# *On the Stable Model Semantics for Intensional Functions*

Michael Bartholomew and Joohyung Lee
*School of Computing, Informatics, and Decision Systems Engineering*
*Arizona State University, Tempe, USA*
(*e-mail:* {mjbartho,joolee}@asu.edu)

*submitted 10 April 2013; revised 23 May 2013; accepted 23 June 2013*

**Abstract**

Several extensions of the stable model semantics are available to describe "intensional" functions—functions that can be described in terms of other functions and predicates by logic programs. Such functions are useful for expressing inertia and default behaviors of systems, and can be exploited for alleviating the grounding bottleneck involving functional fluents. However, the extensions were defined in different ways under different intuitions. In this paper we provide several reformulations of the extensions, and note that they are in fact closely related to each other and coincide on large syntactic classes of logic programs.

*KEYWORDS*: Answer Set Programming, Stable Models, Intensional Functions

## 1 Introduction

Several extensions of the stable model semantics were proposed to allow "intensional" functions—functions that can be described in terms of other functions and predicates by logic programs (Cabalar 2011; Lifschitz 2012; Bartholomew and Lee 2012; Balduccini 2012b). Such functions significantly enhance the modeling capability of the language of answer set programming by providing natural representations of nonBoolean fluents, such as the location of an object, or the level of a water tank. The following example demonstrates the ability to assign a default value to a function, which is useful for expressing inertia and default behaviors of systems.

*Example 1*
*The following program $F$ describes the capacity of a water tank that has a leak but that can be refilled to the maximum amount, say 10, with the action FillUp.*

$$\begin{array}{rcl} \{Amount_1 = x\} & \leftarrow & Amount_0 = x+1 \\ Amount_1 = 10 & \leftarrow & FillUp\ . \end{array} \tag{1}$$

*Here $Amount_1$ is an intensional function constant, and $x$ is a variable ranging over nonnegative integers. According to (Bartholomew and Lee 2013), the first rule is a choice rule standing for $(Amount_1 = x) \vee \neg(Amount_1 = x) \leftarrow Amount_0 = x+1$, which asserts that the amount decreases by default. However, if FillUp action is executed (e.g., if we add FillUp as a fact), this behavior is overridden, and the amount is set to the maximum value.*

Recently, (Bartholomew and Lee 2013) showed that functional stable model semantics can be used as a natural basis of combining answer set programming and satisfiability modulo theories.

For instance, the paper showed that, if a program is "tight," as is the case in the example above, it can be turned into the input language of Satisfiability Modulo Theories (SMT) solvers, thereby allowing us to apply efficient constraint solving methods in SMT to alleviate the grounding bottleneck involving functional fluents. For example, program (1) can be turned into the SMT instance

$$\big((Amount_0 = Amount_1 + 1) \lor (Amount_1 = 10 \land FillUp)\big) \\ \land\ (FillUp \rightarrow Amount_1 = 10)\ .$$

Similarly, (Balduccini 2012a) reports the computational efficiency of a system that computes the semantics of intensional functions defined in (Balduccini 2012b).

However, the existing semantics of intensional functions were defined in very different styles under different intuitions, which obscures the relationships among them. Though the relationship between the language of (Lifschitz 2012) and the language of (Bartholomew and Lee 2012) was discussed in (Bartholomew and Lee 2012), their relationships to other functional stable model semantics were left open. Roughly speaking, the languages in (Lifschitz 2012) and (Bartholomew and Lee 2012) are defined in terms of second-order formulas, which express the nonmonotonicity of the semantics by ensuring the uniqueness of function values. On the other hand, the language in (Cabalar 2011) is defined in terms of a modification to equilibrium logic by allowing functions to be partially defined, and applying a "minimality" condition on such functions. The idea behind the language in (Balduccini 2012b) appears similar to the one in (Cabalar 2011), but the definition looks very different. It is defined in terms of a modification to the notion of a reduct.

In this paper we provide several reformulations of the extensions, which reveal that these semantics are in fact closely related to each other, and coincide on large syntactic classes of logic programs. The relationships allow us to transfer some mathematical results established for one language to another language. Additionally, an implementation of one language can be viewed as an implementation of another language when restricting attention to one of these syntactic classes.

Section 2 reviews the Bartholomew-Lee semantics from (Bartholomew and Lee 2012; Bartholomew and Lee 2013) and the Cabalar semantics from (Cabalar 2011). Section 3 presents reformulations of these definitions, and based on the result, Section 4 relates the two semantics. Further, Section 5 shows how to reduce the Cabalar semantics to the first-order stable model semantics from (Ferraris et al. 2011), and Section 6 shows how the Balduccini semantics (Balduccini 2012b) can be viewed as a special case of the Cabalar semantics.

## 2 Preliminaries

### *2.1 Review: Original Definition of Bartholomew-Lee Semantics in Terms of SOL*

Formulas are built the same as in first-order logic. A signature consists of *function constants* and *predicate constants*. Function constants of arity 0 are called *object constants*, and predicate constants of arity 0 are called *propositional constants*.

Similar to circumscription, for predicate symbols (constants or variables) $v$ and $c$, expression $v \le c$ is defined as shorthand for $\forall \mathbf{x}(v(\mathbf{x}) \rightarrow c(\mathbf{x}))$. Expression $v = c$ is defined as $\forall \mathbf{x}(v(\mathbf{x}) \leftrightarrow c(\mathbf{x}))$ if $v$ and $c$ are predicate symbols, and $\forall \mathbf{x}(v(\mathbf{x}) = c(\mathbf{x}))$ if they are function symbols. For lists of symbols $\mathbf{v} = (v_1, \dots, v_n)$ and $\mathbf{c} = (c_1, \dots, c_n)$, expression $\mathbf{v} \le \mathbf{c}$ is defined as $(v_1 \le c_1) \land \cdots \land (v_n \le c_n)$, and similarly, expression $\mathbf{v} = \mathbf{c}$ is defined as $(v_1 = c_1) \land \cdots \land (v_n = c_n)$. 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 $\mathbf{c}$. Members of $\mathbf{c}$ are called *intensional* constants. By $\mathbf{c}^{pred}$ ($\mathbf{c}^{func}$, respectively) we mean the list of all predicate constants (function constants, respectively) in $\mathbf{c}$, and by $\widehat{\mathbf{c}}^{pred}$ ($\widehat{\mathbf{c}}^{func}$, respectively) we mean the list of the corresponding predicate variables (function variables, respectively) in $\widehat{\mathbf{c}}$.

For any first-order formula $F$, expression $\mathrm{SM}[F;\ \mathbf{c}]$ is defined as

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

where $\widehat{\mathbf{c}} < \mathbf{c}$ is shorthand for $(\widehat{\mathbf{c}}^{pred} \leq \mathbf{c}^{pred}) \wedge \neg(\widehat{\mathbf{c}} = \mathbf{c})$, and $F^*(\widehat{\mathbf{c}})$ is defined recursively as follows.

- When $F$ is an atomic formula, $F^*$ is $F(\widehat{\mathbf{c}}) \wedge F$, where $F(\widehat{\mathbf{c}})$ is obtained from $F$ by replacing all (function and predicate) constants from $\mathbf{c}$ occurring in $F$ with the corresponding (function and predicate) variables from $\widehat{\mathbf{c}}$;
- $(G \wedge H)^* = G^* \wedge H^*$; $\quad (G \vee H)^* = G^* \vee H^*$;
- $(G \rightarrow H)^* = (G^* \rightarrow H^*) \wedge (G \rightarrow H)$;
- $(\forall x G)^* = \forall x G^*$; $\quad (\exists x G)^* = \exists x G^*$.

(We understand $\neg F$ as shorthand for $F \rightarrow \bot$; and $\top$ as $\neg\bot$.)

When $F$ is a sentence (formula with no free variables), the models of $\mathrm{SM}[F;\mathbf{c}]$ are called the *stable* models of $F$ *relative to* $\mathbf{c}$. They are the models of $F$ that are “stable” on $\mathbf{c}$. This definition of a stable model is a proper generalization of the one from (Ferraris et al. 2011), which views logic programs as a special case of formulas.

We will often write the implication $F \rightarrow G$ in a rule form $G \leftarrow F$ as in logic programs. We often identify a program with a finite conjunction of universal closures of formulas.

*Example 1 continued. Consider the formula $F$ in Example 1 and an interpretation $I$ that has the set of nonnegative integers as the universe, interprets integers, arithmetic functions and comparison operators in the standard way, and has $FillUp^I = \mathrm{FALSE}$, $Amount_0^I = 6$, $Amount_1^I = 5$. One can check that $I$ is a model of $\mathrm{SM}[F; Amount_1]$. Consider another interpretation $I_1$ that agrees with $I$ except that $Amount_1^{I_1} = 8$. This is a model of $F$ but not of $\mathrm{SM}[F; Amount_1]$. Another interpretation $I_2$ that agrees with $I$ except that $FillUp^{I_2} = \mathrm{TRUE}$, $Amount_1^{I_2} = 10$ is a model of $F$ as well as a model of $\mathrm{SM}[F; Amount_1]$.*

### 2.2 Review: Bartholomew-Lee Semantics in Terms of Grounding and Reduct

#### 2.2.1 Infinitary Ground Formulas

Since the universe can be infinite, grounding a quantified sentence introduces infinite conjunctions and disjunctions over the elements in the universe. Here we rely on the concept of grounding *relative to an interpretation* from (Truszczynski 2012). The following is the definition of an *infinitary ground formula*, which is adapted from (Truszczynski 2012). One difference is that we do not replace ground terms with their corresponding object names, leaving them unchanged during grounding. This change is necessary in defining a reduct for functional stable model semantics.[1]

For each element $\xi$ in the universe $|I|$ of $I$, we introduce a new symbol $\xi^\diamond$, called an *object name*. By $\sigma^I$ we denote the signature obtained from $\sigma$ by adding all object names $\xi^\diamond$ as additional

[1] Another difference is that grounding in (Truszczynski 2012) refers to “infinitary propositional formulas,” which can be defined on any propositional signature. This generality is not essential for our purpose in this paper.

object constants. We will identify an interpretation $I$ of signature $\sigma$ with its extension to $\sigma^I$ defined by $I(\xi^\diamond) = \xi$.[2]

We assume the primary connectives to be $\bot$, $\{\}^\wedge$, $\{\}^\vee$, and $\rightarrow$. Propositional connectives $\wedge, \vee, \neg, \top$ are considered as shorthands: $F \wedge G$ as $\{F, G\}^\wedge$; $F \vee G$ as $\{F, G\}^\vee$. $\neg$ and $\top$ are defined as before.

Let $A$ be the set of all ground atomic formulas of signature $\sigma^I$. The sets $\mathcal{F}_0, \mathcal{F}_1, \dots$ are defined recursively as follows:

- $\mathcal{F}_0 = A \cup \{\bot\}$;
- $\mathcal{F}_{i+1} (i \geq 0)$ consists of expressions $\mathcal{H}^\vee$ and $\mathcal{H}^\wedge$, for all subsets $\mathcal{H}$ of $\mathcal{F}_0 \cup \ldots \cup \mathcal{F}_i$, and of the expressions $F \rightarrow G$, where $F, G \in \mathcal{F}_0 \cup \cdots \cup \mathcal{F}_i$.

We define $\mathcal{L}_A^{inf} = \bigcup_{i=0}^{\infty} \mathcal{F}_i$, and call elements of $\mathcal{L}_A^{inf}$ *infinitary ground formulas* of $\sigma$ w.r.t. $I$.

For any interpretation $I$ of $\sigma$ and any infinitary ground formula $F$ w.r.t. $I$, the definition of satisfaction, $I \models F$, is as follows:

- For atomic formulas, the definition of satisfaction is the same as in the standard first-order logic;
- $I \models \mathcal{H}^\vee$ if there is a formula $G \in \mathcal{H}$ such that $I \models G$;
- $I \models \mathcal{H}^\wedge$ if, for every formula $G \in \mathcal{H}$, $I \models G$;
- $I \models G \rightarrow H$ if $I \not\models G$ or $I \models H$.

### *2.2.2 Bartholomew-Lee Semantics in Terms of Grounding and Reduct*

Let $F$ be any first-order sentence of a signature $\sigma$, and let $I$ be an interpretation of $\sigma$. By $gr_I[F]$ we denote the infinitary ground formula w.r.t. $I$ that is obtained from $F$ by the following process:

- If $F$ is an atomic formula, $gr_I[F]$ is $F$;
- $gr_I[G \odot H] = gr_I[G] \odot gr_I[H] \quad (\odot \in \{\wedge, \vee, \rightarrow\})$;
- $gr_I[\exists x G(x)] = \{gr_I[G(\xi^\diamond)] \mid \xi \in |I|\}^\vee; \qquad gr_I[\forall x G(x)] = \{gr_I[G(\xi^\diamond)] \mid \xi \in |I|\}^\wedge$.

*Example 1 continued.* Consider again $F$ in Example 1, and the same interpretation $I$. $gr_I[F]$ is the following set of formulas.

$$\begin{gathered}(\mathit{Amount}_1 = 0) \vee \neg(\mathit{Amount}_1 = 0) \ \leftarrow\ \mathit{Amount}_0 = 0{+}1 \\ (\mathit{Amount}_1 = 1) \vee \neg(\mathit{Amount}_1 = 1) \ \leftarrow\ \mathit{Amount}_0 = 1{+}1 \\ \dots \\ \mathit{Amount}_1 = 10 \ \leftarrow\ \mathit{FillUp}\end{gathered}$$

For any two interpretations $I$, $J$ of the same signature and any list $\mathbf{c}$ of distinct predicate and function constants, we write $J <^{\mathbf{c}} I$ if

- $J$ and $I$ have the same universe and agree on all constants not in $\mathbf{c}$;
- $p^J \subseteq p^I$ for all predicate constants $p$ in $\mathbf{c}$; and
- $J$ and $I$ do not agree on $\mathbf{c}$.

The *reduct* $F^{\underline{I}}$ of an infinitary ground formula $F$ relative to an interpretation $I$ is defined as follows:

[2] For details, see (Lifschitz et al. 2008).

- For each atomic formula $F$, $F^{\underline{I}} = \bot$ if $I \not\models F$ and $F^{\underline{I}} = F$ otherwise;
- $(\mathcal{H}^\wedge)^{\underline{I}} = \bot$ if $I \not\models \mathcal{H}^\wedge$; otherwise $(\mathcal{H}^\wedge)^{\underline{I}} = \{G^{\underline{I}} \mid G \in \mathcal{H}\}^\wedge$;
- $(\mathcal{H}^\vee)^{\underline{I}} = \bot$ if $I \not\models \mathcal{H}^\vee$; otherwise $(\mathcal{H}^\vee)^{\underline{I}} = \{G^{\underline{I}} \mid G \in \mathcal{H}\}^\vee$;
- $(G \to H)^{\underline{I}} = \bot$ if $I \not\models G \to H$; otherwise $(G \to H)^{\underline{I}} = G^{\underline{I}} \to H^{\underline{I}}$.

*Theorem 1*
Let $F$ be a first-order sentence of signature $\sigma$ and let $\mathbf{c}$ be a list of intensional constants. For any interpretation $I$ of $\sigma$, $I \models \text{SM}[F;\mathbf{c}]$ iff

- $I$ satisfies $F$, and
- every interpretation $J$ such that $J <^{\mathbf{c}} I$ does not satisfy $(gr_I[F])^{\underline{I}}$.

*Example 1 continued.* *The reduct $(gr_I[F])^{\underline{I}}$ is equivalent to*

$$(Amount_1\!=\!5) \vee \bot \;\leftarrow\; Amount_0\!=\!5\!+\!1. \tag{2}$$

*No interpretation that is different from $I$ only on $Amount_1$ satisfies the reduct. On the other hand, the reduct $(gr_{I_1}[F])^{\underline{I_1}}$ is equivalent to $\bot \vee \neg\bot \;\leftarrow\; Amount_0 = 5\!+\!1$, and other interpretations that are different from $I_1$ only on $Amount_1$ satisfy the reduct.*

### 2.3 Review: Original Definition of the Cabalar Semantics

#### 2.3.1 Partial Interpretation

We first define the notion of a *partial* interpretation. Given a first-order signature $\sigma$ comprised of function and predicate constants, a *partial* interpretation $I$ of $\sigma$ consists of

- a non-empty set $|I|$, called the *universe* of $I$;
- for every function constant $f$ of arity $n$, a function $f^I$ from $(|I|\cup\{u\})^n$ to $|I|\cup\{u\}$, where $u$ is not in $|I|$ ("$u$" stands for *undefined*);
- for every predicate constant $p$ of arity $n$, a function $p^I$ from $(|I|\cup\{u\})^n$ to $\{\text{TRUE}, \text{FALSE}\}$.

For each term $f(t_1,\dots,t_n)$, we define

$$f(t_1,\dots,t_n)^I = \begin{cases} u & \text{if } t_i^I = u \text{ for some } i \in \{1,\dots,n\}; \\ f^I(t_1^I,\dots,t_n^I) & \text{otherwise.} \end{cases}$$

The satisfaction relation $\models_{\text{p}}$ between a partial interpretation $I$ and a first-order formula $F$ is the same as the one for first-order logic except for the following base cases:

- For each atomic formula $p(t_1,\dots,t_n)$,

$$p(t_1,\dots,t_n)^I = \begin{cases} \text{FALSE} & \text{if } t_i^I = u \text{ for some } i \in \{1,\dots,n\}; \\ p^I(t_1^I,\dots,t_n^I) & \text{otherwise.} \end{cases}$$

- For each atomic formula $t_1 = t_2$,

$$(t_1 = t_2)^I = \begin{cases} \text{TRUE} & \text{if } t_1^I \neq u,\ t_2^I \neq u, \text{ and } t_1^I = t_2^I; \\ \text{FALSE} & \text{otherwise.} \end{cases}$$

We say that $I \models_{\text{p}} F$ if $F^I = \text{TRUE}$.

Observe that under a partial interpretation, $t = t$ is not necessarily true: $I \not\models_{\text{p}} t = t$ iff $t^I = u$. On the other hand, $\neg(t_1 = t_2)$, also denoted by $t_1 \neq t_2$, is true under $I$ even when both $t_1^I$ and $t_2^I$ are mapped to the same $u$.

#### *2.3.2 Functional Equilibrium Models by Cabalar*

Given any two partial interpretations $J$ and $I$ of the same signature $\sigma$, and a set of constants $\mathbf{c}$, we write $J \preceq^{\mathbf{c}} I$ if

- $J$ and $I$ have the same universe and agree on all constants not in $\mathbf{c}$;
- $p^J \subseteq p^I$ for all predicate constants in $\mathbf{c}$; and
- $f^J(\boldsymbol{\xi}) = u$ or $f^J(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$ for all function constants in $\mathbf{c}$ and all lists $\boldsymbol{\xi}$ of elements in the universe.

We write $J \prec^{\mathbf{c}} I$ if $J \preceq^{\mathbf{c}} I$ but not $I \preceq^{\mathbf{c}} J$. Note that $J \prec^{\mathbf{c}} I$ is defined similar to $J <^{\mathbf{c}} I$ (Section 2.2.2) except for the treatment of functions.

A *PHT-interpretation* ("Partial HT-interpretation") $\mathcal{I}$ of signature $\sigma$ is a tuple $\langle \mathcal{I}^h, \mathcal{I}^t \rangle$ such that $\mathcal{I}^h$ and $\mathcal{I}^t$ are partial interpretations of $\sigma$ that have the same universe.

The *satisfaction* relation $\models_{\text{pht}}$ between a PHT-interpretation $\mathcal{I}$, a world $w \in \{h, t\}$ ordered by $h < t$ , and a first-order sentence $F$ of the signature $\sigma$ is defined recursively:

- If $F$ is an atomic formula, $\mathcal{I}, w \models_{\text{pht}} F$ if $\mathcal{I}^w \models_{\text{p}} F$;
- $\mathcal{I}, w \models_{\text{pht}} F \wedge G$ if $\mathcal{I}, w \models_{\text{pht}} F$ and $\mathcal{I}, w \models_{\text{pht}} G$;
- $\mathcal{I}, w \models_{\text{pht}} F \vee G$ if $\mathcal{I}, w \models_{\text{pht}} F$ or $\mathcal{I}, w \models_{\text{pht}} G$;
- $\mathcal{I}, w \models_{\text{pht}} F \rightarrow G$ if, for every world $w'$ such that $w \leq w'$, $\mathcal{I}, w' \not\models_{\text{pht}} F$ or $\mathcal{I}, w' \models_{\text{pht}} G$;
- $\mathcal{I}, w \models_{\text{pht}} \forall x F(x)$ if, for every $\xi \in |\mathcal{I}|$, $\mathcal{I}, w \models_{\text{pht}} F(\xi^\diamond)$;
- $\mathcal{I}, w \models_{\text{pht}} \exists x F(x)$ if, for some $\xi \in |\mathcal{I}|$, $\mathcal{I}, w \models_{\text{pht}} F(\xi^\diamond)$.

We say that an HT-interpretation $\mathcal{I}$ *satisfies* $F$, written as $\mathcal{I} \models_{\text{pht}} F$, if $\mathcal{I}, h \models_{\text{pht}} F$.

A PHT-interpretation $\mathcal{I} = \langle I, I \rangle$ of signature $\sigma$ is a *partial equilibrium model* of a sentence $F$ relative to $\mathbf{c}$ if

- $\langle I, I \rangle \models_{\text{pht}} F$, and
- for every partial interpretation $J$ such that $J \prec^{\mathbf{c}} I$, we have $\langle J, I \rangle \not\models_{\text{pht}} F$.

## 3 Reformulations

### *3.1 Cabalar Semantics in Terms of Grounding and Reduct*

The Cabalar semantics can also be reformulated in terms of grounding and reduct. A theorem similar to Theorem 1 can be stated for the Cabalar semantics.

*Theorem 2*
Let $F$ be a first-order sentence of signature $\sigma$ and let $\mathbf{c}$ be a list of intensional constants. For any partial interpretation $I$ of $\sigma$, $\langle I, I \rangle$ is a partial equilibrium model of $F$ iff

- $I \models_{\text{p}} F$, and
- for every partial interpretation $J$ of $\sigma$ such that $J \prec^{\mathbf{c}} I$, we have $J \not\models_{\text{p}} gr_I[F]^{\underline{I}}$.

*Example 1 continued. Consider the same $F$, $I$, and the reduct $gr_I[F]^{\underline{I}}$, which is equivalent to* (2). *If we view $I$ as a partial interpretation, there is only one partial interpretation $J$ such that $J \prec^c I$, which agrees with $I$ except $Amount_1^J = u$. Clearly, $J$ does not satisfy the reduct. In accordance with Theorem 2, $\langle I, I \rangle$ is a partial equilibrium model of $F$.*

Interestingly, this reformulation of the Cabalar semantics is closely related to the language ASP$\{f\}$ (Balduccini 2012b). We discuss the details in Section 6.

Comparing the reformulation of the Cabalar semantics in Theorem 2 and the reformulation of the Bartholomew-Lee semantics in Theorem 1 tells us that the reducts are defined in the same way, whereas interpretations we consider for stability checking and the notions of satisfaction are different. That is, if the intensional constants are function constants only, under the Bartholomew-Lee semantics, the interpretations $J$ we consider for stability checking are all other classical interpretations that are different from $I$, while under the Cabalar semantics, they are partial interpretations that are "smaller" than $I$. For instance, in Example 1, there are multiple such $J$s for the Bartholomew-Lee semantics, while there is only one such $J$ for the Cabalar semantics.

In Section 4, we present some syntactic classes of formulas on which the two semantics coincide despite these differences.

### *3.2 Cabalar Semantics in Terms of Second-Order Logic*

The Cabalar semantics can also be formulated in the style of second-order logic. We extend the formulas to allow predicate and function variables as in the standard second-order logic, but consider partial interpretations in place of classical interpretations. We define $\widehat{\mathbf{c}} \preceq \mathbf{c}$ as

$$(\widehat{\mathbf{c}}^{pred} \leq \mathbf{c}^{pred}) \wedge (\widehat{\mathbf{c}}^{func} \leq \mathbf{c}^{func}) \ ,$$

where $\widehat{\mathbf{c}}^{pred} \leq \mathbf{c}^{pred}$ is defined as in Section 2.1, and $\widehat{\mathbf{c}}^{func} \leq \mathbf{c}^{func}$ is defined as the conjunction of

$$\forall \mathbf{x}((\widehat{f}(\mathbf{x}) \neq \widehat{f}(\mathbf{x})) \vee (\widehat{f}(\mathbf{x}) = f(\mathbf{x}))) \ .$$

for all function constants $f$ in $\mathbf{c}^{func}$ and the corresponding function variables $\widehat{f}$ in $\widehat{\mathbf{c}}^{func}$. As we explained earlier, the first disjunctive term is satisfiable under a partial interpretation, meaning that $\widehat{f}$ is undefined on $\mathbf{x}$; the second disjunctive term means that $\widehat{f}$ and $f$ are both defined on $\mathbf{x}$ and map to the same element in the universe. We define $\widehat{\mathbf{c}} \prec \mathbf{c}$ as $(\widehat{\mathbf{c}} \preceq \mathbf{c}) \wedge \neg(\mathbf{c} \preceq \widehat{\mathbf{c}})$.

We reformulate the Cabalar semantics by using the expression CBL that looks similar to SM. It is defined as:

$$\mathrm{CBL}[F; \mathbf{c}] = F \wedge \neg \exists \widehat{\mathbf{c}}(\widehat{\mathbf{c}} \prec \mathbf{c} \wedge F^{\dagger}(\widehat{\mathbf{c}})) \ ,$$

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

- When $F$ is an atomic formula, $F^{\dagger}(\widehat{\mathbf{c}})$ is $F(\widehat{\mathbf{c}})$ (as defined in Section 2.1). [3]

The following theorem states the correctness of the reformulation.

*Theorem 3*
For any sentence $F$, a PHT-interpretation $\langle I, I \rangle$ is a partial equilibrium model of $F$ relative to $\mathbf{c}$ iff $I \models_{\mathrm{p}} \mathrm{CBL}[F; \mathbf{c}]$.

Note the similarity between this reformulation of the Cabalar semantics given in Theorem 3 and the definition of SM in Section 2.1. The differences are in the comparison operators $\prec$ vs. $<$, and whether to consider partial interpretations or classical interpretations.

[3] In fact, $F^{*}(\widehat{\mathbf{c}})$ can be also used in place of $F^{\dagger}(\widehat{\mathbf{c}})$ for defining $\mathrm{CBL}[F; \mathbf{c}]$ as well, without affecting the models.

### *3.3 Bartholomew-Lee Semantics in Terms of HT-Logic*

The Bartholomew-Lee semantics can be reformulated in terms of a modification to equilibrium logic, similar to the way the Cabalar semantics is defined in (Cabalar 2011), and is also reviewed in Section 2.3

An *FHT-interpretation* ("Functional HT-interpretation") $\mathcal{I}$ of signature $\sigma$ is a tuple $\langle \mathcal{I}^h, \mathcal{I}^t \rangle$ such that $\mathcal{I}^h$ and $\mathcal{I}^t$ are classical interpretations of $\sigma$ that have the same universe. The satisfaction relation $\models_{\rm fht}$ between an FHT-interpretation $\mathcal{I}$, a world $w \in \{h, t\}$ ordered by $h < t$, and a first-order sentence of signature $\sigma$ is defined in the same way as $\models_{\rm pht}$ for PHT-interpretations in Section 2.3 except for the base case:

- If $F$ is an atomic formula, $\mathcal{I}, w \models_{\rm fht} F$ if, for every world $w'$ such that $w \leq w'$, $\mathcal{I}^{w'} \models F$.

We say that FHT-interpretation $\mathcal{I}$ satisfies $F$, written as $\mathcal{I} \models_{\rm fht} F$, if $\mathcal{I}, h \models_{\rm fht} F$.

The following theorem asserts the correctness of the reformulation of the Bartholomew-Lee semantics in terms of equilibrium logic style.

*Theorem 4*
Let $F$ be a first-order sentence of signature $\sigma$ and let $\mathbf{c}$ be a list of predicate and function constants. For any interpretation $I$ of $\sigma$, $I \models \text{SM}[F; \mathbf{c}]$ iff

- $\langle I, I \rangle \models_{\rm fht} F$, and
- for every interpretation $J$ of $\sigma$ such that $J <^{\mathbf{c}} I$, we have $\langle J, I \rangle \not\models_{\rm fht} F$.

## 4 Comparing Bartholomew-Lee Semantics and Cabalar Semantics

Neither semantics is stronger than the other. The following example presents a formula that has a stable model under the Cabalar semantics, but not under the Bartholomew-Lee semantics.

*Example 2*
$\text{SM}[f = g;\ f, g]$ has no models if the universe contains more than one element. Take any $I$ such that $I \models f = g$. The reduct of $f = g$ relative to $I$ is $f = g$ itself, and there are other models of the reduct. Since $I$ is not the unique model of the reduct, $I$ is not a $(f, g)$-stable model of $f = g$. On the other hand, assuming that the universe is $\{1, 2, 3\}$, an interpretation $I$ that assigns 1 to both $f$ and $g$ satisfies $\text{CBL}[f = g;\ f, g]$. The reduct is the same as before, but any interpretation $J$ smaller than $I$ maps either or both $f$ and $g$ to $u$, and hence does not satisfy the reduct. Similarly, there are two other models of $\text{CBL}[f = g;\ f, g]$ with the same universe.

On the other hand, in the following example, the formula has a stable model under the Bartholomew-Lee semantics, but not under the Cabalar semantics.

*Example 3*
Let $F$ be the formula $f(1) = 1 \wedge f(2) = 1 \wedge (f(g) = 1 \rightarrow g = 1)$, and $I$ be an interpretation such that the universe is $\{1, 2\}$, and $1^I = 1$, $2^I = 2$, $f(1)^I = 1$, $f(2)^I = 1$, $g^I = 1$. One can check that $I$ is a model of $\text{SM}[F;\ f, g]$, but not a model of $\text{CBL}[F;\ f, g]$.

### *4.1 Coincidence on* $\mathbf{c}$*-plain formulas*

This section presents a syntactic class of formulas, called "$\mathbf{c}$-plain," on which the Bartholomew-Lee semantics and the Cabalar semantics coincide when we consider "total" interpretations only.

A partial interpretation $I$ is called *total* if $I$ does not map any function constant to $u$. Obviously, a total interpretation can be identified with the classical interpretation.

For any function constant $f$, a first-order formula $F$ is called $f$*-plain* if each atomic formula in $F$

- does not contain $f$, or
- is of the form $f(\mathbf{t}) = t_1$ where $\mathbf{t}$ is a list of terms not containing $f$, and $t_1$ is a term not containing $f$.

For example, $f=1$ is $f$-plain, but each of $p(f)$, $g(f) = 1$, and $1=f$ is not $f$-plain.

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 $\mathbf{c}$. Roughly speaking, $\mathbf{c}$-plain formulas do not allow the functions in $\mathbf{c}$ to be nested in another predicate or function, and at most one function in $\mathbf{c}$ is allowed in each atomic formula. For example, the formula in (1) in Example 1 is $\mathit{Amount}_1$-plain; $f = g$ (Example 2) is not $(f, g)$-plain (because it is not $g$-plain), and neither is $f(g) = 1 \rightarrow g = 1$ (Example 3).

The following theorem states that the two semantics coincide on $\mathbf{c}$-plain formulas.

*Theorem 5*
For any $\mathbf{c}$-plain sentence $F$ of signature $\sigma$, any list $\mathbf{c}$ of intensional constants, and any total interpretation $I$ of $\sigma$ satisfying $\exists xy(x \neq y)$, $I \models \mathrm{SM}[F; \mathbf{c}]$ iff $I \models_{\mathrm{p}} \mathrm{CBL}[F; \mathbf{c}]$.

In accordance with the theorem, we already noted that the two semantics coincide on formula (1). Examples 2 and 3 above demonstrate why the restriction to $\mathbf{c}$-plain formulas is necessary in Theorem 5. This theorem is useful in relating several mathematical results established for the Bartholomew-Lee semantics to the Cabalar semantics as we will see in Section 5 and Appendix B.

The requirement in Theorem 5 that every occurrence of every atomic formula be $\mathbf{c}$-plain can be relaxed if the formula is tight.[4] An occurrence of a symbol or a subformula in a formula $F$ is called *strictly positive* in $F$ if that occurrence is not in the antecedent of any implication in $F$. We say that a formula is *head*-$\mathbf{c}$*-plain* if every strictly positively occurring atomic formula is $\mathbf{c}$-plain. For instance, $f(g)=1 \rightarrow h=1$ is head-$(f, g, h)$-plain, though it is not $(f, g, h)$-plain.

*Theorem 6*
For any head-$\mathbf{c}$-plain sentence $F$ of signature $\sigma$ that is tight on $\mathbf{c}$, and any total interpretation $I$ of $\sigma$ satisfying $\exists xy(x \neq y)$, $I \models \mathrm{SM}[F; \mathbf{c}]$ iff $I \models_{\mathrm{p}} \mathrm{CBL}[F; \mathbf{c}]$.

### *4.2 Different Behaviors for Nested Functions*

Theorem 5 can be extended to non-$\mathbf{c}$-plain formulas by first unfolding $F$, which, roughly speaking, moves nested functions outside by introducing existential quantifiers and variables. The process of *unfolding* $F$ *w.r.t.* $\mathbf{c}$, denoted by $\mathit{UF}_{\mathbf{c}}(F)$, is formally defined as follows.

- If $F$ is of the form $p(t_1, \dots, t_n)$ $(n \geq 0)$ such that $t_{k_1}, \dots, t_{k_j}$ are all the terms in $t_1, \dots, t_n$ that contain some members of $\mathbf{c}$, then $\mathit{UF}_{\mathbf{c}}(p(t_1, \dots, t_n))$ is
$$\exists x_1 \dots x_j \Big( p(t_1, \dots, t_n)'' \wedge \bigwedge_{1 \leq i \leq j} \mathit{UF}_{\mathbf{c}}(t_{k_i} = x_i) \Big)$$

[4] Tight formulas are defined in (Bartholomew and Lee 2013) and also reviewed in Appendix A.

where $p(t_1,\dots,t_n)''$ is obtained from $p(t_1,\dots,t_n)$ by replacing each $t_{k_i}$ with the variable $x_i$.

- If $F$ is of the form $f(t_1,\dots,t_n)=t_0$ ($n\ge 0$) such that $t_{k_1},\dots,t_{k_j}$ are all the terms in $t_0,\dots,t_n$ that contain some members of $\mathbf{c}$, then $\mathit{UF}_{\mathbf{c}}(f(t_1,\dots,t_n)=t_0)$ is

$$\exists x_1\dots x_j\Big((f(t_1,\dots,t_n)=t_0)''\wedge\bigwedge_{0\le i\le j}\mathit{UF}_{\mathbf{c}}(t_{k_i}=x_i)\Big)$$

where $(f(t_1,\dots,t_n)=t_0)''$ is obtained from $f(t_1,\dots,t_n)=t_0$ by replacing each $t_{k_i}$ with the variable $x_i$.
- $\mathit{UF}_{\mathbf{c}}(F\odot G)$ is $\mathit{UF}_{\mathbf{c}}(F)\odot\mathit{UF}_{\mathbf{c}}(G)$ where $\odot\in\{\wedge,\vee,\rightarrow\}$.
- $\mathit{UF}_{\mathbf{c}}(QxF)$ is $Qx\ \mathit{UF}_{\mathbf{c}}(F(x))$ where $Q\in\{\forall,\exists\}$.

For example, $\mathit{UF}_{(f,g)}(f=g)$ is $\exists xy(x=y\wedge f=x\wedge g=y)$.

It is clear that $\mathit{UF}_{\mathbf{c}}(F)$ is equivalent to $F$ under classical logic. Similarly, Theorem 7 below shows that the Cabalar semantics preserves stable models when unfolding is applied. However, this is not the case under the Bartholomew-Lee semantics.

*Theorem 7*
For any sentence $F$, any list $\mathbf{c}$ of constants, and any partial interpretation $I$, we have $I\models_{\rm p}$ CBL$[F;\mathbf{c}]$ iff $I\models_{\rm p}$ CBL$[\mathit{UF}_{\mathbf{c}}(F);\mathbf{c}]$.

*Example 4*
Let $F$ be $f=g$. Recall that $\mathit{UF}_{\mathbf{c}}(F)$ is $\exists xy(x=y\wedge f=x\wedge g=y)$. Let $I_1$, $I_2$, $I_3$ be interpretations whose universe is $\{1,2,3\}$, and each $I_i$ maps $f$ and $g$ to $i$ ($1\le i\le 3$). Each of them satisfies CBL$[F;\ f,g]$ and CBL$[\mathit{UF}_{(f,g)}(F);\ f,g]$, but as we observed, none of them is a model of SM$[F;\ f,g]$.

However, since $\mathit{UF}_{\mathbf{c}}(F)$ is $\mathbf{c}$-plain, the following corollary follows from Theorems 5 and 7.

*Corollary 1*
For any sentence $F$, any list $\mathbf{c}$ of constants, and any total interpretation $I$ satisfying $\exists xy(x\ne y)$, we have $I\models_{\rm p}$ CBL$[F;\mathbf{c}]$ iff $I\models_{\rm p}$ CBL$[\mathit{UF}_{\mathbf{c}}(F);\mathbf{c}]$ iff $I\models$ SM$[\mathit{UF}_{\mathbf{c}}(F);\mathbf{c}]$.

For example, SM$[\mathit{UF}_{(f,g)}(f=g);\ f,g]$ has the same models as CBL$[f=g;\ f,g]$.

## 5 Relating the Cabalar Semantics to General Stable Models

Corollary 1 tells us that the results established for the Bartholomew-Lee semantics can be transferred to the Cabalar semantics as long as we are interested in total interpretations only. For instance, (Bartholomew and Lee 2012) shows a method of eliminating intensional function constants in favor of intensional predicate constants; (Bartholomew and Lee 2013) shows that, for tight programs, the stable model semantics and completion coincide. These results can be extended to the Cabalar semantics as well by first rewriting the formula to be $\mathbf{c}$-plain by applying unfolding (Theorem 7), and then applying Corollary 1 since the two semantics coincide on $\mathbf{c}$-plain formulas.

Below we show how to turn formulas under the Cabalar semantics into formulas under the stable model semantics from (Ferraris et al. 2011). The method is similar to the one from (Bartholomew and Lee 2012). This is done by eliminating intensional functions under the Cabalar semantics in favor of intensional predicates.

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 a new intensional predicate constant $p$ of arity $n+1$, where $n$ is the arity of $f$;
- replace each subformula $f(\mathbf{t}) = t'$ in $F$ with $p(\mathbf{t}, t')$.

*Theorem 8*
For any $f$-plain sentence $F$ and any partial interpretation $I$, if $I \models_{\rm p} \forall \mathbf{x} y(p(\mathbf{x}, y) \leftrightarrow f(\mathbf{x}) = y)$, then $I \models_{\rm p} \text{CBL}[F;\ f, \mathbf{c}]$ iff $I \models_{\rm p} \text{CBL}[F^f_p;\ p, \mathbf{c}]$.

By $UC_p$, we denote the following formulas that enforce the partial functional image on the predicate $p$:

$$\forall \mathbf{x} yz(y \neq z \wedge p(\mathbf{x}, y) \wedge p(\mathbf{x}, z) \rightarrow \bot), \tag{3}$$

where $\mathbf{x}$ is a $n$-tuple of variables, and all variables in $\mathbf{x}$, $y$, and $z$ are pairwise distinct. Note that each formula is a constraint,[5] which can only remove stable models when it is added.

The following corollary shows that there is a simple 1–1 correspondence between the models of $F$ and the models of $F^f_p$. Recall that the signature of $F^f_p$ is obtained from the signature of $F$ by replacing $f$ with $p$. For any interpretation $I$ of the signature of $F$, by $I^f_p$ we denote the interpretation of the signature of $F^f_p$ 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 $I$ such that $f^I(\xi_1, \dots, \xi_n) \neq u$. The notations are straightforwardly extended to $F^{\mathbf{f}}_{\mathbf{p}}$ and $I^{\mathbf{f}}_{\mathbf{p}}$ where $\mathbf{f}$ and $\mathbf{p}$ are lists of function and predicate constants.

*Corollary 2*
Let $F$ be an $f$-plain sentence. (a) For any partial interpretation $I$ of the signature of $F$, $I \models_{\rm p} \text{CBL}[F;\ f, \mathbf{c}]$ iff $I^f_p \models_{\rm p} \text{CBL}[F^f_p \wedge UC_p;\ p, \mathbf{c}]$. (b) For any partial interpretation $J$ of the signature of $F^f_p$, $J \models_{\rm p} \text{CBL}[F^f_p \wedge UC_p;\ p, \mathbf{c}]$ iff $J = I^f_p$ for some partial interpretation $I$ such that $I \models_{\rm p} \text{CBL}[F;\ f, \mathbf{c}]$.

Repeated applications of Corollary 2 tells us that all intensional functions under the Cabalar semantics can be eliminated in favor of intensional predicates, which essentially reduces the Cabalar semantics to the first-order stable model semantics defined in (Ferraris et al. 2011). In the following $\mathbf{c}^{\mathbf{f}}_{\mathbf{p}}$ denotes the list of constants where every member of $\mathbf{f}$ in $\mathbf{c}$ is replaced with a new predicate constant in $\mathbf{p}$.

*Corollary 3*
Let $\mathbf{c}$ be a set of intensional constants consisting of intensional function constants $\mathbf{f}$ and intensional predicate constants, and let $F$ be an $\mathbf{c}$-plain sentence. (a) For any total interpretation $I$ of the signature of $F$, $I \models_{\rm p} \text{CBL}[F; \mathbf{c}]$ iff $I^{\mathbf{f}}_{\mathbf{p}} \models \text{SM}[F^{\mathbf{f}}_{\mathbf{p}} \wedge UC_{\mathbf{p}}; \mathbf{c}^{\mathbf{f}}_{\mathbf{p}}]$. (b) For any total interpretation $J$ of the signature of $F^{\mathbf{f}}_{\mathbf{p}}$, $J \models \text{SM}[F^{\mathbf{f}}_{\mathbf{p}} \wedge UC_{\mathbf{p}}; \mathbf{c}^{\mathbf{f}}_{\mathbf{p}}]$ iff $J = I^{\mathbf{f}}_{\mathbf{p}}$ for some total interpretation $I$ such that $I \models_{\rm p} \text{CBL}[F; \mathbf{c}]$.

[5] A rule with the empty head, or a formula that has no strictly positive occurrence of an atom (Ferraris et al. 2011).

Corollary 3 generalizes Theorem 1 from (Cabalar 2011), which restricted $F$ to be in the syntax of "FLP rules." The corollary is similar to Corollary 2 from (Bartholomew and Lee 2012), which shows how to turn $\mathbf{c}$-plain formulas under the Bartholomew-Lee semantics to formulas under the first-order stable model semantics. On the other hand, since any formula under the Cabalar semantics can be turned into a $\mathbf{c}$-plain formula (Corollary 1), Corollary 3 can be extended to arbitrary formulas.

## 6 Comparing Cabalar Semantics and Balduccini Semantics

Due to lack of space, we refer the reader to the online appendix for the review of the Balduccini Semantics.

It turns out that the Balduccini semantics is closely related to the Cabalar semantics. This is shown by reformulating the Balduccini semantics using the notion of partial interpretations and partial satisfaction. We identify a consistent set of seed literals $I$ with a partial interpretation that maps all object constants in $\sigma \setminus \mathbf{c}$ to themselves. For example, for signature $\sigma = \{f, g, 1, 2\}$ where $f, g \in \mathbf{c}$, we identify the consistent set of seed literals $I = \{f = 1\}$ with the partial interpretation $I$ such that $f^I = 1, g^I = u, 1^I = 1, 2^I = 2$.

The following theorem states that, in the absence of strong negation, Balduccini semantics can be viewed as a special, ground case of the Cabalar semantics.

*Theorem 9*
For any ASP$\{$f$\}$ program $\Pi$ with intensional constants $\mathbf{c}$ and any consistent set $I$ of seed literals, if $\Pi$ contains no strong negation, then $I$ is a Balduccini answer set of $\Pi$ iff $I \models_{\text{p}} \text{CBL}[\Pi; \mathbf{c}]$.

Theorem 9 can be extended to full ASP$\{$f$\}$ programs that contain strong negation. Since the language in (Cabalar 2011) does not allow strong negation, this requires us to eliminate strong negation. It is well known that strong negation in front of standard atoms can be eliminated using new atoms.

In order to eliminate strong negation in front of t-atoms, by $\Pi^{\#}$ we denote the program obtained from $\Pi$ by replacing $\sim(f = g)$ with $(f = f) \wedge (g = g) \wedge \neg(f = g)$. As we noted earlier, this formula is true iff $f^I$ and $g^I$ are defined, and have different values. This is the same understanding as the construct $f \# g$ in (Cabalar 2011).

*Theorem 10*
For any ASP$\{$f$\}$ program $\Pi$ with intensional constants $\mathbf{c}$ and any consistent set $I$ of seed literals, $I$ is a Balduccini answer set of $\Pi$ iff $I$ is a Balduccini answer set of $\Pi^{\#}$.

## 7 Conclusion

We presented several reformulations of functional stable models—in terms of second-order logic, in terms of grounding and reduct, and in terms of variants of the logic of here-and-there. The reformulations helped us compare them and identify the relationships between them. The functional stable model semantics by Bartholomew and Lee is simpler as it does not need to rely on the extra notion of partial satisfaction, but is limited to total interpretations only. On the other hand, the Cabalar semantics and its special case, the Balduccini semantics, allow functions to be undefined at the price of relying on a rather complicated non-standard definition of partial satisfaction. Nevertheless, all three semantics coincide on large syntactic classes of formulas.

**Acknowledgements:** We are grateful to Pedro Cabalar and the anonymous referees for their useful comments. This work was partially supported by the National Science Foundation under Grant IIS-0916116 and by the South Korea IT R&D program MKE/KIAT 2010-TD-300404-001.

## References

BALDUCCINI, M. 2012a. An answer set solver for non-Herbrand programs: Progress report. In *ICLP (Technical Communications)*. 49–60.

BALDUCCINI, M. 2012b. A "conservative" approach to extending answer set programming with non-Herbrand functions. In *Correct Reasoning - Essays on Logic-Based AI in Honour of Vladimir Lifschitz*. 24–39.

BARTHOLOMEW, M. AND LEE, J. 2012. Stable models of formulas with intensional functions. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 2–12.

BARTHOLOMEW, M. AND LEE, J. 2013. Functional stable model semantics and answer set programming modulo theories. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*. To appear.

CABALAR, P. 2011. Functional answer set programming. *TPLP 11,* 2-3, 203–233.

FERRARIS, P., LEE, J., AND LIFSCHITZ, V. 2011. Stable models and circumscription. *Artificial Intelligence 175*, 236–263.

LIFSCHITZ, V. 2012. Logic programs with intensional functions. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 24–31.

LIFSCHITZ, V., MORGENSTERN, L., AND PLAISTED, D. 2008. Knowledge representation and classical logic. In *Handbook of Knowledge Representation*, F. van Harmelen, V. Lifschitz, and B. Porter, Eds. Elsevier, 3–88.

TRUSZCZYNSKI, M. 2012. Connecting first-order ASP and the logic FO(ID) through reducts. In *Correct Reasoning - Essays on Logic-Based AI in Honour of Vladimir Lifschitz*. 543–559.