that $b^I(\xi_1, \dots, \xi_k, \xi_{k+1}) = \textsc{true}$. This is a valid definition of a function since we assume $J \models \mathrm{SM}[F^f_p \wedge UE_b; b\mathbf{c}]$, from which we obtain $J \models UE_b$. Clearly, $J = I^f_b$ so it only remains to be shown that $I \models \mathrm{SM}[F; f\mathbf{c}]$.

Since I and J have the same universe and $J \models \exists xy(x \neq y)$, it follows that $I \cup J \models \exists xy(x \neq y)$. Also by the definition of J^b_f $I \cup J \models$ (19). Thus by Theorem 4, $I \cup J \models \mathrm{SM}[F; f\mathbf{c}] \leftrightarrow \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$

Since we assume $J \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$, it is the case that $I \cup J \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$ and thus it must be the case that $I \cup J \models \mathrm{SM}[F; f\mathbf{c}]$. Now since the signature of J does not contain f, we conclude $I \models \mathrm{SM}[F; f\mathbf{c}]$.

(b$\Leftarrow$)Take any I such that $J = I^f_b$ and $I \models \mathrm{SM}[F; f\mathbf{c}]$. Since $J \models \exists xy(x \neq y)$ and I and J share the same universe, $I \cup J \models \exists xy(x \neq y)$. By definition of $J = I^f_b$, $I \cup J \models$ (19). Thus by Theorem 4, $I \cup J \models \mathrm{SM}[F; f\mathbf{c}] \leftrightarrow \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$

Since we assume $I \models \mathrm{SM}[F; f\mathbf{c}]$, it is the case that $I \cup J \models \mathrm{SM}[F; f\mathbf{c}]$ and thus it must be the case that $I \cup J \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$. However since the signature of I does not contain b, we conclude $J \models \mathrm{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$. ∎

A.5 Proof of Theorem 5 and Corollary 3

Theorem 5 *For any f-plain formula F, formulas*

$$\forall \mathbf{x}y(f(\mathbf{x}) = y \leftrightarrow p(\mathbf{x}, y)), \tag{20}$$

$$\forall \mathbf{x}y(f(\mathbf{x}) \neq y \leftrightarrow {\sim}p(\mathbf{x}, y)), \tag{21}$$

$$\exists xy(x \neq y)$$

entail

$$\text{SM}[F; f\mathbf{c}] \leftrightarrow \text{SM}[F^f_p \wedge UE_p; {\sim}pp\mathbf{c}]\ .$$

Proof. Let b be a function such that the following hold:

$$\forall \mathbf{x}y(f(\mathbf{x}) = y \leftrightarrow b(\mathbf{x}, y) = \text{TRUE})$$

$$\forall \mathbf{x}y(f(\mathbf{x}) \neq y \leftrightarrow b(\mathbf{x}, y) = \text{FALSE})$$

$$\forall \mathbf{x}y(b(\mathbf{x}, y) = \text{TRUE} \leftrightarrow p(\mathbf{x}, y))$$

$$\forall \mathbf{x}y(b(\mathbf{x}, y) = \text{FALSE} \leftrightarrow {\sim}p(\mathbf{x}, y))$$

Note that such a definition is only valid for interpretations satisfying (20) and (20), which we assume to be the case. Now, by Theorem 4, $\text{SM}[F; f\mathbf{c}] \leftrightarrow \text{SM}[F^f_b \wedge UE_b; b\mathbf{c}]$. By Theorem 4, $\text{SM}[F^f_b \wedge UE_b; b\mathbf{c}] \leftrightarrow \text{SM}[(F^f_b \wedge UE_b)^b_p; {\sim}pp\mathbf{c}]$. Thus, it follows that $\text{SM}[F; f\mathbf{c}] \leftrightarrow \text{SM}[(F^f_b \wedge UE_b)^b_p; {\sim}pp\mathbf{c}]$.

It only remains to show that $\text{SM}[(F^f_b \wedge UE_b)^b_p; {\sim}pp\mathbf{c}]$ is precisely $\text{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$. $(F^f_b)^b_p$ first replaces f in the signature with b, replaces all $f(\mathbf{t}) = c$ with $b(\mathbf{t}, c) = \text{TRUE}$, The composed translation then replaces b with p and ${\sim}p$ in the signature, replaces all $b(\mathbf{t}, c) = \text{TRUE}$ with $p(\mathbf{t}, c)$, and replaces all $b(\mathbf{t}, c) = \text{FALSE}$ with ${\sim}p(\mathbf{t}, c)$ (however F^f_b does not contain $b(\mathbf{t}, c) = \text{FALSE}$; only UE_b contains $b(\mathbf{t}, c) = \text{FALSE}$). This part is equivalent to F^f_p. It is easy to see that $(UE_b)^b_p$ is $UE_{\sim p}$. ∎

Corollary 3 *Let F be an f-plain sentence. (a) An interpretation I of the signature of F that satisfies $\exists xy(x \neq y)$ is a model of $\text{SM}[F; f\mathbf{c}]$ iff $I^f_{\sim pp}$ is a model of $\text{SM}[F^f_p \wedge UE_p; {\sim}pp\mathbf{c}]$. (b) An interpretation J of the signature of F^f_p that satisfies $\exists xy(x \neq y)$ is a model of $\text{SM}[F^f_p \wedge UE_p; {\sim}pp\mathbf{c}]$ iff $J = I^f_{\sim pp}$ for some model I of $\text{SM}[F; f\mathbf{c}]$.*

Proof. For two interpretations I of signature σ_1 and J of signature σ_2, by $I \cup J$ we denote the interpretation of signature $\sigma_1 \cup \sigma_2$ and universe $|I| \cup |J|$ that interprets all symbols occurring only in σ_1 in the same way I does and similarly for σ_2 and J. For symbols appearing in both σ_1 and σ_2, I must interpret these the same as J does, in which case $I \cup J$ also interprets the symbol in this way.

(a$\Rightarrow$) Assume $I \models \text{SM}[F; b\mathbf{c}] \wedge \exists xy(x \neq y)$. Since $I \models \exists xy(x \neq y)$, $I \cup I^f_{\sim pp} \models \exists xy(x \neq y)$ since by definition of $I^f_{\sim pp}$, I and $I^f_{\sim pp}$ share the same universe. By definition of $I^f_{\sim pp}$, $I \cup I^f_{\sim pp} \models (20) \wedge (21)$. Thus by Theorem 5, $I \cup I^f_{\sim pp} \models \text{SM}[F; f\mathbf{c}] \leftrightarrow \text{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$.

Since we assume $I \models \mathrm{SM}[F; f\mathbf{c}]$, it is the case that $I \cup I^f_{\sim pp} \models \mathrm{SM}[F; f\mathbf{c}]$ and thus it must be the case that $I \cup I^f_{\sim pp} \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$. However since the signature of I does not contain p or ${\sim}p$, we conclude $I^f_{\sim pp} \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$.

(a⇐) Assume $I \models \exists xy(x \neq y)$ and $I^f_{\sim pp} \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$. Since $I \models \exists xy(x \neq y)$, $I \cup I^f_{\sim pp} \models \exists xy(x \neq y)$ since by definition of $I^f_{\sim pp}$, I and $I^f_{\sim pp}$ share the same universe. By definition of $I^f_{\sim pp}$, $I \cup I^f_{\sim pp} \models (20) \wedge (21)$. Thus by Theorem 5, $I \cup I^f_{\sim pp} \models \mathrm{SM}[F; f\mathbf{c}] \leftrightarrow \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$.

Since we assume $I^f_{\sim pp} \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$, it is the case that $I \cup I^f_{\sim pp} \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$ and thus it must be the case that $I \cup I^f_{\sim pp} \models \mathrm{SM}[F; f\mathbf{c}]$. Therefore since the signature of $I^f_{\sim pp}$ does contain f, we conclude $I \models \mathrm{SM}[F; f\mathbf{c}]$.

(b⇒) Assume $J \models \exists xy(x \neq y)$ and $J \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$. Let $I = J^{\sim pp}_f$ where $J^{\sim pp}_f$ denotes the interpretation of the signature of F obtained from J by replacing the predicates p and ${\sim}p$ with the function f such that
$f^I(\xi_1, \dots, \xi_k) = \xi_{k+1}$ for all tuples such that $J \models p^J(\xi_1, \dots, \xi_k, \xi_{k+1})$. Note that this definition of f is well-defined due to the fact that $J \models UE_{\sim pp}$.
Clearly, $J = I^f_{\sim pp}$ so it only remains to be shown that $I \models \mathrm{SM}[F; f\mathbf{c}]$.

Since I and J have the same universe and $J \models \exists xy(x \neq y)$, it follows that $I \cup J \models \exists xy(x \neq y)$. Also by the definition of $J^{\sim pp}_f$, $I \cup J \models (20) \wedge (21)$. Thus by Theorem 5, $I \cup J \models \mathrm{SM}[F; f\mathbf{c}] \leftrightarrow \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$.

Since we assume $J \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$, it is the case that $I \cup J \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$ and thus it must be the case that $I \cup J \models \mathrm{SM}[F; f\mathbf{c}]$. Now since the signature of J does not contain f, we conclude $I \models \mathrm{SM}[F; f\mathbf{c}]$.

(b⇐)Take any I such that $J = I^f_{\sim pp}$ and $I \models \mathrm{SM}[F; f\mathbf{c}]$. Since $J \models \exists xy(x \neq y)$ and I and J share the same universe, $I \cup J \models \exists xy(x \neq y)$. By definition of $J = I^f_{\sim pp}$, $I \cup J \models (20) \wedge (21)$. Thus by Theorem 5, $I \cup J \models \mathrm{SM}[F; f\mathbf{c}] \leftrightarrow \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$.

Since we assume $I \models \mathrm{SM}[F; f\mathbf{c}]$, it is the case that $I \cup J \models \mathrm{SM}[F; f\mathbf{c}]$ and thus it must be the case that $I \cup J \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$. However since the signature of I does not contain p or ${\sim}p$, we conclude $J \models \mathrm{SM}[F^f_p \wedge UE_{\sim p}; {\sim}pp\mathbf{c}]$. ∎

A.6 Proof of Theorem 6

Theorem 6 *For any two-valued program Π of signature σ, an interpretation I is a stable model of Π in the sense of Lifchiz iff I' is a stable model of tvlp2sm(Π) in the sense of [Bartholomew and Lee, 2012].*

Proof. Given a two-valued program Π and a multi-valued interpretation I, we begin by showing that $I \models \Pi^I$ iff $I' \models T(\Pi)^{I'}$. Consider a rule R of the form (17). There are three cases–either $I \models F$ and $I \models R^I$, $I \models F$ and $I \not\models R^I$ or $I \not\models F$:

- $I \models F$ and $I \models R^I$.
  In this case, the reduct R^I is $L_0 \leftarrow L_1, \dots, L_n$ and since $I \models R^I$, then either $I \models L_0$ or there is some $L_i \in \{L_1, \dots, L_n\}$ such that $I \not\models L_i$. If $I \models L_0$, then by definition of I', $I' \models T(L_0)$ and thus the reduct $T(R)^{I'}$ is *body* $\rightarrow T(L_0)$ which is satisfied by I' no matter what *body* happens to be. If on the other than $I \not\models L_i$ for some

$L_i \in \{L_1, \ldots, L_n\}$, then the reduct $T(R)^{I'}$ is $\bot \rightarrow head$ since by definition of I', $I' \not\models T(L_i)$ and thus the subformula $T(L_1) \wedge \ldots T(L_i) \wedge \ldots, T(L_n)$ is replaced by $\bot$ and then in this case too, the reduct $T(R)^{I'}$ is satisfied by I'.

- $I \models F$ and $I \not\models R^I$
  In this case, the reduct R^I is $L_0 \leftarrow L_1, \ldots, L_n$ and since $I \not\models R^I$, it must be the case that $I \models L_1, \ldots, L_n$ and $I \not\models L_0$. On the other hand, by definition of I', this means that $I' \models T(L_1) \wedge \cdots \wedge T(L_n)$. Since $I \models F$, $I' \models F$ and thus $I' \models \neg\neg F$. However $I' \not\models T(L_0)$ so the entire rule is a maximal subformula not satisfied by I' so the entire rule becomes $\bot$ in the reduct and thus $I' \not\models T(R)^{I'}$.
- $I \not\models F$
  In this case, the reduct R^I omits this rule entirely so certainly $I \models R^I$. On the other hand, by definition of I', $I' \not\models T(F)$ and thus $I' \not\models \neg\neg T(F)$ and further $I' \not\models \neg\neg T(F) \wedge T(L_1) \wedge \cdots \wedge T(L_n)$. Thus, the reduct $T(R)^{I'}$ is $\bot \rightarrow head$ since the entire body is not satisfied by I' and so $I' \models T(R)^{I'}$ no matter what $head$ happens to be.

■