Omega-Completeness of the Logic of Here-and-There and Strong Equivalence of Logic Programs

Jorge Fandinno[1] , **Vladimir Lifschitz**[2]
[1]University of Nebraska Omaha, USA
[2]University of Texas at Austin, USA
jfandinno@unomaha.edu, vl@utexas.edu

Abstract

Theory of strongly equivalent transformations is an essential part of the methodology of representing knowledge in answer set programming. Strong equivalence of two programs can be sometimes characterized as the possibility of deriving the rules of each program from the rules of the other in some deductive system. This paper describes a system with this property for the language mini-GRINGO. The key to the proof is an ω-completeness theorem for the many-sorted logic of here-and-there.

1 Introduction

In answer set programming, two sets of rules are considered strongly equivalent if, informally speaking, they have the same meaning in any context. This equivalence relation has been extensively studied in the literature. Understanding its properties is important because it may help the programmer recognize the possibility of simpifying a rule, or a group of rules, within a program so that the set of stable models is not affected.

Strong equivalence of two programs can be sometimes established by deriving the rules of each program from the rules of the other in an appropriate deductive system (Lifschitz, Pearce, and Valverde 2001; Lifschitz, Pearce, and Valverde 2007; Harrison et al. 2017). The deductive system *HTA* ("here-and-there with arithmetic") allows us to apply this method to programs in the answer set programming language mini-GRINGO (Fandinno et al. 2020, Section 5); (Lifschitz 2021, Section 2.1). Two programs in this language are strongly equivalent to each other if the first-order sentences obtained from them by applying the syntactic transformation τ^* can be derived from each other in *HTA* (Lifschitz 2021, Section 4).

The converse does not hold, however: mini-GRINGO programs Π_1, Π_2 may be strongly equivalent to each other even though the deductive possibilities of *HTA* are not sufficient for establishing the equivalence between $\tau^*\Pi_1$ and $\tau^*\Pi_2$ (Lifschitz 2021, Section 6). Extending *HTA* that would allow us to replace the result of that paper by an if-and-only-if condition is posed there as a topic for future work.

In this paper we show that this goal can be achieved using rules with infinitely many premises, similar to the ω-rule in arithmetic,

$$\frac{F(0) \quad F(1) \quad \ldots}{\forall n F(n)}.$$

This theorem closes a gap in our understanding of strong equivalence for programs containing operations on integers.

The key to the proof is an ω-completeness theorem for the many-sorted logic of here-and-there—an assertion similar to the ω-completeness property of classical logic, established by Henkin (1954). (Many-sorted languages are relevant here because the language of *HTA* has variables of two sorts, *general* and *integer*.) The proof extends Henkin's construction, which involves an omitting types theorem (Kiesler 1977, Section 6.15), to the many-sorted logic of here-and-there. Omitting types in the context of intuitionistic and intermediate logics was earlier explored by Marković (1979, 1995) and by Bagheri and Pourmahdian (2011).

We start by presenting background material related to mini-GRINGO, many-sorted languages and the translation τ^* (Section 2). Then we describe an extension of the first-order logic of here-and-there (Pearce and Valverde 2004; Ferraris, Lee, and Lifschitz 2011) to many-sorted formulas (Section 3) and state a theorem that relates strong equivalence of mini-GRINGO programs to the translation τ^* (Section 4). The main results of the paper—the ω-completeness theorem and its application to the study of strong equivalence—are presented in Section 5. Most proofs are relegated to Section 6.

2 Preliminaries

2.1 Programs

We assume that three countably infinite sets of symbols are selected: *numerals*, *symbolic constants*, and *variables*. We assume that a 1-1 correspondence between numerals and integers is chosen; the numeral corresponding to an integer n is denoted by $\overline{n}$. *Precomputed terms* are numerals and symbolic constants. We assume that a total order on precomputed terms is chosen such that for all integers m and n, $\overline{m} < \overline{n}$ iff $m < n$.

Terms allowed in a mini-GRINGO program are formed from precomputed terms and variables using the absolute value symbol[1] $|\,|$ and six binary operation names

$$+ \quad - \quad \times \quad / \quad \backslash \quad ..$$

(the last three serve to represent integer division, modulo and intervals). An *atom* is a symbolic constant optionally followed by a tuple of terms in parentheses. A *literal* is an atom possibly preceded by one or two occurrences of *not*. A *comparison* is an expression of the form $t_1\ rel\ t_2$, where t_1, t_2 are terms and rel is $=$ or one of the comparison symbols

$$\neq \quad < \quad > \quad \leq \quad \geq \tag{1}$$

A *rule* is an expression of the form *Head* $\leftarrow$ *Body*, where

- *Body* is a conjunction (possibly empty) of literals and comparisons, and
- *Head* is either an atom, or an atom in braces (then this is a *choice rule*), or empty (then this is a *constraint*).

A *(mini-*GRINGO*) program* is a finite set of rules.

The semantics of ground terms is defined by assigning to every ground term t the finite set $[t]$ of its *values* (Lifschitz, Lühne, and Schaub 2019, Section 3). Values of a ground term are precomputed terms. For instance,

$$[\overline{2}/\overline{3}] = \{\overline{0}\},\ [\overline{2}/\overline{0}] = \emptyset,\ [\overline{0}\,..\,\overline{2}] = \{\overline{0}, \overline{1}, \overline{2}\}.$$

A *predicate symbol* is a pair p/n, where p is a symbolic constant, and n is a nonnegative integer.

Stable models of a program are defined as stable models of the set of propositional formulas[2] obtained from it by a syntactic transformation denoted by τ (Lifschitz, Lühne, and Schaub 2019, Section 3). Atomic parts of these formulas are *precomputed atoms*—atoms $p(\mathbf{t})$ such that the members of the tuple $\mathbf{t}$ are precomputed terms. For example, τ transforms the rule

$$\{q(X)\} \leftarrow p(X) \tag{2}$$

into the set of formulas $p(t) \rightarrow (q(t) \vee \neg q(t))$ for all precomputed terms t. The rule

$$q(\overline{0}\,..\,\overline{2}) \leftarrow not\ p \tag{3}$$

is transformed into $\neg p \rightarrow (q(\overline{0}) \wedge q(\overline{1}) \wedge q(\overline{2}))$. Thus stable models of mini-GRINGO programs are sets of precomputed atoms.

2.2 Many-sorted Theories

A *(many-sorted) signature* consists of symbols of three kinds—*sorts*, *function constants*, and *predicate constants*. A reflexive and transitive *subsort* relation $\preceq$ is defined on the set of sorts. A tuple $s_1, \dots, s_n$ ($n \geq 0$) of *argument sorts* is assigned to every function constant and to every predicate constant; in addition, a *value sort* is assigned to every function constant. Function constants with $n = 0$ are called *object constants*.

We assume that for every sort, an infinite sequence of *object variables* of that sort is chosen. *Terms* over a signature σ are defined recursively:

- object constants and object variables of a sort s are terms of sort s;
- if f is a function constant with argument sorts $s_1, \dots, s_n$ ($n > 0$) and value sort s, and $t_1, \dots, t_n$ are terms such that the sort of t_i is a subsort of s_i ($i = 1, \dots, n$), then $f(t_1, \dots, t_n)$ is a term of sort s.

The sort of a term t will be denoted by sort(t). *Atomic formulas* over σ are

- expressions of the form $p(t_1, \dots, t_n)$, where p is a predicate constant with argument sorts $s_1, \dots, s_n$, and $t_1, \dots, t_n$ are terms such that sort$(t_i) \preceq s_i$, and
- expressions of the form $t_1 = t_2$, where t_1 and t_2 are terms.

Formulas over σ are formed from atomic formulas and the 0-place connective $\bot$ (falsity) using the binary connectives $\wedge$, $\vee$, $\rightarrow$ and the quantifiers $\forall$, $\exists$. The other connectives are treated as abbreviations: $\neg F$ stands for $F \rightarrow \bot$ and $F \leftrightarrow G$ stands for $(F \rightarrow G) \wedge (G \rightarrow F)$.

A *sentence* is a formula without free variables. A *theory* over σ is a set T of sentences over σ, which are called the *axioms* of T.

An *interpretation* I of a signature σ assigns

- a non-empty *domain* $|I|^s$ to every sort s of σ, so that $|I|^{s_1} \subseteq |I|^{s_2}$ whenever $s_1 \preceq s_2$,
- a function f^I from $|I|^{s_1} \times \cdots \times |I|^{s_n}$ to $|I|^s$ to every function constant f with argument sorts $s_1, \dots, s_n$ ($n \geq 0$) and value sort s, and
- a truth-valued function p^I on $|I|^{s_1} \times \cdots \times |I|^{s_n}$ to every predicate constant p with argument sorts $s_1, \dots, s_n$.

If I is an interpretation of a signature σ then by σ^I we denote the signature obtained from σ by adding, for every element d of a domain $|I|^s$, its *name* d^*_s as an object constant of sort s. The interpretation I is extended to σ^I by defining $(d^*_s)^I = d$. We will drop the subscript s in d^*_s when it is clear from context. The value t^I assigned by an interpretation I of σ to a ground term t over σ^I and the satisfaction relation $\models$ between an interpretation of σ and a sentence over σ^I are defined recursively, in the usual way (Lifschitz, Morgenstern, and Plaisted 2008, Section 1.2.2).

If $\mathbf{d}$ is a tuple $d_1, \dots, d_n$ of elements of domains of I then $\mathbf{d}^*$ stands for the tuple $d^*_1, \dots, d^*_n$ of their names. If $\mathbf{t}$ is a tuple $t_1, \dots, t_n$ of ground terms then $\mathbf{t}^I$ stands for the tuple $t^I_1, \dots, t^I_n$ of values assigned to them by I.

For example, the signature σ_0 includes

- the sort *general* and its subsort *integer*;
- all precomputed terms of mini-GRINGO (Section 2.1) as object constants; an object constant is assigned the sort *integer* iff it is a numeral;
- the symbol $|\,|$ as a unary function constant; its argument and value have the sort *integer*;
- the symbols $+$, $-$ and $\times$ as binary function constants; their arguments and values have the sort *integer*;
- predicate symbols p/n as n-ary predicate constants; their arguments have the sort *general*;

[1] The absolute value symbol was not included in previous publications on mini-GRINGO.

[2] The definition of a stable model (Gelfond and Lifschitz 1988) was extended to sets of propositional formulas by Ferraris (2005).

- the symbols
$$\neq \quad < \quad > \quad \leq \quad \geq \tag{4}$$
as binary predicate constants; their arguments have the sort *general*.

A formula of the form $(p/n)(\mathbf{t})$ can be written also as $p(\mathbf{t})$. This convention allows us to view precomputed atoms (Section 2.1) as sentences over σ_0. Conjunctions of equalities and inequalities can be abbreviated as usual in algebra; for instance, $X = Y < Z$ stands for $X = Y \wedge Y < Z$.

We are interested in the interpretations of σ_0 that are *standard* in the sense that

- the domain of the sort *general* is the set of precomputed terms;
- the domain of the sort *integer* is the set of numerals;
- every object constant represents itself;
- the absolute value symbol and the binary function constants are interpreted as usual in arithmetic;
- predicate constants (4) are interpreted in accordance with the total order on precomputed terms chosen in the definition of mini-GRINGO (Section 2.1).

2.3 Representing Rules by Formulas

We assume that every symbol designated as a mini-GRINGO variable is among general variables of the signature σ_0.

For every mini-GRINGO term t, we will now define a formula $val_t(Z)$ over the signature σ_0, where Z is a general variable that does not occur in t. That formula expresses, informally speaking, that Z is one of the values of t. The definition is recursive:

- if t is a precomputed term or a variable then $val_t(Z)$ is $Z = t$,
- if t is $|t_1|$ then $val_t(Z)$ is $\exists I(val_{t_1}(I) \wedge Z = |I|)$,
- if t is t_1 *op* t_2, where *op* is $+$, $-$, or $\times$ then $val_t(Z)$ is
$$\exists IJ(val_{t_1}(I) \wedge val_{t_2}(J) \wedge Z = I \; op \; J),$$
- if t is $t_1 \,/\, t_2$ then $val_t(Z)$ is
$$\begin{aligned}\exists IJK(&val_{t_1}(I) \wedge val_{t_2}(J)\\ &\wedge K \times |J| \leq |I| < (K + \overline{1}) \times |J|\\ &\wedge ((I \times J \geq \overline{0} \wedge Z = K)\\ &\quad \vee (I \times J < \overline{0} \wedge Z = -K))),\end{aligned}$$
- if t is $t_1 \backslash t_2$ then $val_t(Z)$ is
$$\begin{aligned}\exists IJK(&val_{t_1}(I) \wedge val_{t_2}(J)\\ &\wedge K \times |J| \leq |I| < (K + \overline{1}) \times |J|\\ &\wedge ((I \times J \geq \overline{0} \wedge Z = I - K \times J)\\ &\quad \vee (I \times J < \overline{0} \wedge Z = I + K \times J))),\end{aligned}$$
- if t is $t_1 \,..\, t_2$ then $val_t(Z)$ is
$$\exists IJK(val_{t_1}(I) \wedge val_{t_2}(J) \wedge I \leq K \leq J \wedge Z = K),$$
where I, J, K are fresh integer variables.[3]

[3] The use of the absolute value sign in two of these formulas is motivated by the fact that the grounder GRINGO (Gebser et al. 2019) truncates the quotient toward zero, instead of applying the floor function. This feature of GRINGO was not taken into account in earlier publications (Gebser et al. 2015, Section 4.2), (Lifschitz, Lühne, and Schaub 2019, Section 6), (Fandinno et al. 2020, Section 3).

If $\mathbf{t}$ is a tuple $t_1, \dots, t_n$ of mini-GRINGO terms, and $\mathbf{Z}$ is a tuple $Z_1, \dots, Z_n$ of distinct general variables, then $val_{\mathbf{t}}(\mathbf{Z})$ stands for the conjunction $val_{t_1}(Z_1) \wedge \cdots \wedge val_{t_n}(Z_n)$.

The translation τ^B, described below, transforms literals and comparisons into formulas over the signature σ_0. (The superscript B reflects the fact that this translation is close to the meaning of expressions in *bodies* of rules.)

- $\tau^B(p(\mathbf{t}))$ is $\exists \mathbf{Z}(val_{\mathbf{t}}(\mathbf{Z}) \wedge p(\mathbf{Z}))$;
- $\tau^B(\textit{not}\ p(\mathbf{t}))$ is $\exists \mathbf{Z}(val_{\mathbf{t}}(\mathbf{Z}) \wedge \neg p(\mathbf{Z}))$;
- $\tau^B(\textit{not not}\ p(\mathbf{t}))$ is $\exists \mathbf{Z}(val_{\mathbf{t}}(\mathbf{Z}) \wedge \neg\neg p(\mathbf{Z}))$;
- $\tau^B(t_1 \ rel \ t_2)$ is
$$\exists Z_1 Z_2(val_{t_1}(Z_1) \wedge val_{t_2}(Z_2) \wedge Z_1 \ rel \ Z_2).$$

If *Body* is a conjunction $B_1 \wedge B_2 \wedge \cdots$ of literals and comparisons then $\tau^B(\textit{Body})$ stands for the conjunction $\tau^B(B_1) \wedge \tau^B(B_2) \wedge \cdots$.

We are ready now to define the operator τ^*. This operator converts a basic rule
$$p(\mathbf{t}) \leftarrow \textit{Body} \tag{5}$$
into the sentence
$$\widetilde{\forall}(val_{\mathbf{t}}(\mathbf{Z}) \wedge \tau^B(\textit{Body}) \rightarrow p(\mathbf{Z})),$$
where $\mathbf{Z}$ is a tuple of fresh general variables, and $\widetilde{\forall}$ denotes universal closure. A choice rule
$$\{p(\mathbf{t})\} \leftarrow \textit{Body}$$
is converted into
$$\widetilde{\forall}(val_{\mathbf{t}}(\mathbf{Z}) \wedge \tau^B(\textit{Body}) \rightarrow p(\mathbf{Z}) \vee \neg p(\mathbf{Z})),$$
and a constraint $\leftarrow$ *Body* becomes $\widetilde{\forall}\neg\tau^B(\textit{Body})$.

For example, τ^* transforms rule (2) into the sentence
$$\begin{aligned}\forall X Z_1(Z_1 = X \wedge \exists Z_2(Z_2 = X \wedge p(Z_2))\\ \rightarrow q(Z_1) \vee \neg q(Z_1)),\end{aligned} \tag{6}$$
and (3) into
$$\begin{aligned}\forall Z(\exists IJK(I = \overline{0} \wedge J = \overline{2} \wedge I \leq K \leq J \wedge Z = K) \wedge \neg p\\ \rightarrow q(Z)).\end{aligned} \tag{7}$$

For any program Π, $\tau^*\Pi$ stands for the conjunction of first-order sentences $\tau^* R$ for all rules R of Π.

3 Many-sorted Logic of Here-and-There

In the rest of the paper, σ is a countable many-sorted signature with its predicate constants partitioned into two (possibly empty) subsets—*intensional* and *extensional*. For any interpretation I of σ, I^{int} stands for the set of atomic formulas of the form $p(\mathbf{d}^*)$, where p is an intensional symbol and $\mathbf{d}$ is a tuple of elements of the domains of I corresponding to the argument sorts of p, such that $I \models p(\mathbf{d}^*)$.

An *HT-interpretation* of σ is a pair $\langle \mathcal{H}, I \rangle$, where I is an interpretation of σ, and $\mathcal{H}$ is a subset of I^{int}. (In terms of Kripke models with two worlds, I is the there-world, and $\mathcal{H}$ describes the intensional predicates in the here-world). The satisfaction relation $\models_{ht}$ between an HT-interpretation $\langle \mathcal{H}, I \rangle$ of σ and a sentence F over σ^I is defined recursively as follows:

- $\langle \mathcal{H}, I\rangle \models_{ht} p(\mathbf{t})$, where p is intensional, if $p((\mathbf{t}^I)^*) \in \mathcal{H}$;
- $\langle \mathcal{H}, I\rangle \models_{ht} p(\mathbf{t})$, where p is extensional, if $I \models p(\mathbf{t})$;
- $\langle \mathcal{H}, I\rangle \models_{ht} t_1 = t_2$ if $t_1^I = t_2^I$;
- $\langle \mathcal{H}, I\rangle \not\models_{ht} \bot$;
- $\langle \mathcal{H}, I\rangle \models_{ht} F \wedge G$ if $\langle \mathcal{H}, I\rangle \models_{ht} F$ and $\langle \mathcal{H}, I\rangle \models_{ht} G$;
- $\langle \mathcal{H}, I\rangle \models_{ht} F \vee G$ if $\langle \mathcal{H}, I\rangle \models_{ht} F$ or $\langle \mathcal{H}, I\rangle \models_{ht} G$;
- $\langle \mathcal{H}, I\rangle \models_{ht} F \rightarrow G$ if
  (i) $\langle \mathcal{H}, I\rangle \not\models_{ht} F$ or $\langle \mathcal{H}, I\rangle \models_{ht} G$, and
  (ii) $I \models F \rightarrow G$;
- $\langle \mathcal{H}, I\rangle \models_{ht} \forall X\, F(X)$ if $\langle \mathcal{H}, I\rangle \models_{ht} F(d^*)$ for each d in $|I|^{\mathrm{sort}(X)}$;
- $\langle \mathcal{H}, I\rangle \models_{ht} \exists X\, F(X)$ if $\langle \mathcal{H}, I\rangle \models_{ht} F(d^*)$ for some d in $|I|^{\mathrm{sort}(X)}$.

This relation is monotonic, in the sense that $\langle \mathcal{H}, I\rangle \models_{ht} F$ implies $I \models F$ (by induction on the size of F). The converse holds if F does not contain intensional symbols.

An *HT-model* of a theory T is an HT-interpretation that satisfies all sentences in T. If T is a theory and F is a sentence over σ, then we write $T \models_{ht} F$ to express that every HT-model of T satisfies F.

4 Strong Equivalence

Mini-GRINGO programs Π_1 and Π_2 are *strongly equivalent* to each other if, for every set Ω of propositional combinations of precomputed atoms, $\tau\Pi_1 \cup \Omega$ has the same stable models as $\tau\Pi_2 \cup \Omega$. This condition implies that for every mini-GRINGO program Π, the program $\Pi_1 \cup \Pi$ has the same stable models as $\Pi_2 \cup \Pi$ (take $\Omega = \tau\Pi$).

For instance, rule (2) is strongly equivalent to the rule

$$q(X) \leftarrow p(X) \wedge \textit{not not}\ q(X). \tag{8}$$

It follows that replacing rule (2) by (8) within any program preserves the set of stable models. Rule (3) is strongly equivalent to the group of three rules

$$q(\overline{0}) \leftarrow \textit{not}\ p,\ q(\overline{1}) \leftarrow \textit{not}\ p,\ q(\overline{2}) \leftarrow \textit{not}\ p. \tag{9}$$

We will return to these examples in Section 5.5.

Theorem 1 below shows that strong equivalence of mini-GRINGO programs can be characterized in terms of HT-interpretations of the signature σ_0. For this signature, predicate constants (4) are classified as extensional, and predicate constants of the form p/n are intensional. An HT-interpretation $\langle \mathcal{H}, I\rangle$ of σ_0 is *standard* if I is standard.

Theorem 1. *Mini-GRINGO programs Π_1, Π_2 are strongly equivalent iff the formula $\tau^*\Pi_1 \leftrightarrow \tau^*\Pi_2$ is satisfied by all standard HT-interpretations of σ_0.*

5 ω-Completeness

5.1 Many-Sorted *SQHT*$^=$

For the special case when the signature σ has a single sort, and each of its predicate symbols is intensional, Lifschitz, Pearce, and Valverde (2007) defined a deductive system that is sound and complete with respect to the semantics described in Section 3. Theorem 2 below extends that result to the general case.

Consider first a natural deduction system of many-sorted intuitionistic logic. The derivable objects of this system *Int* are *sequents*—expressions $\Gamma \Rightarrow F$, in which Γ is a finite set of formulas over σ (“assumptions”), and F is a formula over σ. We write sets of assumptions as lists. A sequent of the form $\Rightarrow F$ will be identified with the formula F.

The axiom schemas of *Int* are $F \Rightarrow F$ and $t = t$. The inference rules of *Int* are the usual inference rules of propositional logic (Lifschitz, Morgenstern, and Plaisted 2008, Figure 1.1) and rules for quantifiers and equality shown in Figure 1.

The deductive system *SQHT*$^=$ is the result of extending *Int* by four axiom schemas:

$$F \vee (F \rightarrow G) \vee \neg G, \tag{10}$$

$$\exists X (F(X) \rightarrow \forall X\, F(X)), \tag{11}$$

$$X = Y \vee X \neq Y \tag{12}$$

and

$$p(\mathbf{X}) \vee \neg p(\mathbf{X}) \tag{13}$$

for all extensional precicate symbols p, where $\mathbf{X}$ is a tuple of pairwise distinct variables of appropriate sorts. Schema (10), known as the Hosoi axiom (Hosoi 1966), is useful primarily because of its intuitionistic consequence

$$\neg F \vee \neg\neg F, \tag{14}$$

known as the weak law of excluded middle. (Take G in (10) to be $\neg F$.)

For any theory T and and any formula F, we write $T \vdash F$ if F is derivable from the axioms of T in *SQHT*$^=$.

Theorem 2. *For any theory T and any sentence F over σ, $T \vdash F$ iff $T \models_{ht} F$.*

5.2 ω-Interpretations

Let S be a subset of the set of sorts of σ. We assume that for every sort s in S, $\omega(s)$ is a non-empty subset of the set of ground terms t such that $\mathrm{sort}(t) \preceq s$. An interpretation I of σ is an *ω-interpretation* if for every s in S and every d in $|I|^s$ there exists a term t in $\omega(s)$ such that $t^I = d$.

In the case of the signature σ_0 we define:

- S is $\{\textit{general}, \textit{integer}\}$;
- $\omega(\textit{general})$ is the set of precomputed terms;
- $\omega(\textit{integer})$ is the set of numerals.

Theorem 3. *For any interpretation I of σ_0, the following conditions are equivalent:*

(a) I is isomorphic to a standard interpretation;

(b) I is an ω-interpretation and satisfies

(b1) the formulas $c_1 \neq c_2$ for all pairs c_1, c_2 of distinct precomputed terms;

(b2) all formulas of the forms

$$c_1\ \textit{rel}\ c_2, \quad \neg(c_1\ \textit{rel}\ c_2),$$

where c_1, c_2 are precomputed terms and rel is one of symbols (4), that are true for the total order chosen in the definition of mini-GRINGO;

$$(\forall I)\ \frac{\Gamma \Rightarrow F(X)}{\Gamma \Rightarrow \forall X\, F(X)} \qquad (\forall E)\ \frac{\Gamma \Rightarrow \forall X\, F(X)}{\Gamma \Rightarrow F(t)}$$

where X is not free in Γ

$$(\exists I)\ \frac{\Gamma \Rightarrow F(t)}{\Gamma \Rightarrow \exists X\, F(X)} \qquad (\exists E)\ \frac{\Gamma \Rightarrow \exists X\, F(X) \quad \Delta, F(X) \Rightarrow G}{\Gamma, \Delta \Rightarrow G}$$

where sort$(t) \preceq$ sort(X) and t is free for X in $F(X)$ — where X is not free in Δ, G

$$(Eq) \quad \frac{\Gamma \Rightarrow t_1 = t_2 \quad \Delta \Rightarrow F(t_1)}{\Gamma, \Delta \Rightarrow F(t_2)} \quad \frac{\Gamma \Rightarrow t_1 = t_2 \quad \Delta \Rightarrow F(t_2)}{\Gamma, \Delta \Rightarrow F(t_1)}$$

where sort$(t_1) \preceq$ sort(X), sort$(t_2) \preceq$ sort(X), and t_1, t_2 are free for X in $F(X)$

Figure 1: Inference rules for quantifiers and equality

(b3) the formulas

$$\overline{m+n} = \overline{m}+\overline{n}; \quad \overline{m-n} = \overline{m}-\overline{n}; \quad \overline{m\times n} = \overline{m}\times\overline{n}$$

for all pairs m, n of integers.

Proof. The implication from (a) to (b) is obvious. If I satisfies (b) then the function $c \mapsto c^I$ an isomorphism between a standard interpretation and I. □

5.3 Deductive System *SQHT*$^\omega$

An *ω-model* of a theory T is an HT-model $\langle \mathcal{H}, I\rangle$ of T such that I is an ω-interpretation. Theorem 2 shows that the deductive system *SQHT*$^=$ matches the semantics based on HT-models of a theory. We would like to extend that system so that it will match the semantics based on ω-models.

The theorem stated below shows that this can be accomplished by adding the inference rule

$$\frac{\Gamma \Rightarrow F(t) \text{ for all terms } t \text{ in } \omega(\text{sort}(X))}{\Gamma \Rightarrow \forall X\, F(X)} \tag{15}$$

where sort$(X) \in S$. The deductive system obtained from *SQHT*$^=$ by adding this rule will be denoted by *SQHT*$^\omega$.

Theorem 4. *For any theory T and any sentence F over σ, F is derivable in SQHT$^\omega$ from the axioms of T iff every ω-model of T satisfies F.*

In case of the signature σ_0, inference rule (15) can be represented as a pair of rules:

$$\frac{\Gamma \Rightarrow F(t) \text{ for all precomputed terms } t}{\Gamma \Rightarrow \forall X\, F(X)}$$

where X is a general variable, and

$$\frac{\Gamma \Rightarrow F(\overline{n}) \text{ for all integers } n}{\Gamma \Rightarrow \forall N\, F(N)} \tag{16}$$

where N is an integer variable.

Theorem 5. *For any theory T over σ_0, a sentence F over σ_0 is satisfied by all standard HT-models of T iff F is derivable in SQHT$^\omega$ from the axioms of T and formulas (b1)–(b3).*

Proof. From Theorem 3 we can conclude that F is satisfied by all standard HT-models of T iff F is satisfied by all ω-models $\langle \mathcal{H}, I\rangle$ of T such that I satisfies formulas (b1)–(b3). Since these formulas do not contain intensional symbols, they are satisfied by I iff they are satisfied by $\langle \mathcal{H}, I\rangle$. The assertion to be proved follows by Theorem 4 applied to the theory obtained from T by adding axioms (b1)–(b3). □

5.4 Application to Strong Equivalence

From Theorem 1 and Theorem 5 with empty T we conclude:

Theorem 6. *Mini-*GRINGO *programs Π_1, Π_2 are strongly equivalent iff the formula $\tau^*\Pi_1 \leftrightarrow \tau^*\Pi_2$ is derivable in SQHT$^\omega$ from formulas (b1)–(b3).*

The if-part of this assertion is stronger than the similar property of the deductive system *HTA* (Lifschitz 2021, Section 4), because every formula provable in *HTA* can be derived in *SQHT*$^\omega$ from formulas (b1)–(b3), but not the other way around. Consider, for instance, the program Π_1 consisting of the rules

$$p(\overline{0}),$$
$$p(X+\overline{1}) \leftarrow p(X)$$

and the program Π_2, obtained from Π_1 by adding the rule

$$p(X) \leftarrow X + \overline{1} > \overline{0}.$$

These programs are strongly equivalent, but the formula $\tau^*\Pi_1 \leftrightarrow \tau^*\Pi_2$ is not provable in *HTA* in this case (Lifschitz 2021, Section 6). The reason is that the set of postulates of *HTA* does not include induction axioms for formulas that contain intensional symbols. Such an axiom

$$G(\overline{0}) \wedge \forall N(G(N) \to G(N+\overline{1})) \to \forall N(N \geq \overline{0} \to G(N))$$

can be derived, however, in *SQHT*$^\omega$ from formulas (b1)–(b3) using rule (16) with

$$G(\overline{0}) \wedge \forall N(G(N) \to G(N+\overline{1}))$$

as Γ, and with $N \geq \overline{0} \to G(N)$ as $F(N)$. The premise

$$\Gamma \Rightarrow \overline{n} \geq \overline{0} \to G(\overline{n})$$

for negative n follows from the formula $\neg(\overline{n} \geq \overline{0})$, which belongs to (b2). For nonnegative n, it can be derived from the sequents

$$\begin{array}{l}\Gamma \Rightarrow G(\overline{0}),\\ \Gamma \Rightarrow G(\overline{0}) \to G(\overline{0}+\overline{1}),\\ \Gamma \Rightarrow G(\overline{1}) \to G(\overline{1}+\overline{1}),\\ \dots\\ \Gamma \Rightarrow G(\overline{n-1}) \to G(\overline{n-1}+\overline{1})\end{array}$$

and the formulas

$$\overline{1} = \overline{0}+\overline{1},\ \dots,\ \overline{n} = \overline{n-1}+\overline{1},$$

which belong to (b3).

5.5 Examples

Example 1: Π_1 is rule (2); Π_2 is rule (8). According to Theorem 6, the claim that these rules are strongly equivalent can be justified by deriving the equivalence between the result (6) of applying τ^* to Π_1 and the result

$$\begin{array}{r}\forall X Z_1(Z_1 = X \wedge \exists Z_2(Z_2 = X \wedge p(Z_2))\\ \wedge\, \exists Z_3(Z_3 = X \wedge \neg\neg q(Z_3))\\ \to q(Z_1))\end{array} \tag{17}$$

of applying τ^* to Π_2 using postulates of the deductive system *SQHT*$^\omega$ and assumptions (b1)–(b3). This equivalence can be actually proved in *SQHT*$^=$. Indeed, formula (6) is intuitionistically equivalent to

$$\forall X(p(X) \to q(X) \vee \neg q(X));$$

formula (17) is intuitionistically equivalent to

$$\forall X(p(X) \to (\neg\neg q(X) \to q(X)).$$

The equivalence between the consequents

$$q(X) \vee \neg q(X) \text{ and } \neg\neg q(X) \to q(X)$$

of these implications is provable in *SQHT*$^=$, because it is an intuitionistic consequence of weak excluded middle (14) with $q(X)$ as F.

Example 2: We will use Theorem 6 to check that rule (3) is strongly equivalent to rule (9). The result (7) of applying τ^* to (3) is intuitionistically equivalent to

$$\neg p \to \forall K(\overline{0} \leq K \leq \overline{2} \to q(K)).$$

The result of applying τ^* to (9) is intuitionistically equivalent to

$$\neg p \to \forall K(K = \overline{0} \vee K = \overline{1} \vee K = \overline{2} \to q(K)).$$

It remains to note that the equivalence

$$\forall K(\overline{0} \leq K \leq \overline{2} \leftrightarrow K = \overline{0} \vee K = \overline{1} \vee K = \overline{2})$$

can be derived from assumptions (b1), (b2) using rule (16).

6 Proofs

6.1 Proof of Theorem 1

The proof refers to infinitary propositional logic of here-and-there (Harrison et al. 2017, Section 2.3) for formulas built from precomputed atoms. Thus we distinguish between HT-interpretations $\langle \mathcal{H}, I\rangle$ of σ_0 and *propositional HT-interpretations*—pairs $\langle \mathcal{H}, \mathcal{T}\rangle$, where $\mathcal{H}$, $\mathcal{T}$ are sets of precomputed atoms and $\mathcal{H} \subseteq \mathcal{T}$. Harrison et al. defined strong equivalence for infinitary formulas and proved the following fact (Harrison et al. 2017, Theorem 3):

Lemma 1. *Two infinitary propositional formulas are strongly equivalent iff they are satisfied by the same propositional HT-interpretations.*

The first component $\mathcal{H}$ of a standard HT-interpretation $\langle \mathcal{H}, I\rangle$ of σ_0 is the set of all atoms $p(\mathbf{t}^*)$ such that $p(\mathbf{t})$ is a precomputed atom satisfied by I. The correpondence between the tuples $\mathbf{t}$ and $\mathbf{t}^*$ is one-to-one, and we will take the liberty to identify them. Then we can say that propositional HT-interpretations can be characterized as pairs $\langle \mathcal{H}, I^{int}\rangle$, where I is a standard interpretation of σ_0, and $\mathcal{H}$ is a subset of I^{int}.

The proof of Theorem 1 refers also to the translation $F \mapsto F^{\rm prop}$ (Lifschitz, Lühne, and Schaub 2019, Section 5), which transforms sentences over σ_0 into infinitary propositional formulas. This translation is defined as follows:

- if F is $p(\mathbf{t})$ then $F^{\rm prop}$ is obtained from F by replacing each member of $\mathbf{t}$ by the value obtained after evaluating all arithmetic functions;
- if F is t_1 *rel* t_2 then $F^{\rm prop}$ is $\top$ if the values of t_1 and t_2 are in the relation *rel*, and $\bot$ otherwise;
- $\bot^{\rm prop}$ is $\bot$;
- $(F \odot G)^{\rm prop}$ is $F^{\rm prop} \odot G^{\rm prop}$ for every binary connective $\odot$;
- $(\forall X\, F(X))^{\rm prop}$ is the conjunction of the formulas $F(t)^{\rm prop}$ over all precomputed terms t if X is a variable of the sort *general*, and over all numerals t if X is a variable of the sort *integer*;
- $(\exists X\, F(X))^{\rm prop}$ is the disjunction of the formulas $F(t)^{\rm prop}$ over all precomputed terms t if X is a variable of the sort *general*, and over all numerals t if X is a variable of the sort *integer*.

This translation is similar to the grounding process defined by Truszczynski (2012, Section 2), and the lemma below is similar to Propositions 2 and 4 from that paper.

Lemma 2. *For any standard interpretation I of σ_0 and any sentence F over σ_0^I,*

(i) I satisfies F iff I^{int} satisfies $F^{\rm prop}$;

(ii) for any subset $\mathcal{H}$ of I^{int}, the HT-interpretation $\langle \mathcal{H}, I\rangle$ of σ_0 satisfies F iff the propositional HT-interpretation $\langle \mathcal{H}, I^{int}\rangle$ satisfies $F^{\rm prop}$.

Proof. Part (i) is proved by induction on the size of F. Part (ii) is proved by induction on the size of F using part (i). ☐

Proof of Theorem 1. The condition

$$\Pi_1 \text{ is strongly equivalent to } \Pi_2$$

holds iff

$$(\tau^*\Pi_1)^{\mathrm{prop}} \text{ is strongly equivalent to } (\tau^*\Pi_2)^{\mathrm{prop}}$$

(Lifschitz, Lühne, and Schaub 2019, Proposition 4). The latter is equivalent to the condition

$$(\tau^*\Pi_1 \leftrightarrow \tau^*\Pi_2)^{\mathrm{prop}} \text{ is satisfied}$$
$$\text{by all propositional HT-interpretations}$$

(Lemma 1) and, by Lemma 2(ii), to the condition

$$\tau^*\Pi_1 \leftrightarrow \tau^*\Pi_2 \text{ is satisfied}$$
$$\text{by all standard HT-interpretations of } \sigma_0.$$

□

6.2 Proof of Theorem 2: Soundness

To prove the soundness of *SQHT*$^=$, we extend the definition of entailment to sequents as follows: we write

$$T \models_{ht} \Gamma \Rightarrow F$$

if

$$T \models_{ht} \widetilde{\forall}(\Gamma^\wedge \to F),$$

where $\Gamma^\wedge$ is the conjunction of all formulas in Γ, and $\widetilde{\forall}$ denotes universal closure. The soundness of *SQHT*$^=$ is proved by verifying that

(i) every axiom of *SQHT*$^=$ is satisfied by all HT-interpretations, and

(ii) whenever a sequent S is derived from sequents $S_1, \ldots, S_k$ by one application of an inference rule of *Int*, every HT-interpretation satisfying $S_1, \ldots, S_k$ satisfies S also.

The proof of (ii) for rules $(\forall E)$ and $(\exists I)$ uses the following lemma, which is easy to verify by induction:

Lemma 3. *For any formula $F(X)$ that has no free variables other than X, for any ground term t such that* sort$(t) \preceq$ sort(X), *and for any HT-interpretation $\langle \mathcal{H}, I\rangle$,*

$$\langle \mathcal{H}, I\rangle \models_{ht} F(t) \ \textit{iff}\ \langle \mathcal{H}, I\rangle \models_{ht} F\left(\left(t^I\right)^*\right).$$

6.3 Proof of Theorem 2: Completeness

The proof is similar to the proof of a special case due to Lifschitz, Pearce, and Valverde (2007).

Lemma 4.

$$\begin{array}{rl} (i) & \vdash \neg\forall X\, F(X) \leftrightarrow \exists X \neg F(X). \\ (ii) & \vdash \neg\neg\forall X\, F(X) \leftrightarrow \forall X \neg\neg F(X). \\ (iii) & \vdash \neg\neg\exists X\, F(X) \leftrightarrow \exists X \neg\neg F(X). \end{array}$$

Proof. (i) The implication left-to-right is an intuitionistic consequence of axiom (11). The implication right-to-left is provable intuitionistically. (ii) This is an intuitionistic consequence of (i). (iii) In (ii), take $F(X)$ to be $\neg F(X)$ and note that $\forall X \neg$ is intuitionistically equivalent to $\neg\exists X$. □

For any theory T and any sentence F, we write $T \vdash_c F$ if F is derivable from the axioms of T classically, that is, derivable in the extension of *SQHT*$^=$ obtained by replacing axiom schemas (10)–(13) with the law of the excluded middle

$$F \vee \neg F$$

for all formulas F.

Lemma 5. *(i) For any formula F,*

$$\vdash_c F \ \textit{iff}\ \vdash \neg\neg F.$$

(ii) For any theory T,

$$T \vdash_c \bot \ \textit{iff}\ T \vdash \bot.$$

Proof. (i) The if part is obvious. Only if: consider Gödel's negative translation F^{neg} of F, which is defined recursively:

- $F^{neg} = \neg\neg F$ if F is atomic;
- $\bot^{neg} = \bot$;
- $(F \wedge G)^{neg} = F^{neg} \wedge G^{neg}$;
- $(F \vee G)^{neg} = \neg(\neg F^{neg} \wedge \neg G^{neg})$;
- $(F \to G)^{neg} = F^{neg} \to G^{neg}$;
- $(\forall X\, F(X))^{neg} = \forall X(F(X)^{neg})$;
- $(\exists X\, F(X))^{neg} = \neg\forall X \neg F(X)^{neg}$.

If $\vdash_c F$ then F^{neg} is provable in *Int* (Mints 2000, Theorem 13.1 extended to the many-sorted case). It remains to show that $\vdash F^{neg} \leftrightarrow \neg\neg F$ for all F. The proof is by induction on F. Consider the case of $\forall X\, F(X)$. From the induction hypothesis

$$\vdash F(X)^{neg} \leftrightarrow \neg\neg F(X)$$

we need to derive

$$\vdash \forall X(F(X)^{neg}) \leftrightarrow \neg\neg\forall X\, F(X).$$

This is immediate from Lemma 4(ii). For the other cases, we only need deductive means of intuitionistic logic.

(ii) The if part is obvious. Only if: we can assume without loss of generality that T is finite, because any classical derivation of F from T uses only finitely many elements of T. If $T \vdash_c \bot$ then $\vdash_c \neg T^\wedge$. By part (i) of the lemma, $\vdash \neg\neg\neg T^\wedge$, so that $\vdash \neg T^\wedge$ and consequently $T \vdash \bot$. □

Given a theory T and a sentence F over σ such that $T \not\vdash F$, we need to construct a counterexample, that is, an HT-interpretation $\langle \mathcal{H}, I\rangle$ that satisfies all formulas in T but does not satisfy F.

By σ' we denote the signature obtained from σ by adding, for every sort s, a countable set C_s of object constants of that sort.

Lemma 6. *There exists a theory T' over σ' such that*

(α) *$T \subseteq T'$,*

(β) *$F \notin T'$,*

(γ) *T' is closed under $\vdash$,*

(δ) *for any sentence of the form $G \vee H$ in T', $G \in T'$ or $H \in T'$,*

(ϵ) *for any sentence of the form $\exists X\, F(X)$ in T' there exists an object constant c in $C_{\mathrm{sort}(X)}$ such that $F(c) \in T'$.*

Proof. Let E_0 be the set of all sentences of the form $\exists X G(X)$ over σ', and let D_0 be the set of all sentences of the form $G \vee H$ over σ'. Define T_0 to be T. We will define sets T_n, E_n, D_n for all positive n recursively in such a way that T_{n+1} will be obtained from T_n by adding one sentence so that, for all n, $T_n \not\vdash F$; E_{n+1} will be obtained from E_n by removing at most one sentence; and D_{n+1} will be obtained from D_n by removing at most one sentence. For each of the sets E_0, D_0, choose an enumeration of its elements.

Case 1: n is even. Let $\exists X G(X)$ be the first sentence from E_n such that $T_n \vdash \exists X G(X)$. (Such a sentence exists because E_0 contains infinitely many sentences with this property, and E_n is obtained from E_0 by removing finitely many sentences.) Let c be a constant from $C_{\text{sort}(s)}$ that occurs neither in T_n nor in $G(X)$. (Such a constant exists because T_n and $G(X)$ contain finitely many constants from $C_{\text{sort}(s)}$.) Then $T_{n+1} = T_n \cup \{G(c)\}$,

$$E_{n+1} = E_n \setminus \{\exists X G(X)\},\ D_{n+1} = D_n.$$

To show that the property $T_n \not\vdash F$ is preserved, assume that $T_{n+1} \vdash F$. Then $T_n \vdash G(c) \rightarrow F$. We can conclude that $T_n \vdash G(X) \rightarrow F$. (Take a derivation of $G(c) \rightarrow F$ from T_n that does not contain X, and replace all occurrences of c in it by X. The result is a derivation of $G(X) \rightarrow F$ from T_n, because c occurs neither in $G(X) \rightarrow F$ nor in T_n.) Since $T_n \vdash \exists X G(X)$, it follows that $T_n \vdash F$, which we assumed is not the case.

Case 2: n is odd. Let $G \vee H$ be the first sentence from D_n such that $T_n \vdash G \vee H$. (Such a sentence exists because D_0 contains infinitely many sentences with this property, and D_n is obtained from D_0 by removing finitely many sentences.) Define T_{n+1} to be $T_n \cup \{G\}$ if $T_n, G \not\vdash F$, and $T_n \cup \{H\}$ otherwise; $E_{n+1} = E_n$, and $D_{n+1} = D_n \setminus \{G \vee H\}$. Let us show that the property $T_n \not\vdash F$ is preserved. The assertion $T_{n+1} \not\vdash F$ is obvious if $T_n, G \not\vdash F$ and T_{n+1} is defined as $T_n \cup \{G\}$. Consider the case when $T_n, G \vdash F$ and T_{n+1} is defined as $T_n \cup \{H\}$. Assume that $T_{n+1} \vdash F$. Then $T_n, G \vee H \vdash F$. Since $T_n \vdash G \vee H$, it follows that $T_n \vdash F$, which we assumed is not the case.

Finally, we define T' to be $\cup_{n \geq 0} T_n$.

It is clear that condition (α) is satisfied. Condition (β) follows from the fact that $T_n \not\vdash F$ for all n. The verification of the remaining conditions uses two facts:

(a) for any sentence G from E_0 such that $T' \vdash G$ there exists n such that $G \notin E_n$;

(b) for any sentence G from D_0 such that $T' \vdash G$ there exists n such that $G \notin D_n$,

To verify condition (γ), we need to show that $T' \vdash G$ implies $G \in T'$. Assume that $T' \vdash G$. Then $T' \vdash G \vee G$ and, by (b), there exists n such that $G \vee G \notin D_n$. Take the smallest such n, so that $G \vee G \in D_{n-1}$. From the recursive definition of the sets D_n we see that $T_{n-1} \vdash G \vee G$. It follows that $G \in T_n$, and consequently $G \in T'$.

To prove (δ), assume that $G \vee H \in T'$. Then, by (b), there exists n such that $G \vee H \notin D_n$. Take the smallest such n, so that $G \vee H \in D_{n-1}$. From the recursive definition of the sets D_n and T_n we see that T_n is $T_{n-1} \cup \{G\}$ or $T_{n-1} \cup \{H\}$. Thus one of the formulas G, H belongs to T_n, and consequently to T'.

To prove (ϵ), assume that $\exists X G(X) \in T'$. Then, by (a), there exists n such that $\exists X G(X) \notin E_n$. Take the smallest such n, so that $\exists X G(X) \in E_{n-1}$. From the recursive definition of the sets E_n and T_n we see that T_n is $T_{n-1} \cup \{G(c)\}$ for some constant c from C_s, where $s =$sort(X). Thus $G(c)$ belongs to T_n, and consequently to T'. □

Now we are ready to define the counterexample $\langle \mathcal{H}, I \rangle$. Take a set T' of sentences over σ' satisfying conditions (α)–(ϵ) from Lemma 6. For any ground terms t_1 and t_2 over σ', we write $t_1 \approx t_2$ if the formula $t_1 = t_2$ belongs to T'. Then

(a) the domain $|I|^s$ is the set of all equivalence classes of $\approx$ that contain a term t such that sort$(t) \preceq$ sort(X);

(b) for each object constant c of σ, c^I is the equivalence class of $\approx$ that contains c;

(c) for each function constant f of positive arity, $f^I(d_1, d_2, \dots)$ is the equivalence class of $\approx$ that contains the term $f(t_1, t_2, \dots)$ for all terms $t_1 \in d_1$, $t_2 \in d_2$, $\dots$ over σ'.

To conclude the definition of I, we need to define p^I for predicate constants p. From $T' \not\vdash F$ we can conclude that $T' \not\vdash \bot$, and, by Lemma 5(ii), that $T' \not\vdash_c \bot$. Then, by Lindenbaum's Lemma (Mendelson 1987, Lemma 2.14 extended to the many-sorted case), there exists a complete, consistent extension T'' of T'. We define:

(d) for each predicate constant p, $p^I(d_1, d_2, \dots)$ is *true* if $p(t_1, t_2, \dots) \in T''$ for all terms $t_1 \in d_1$, $t_2 \in d_2$, $\dots$ over σ'.

Finally,

(e) $\mathcal{H}$ is the set of all formulas of the form $p(d_1^*, d_2^*, \dots)$ such that p is intensional and $p(t_1, t_2, \dots) \in T'$ for all terms $t_1 \in d_1$, $t_2 \in d_2$, $\dots$ over σ'.

The HT-interpretation $\langle \mathcal{H}, I \rangle$ of σ can be extended to the signature σ' by allowing c in clause (b) of the definition to be an arbitrary object constant from σ'.

We will show that for any sentence G over σ',

$$\langle \mathcal{H}, I \rangle \models_{ht} G \text{ iff } G \in T' \tag{18}$$

(Lemma 11 below). The desired properties of the HT-interpretation $\langle \mathcal{H}, I \rangle$—it satisfies all sentences in T but does not satisfy F—follow from this fact, because $T \subseteq T'$ and $F \notin T'$.

Lemma 7. *(i) For any sentence of the form* $t_1 = t_2$ *over* σ',

$$(t_1 = t_2) \in T' \textit{ iff } (t_1 = t_2) \in T''.$$

(ii) For any sentence of the form $p(\mathbf{t})$ *over* σ' *such that* p *is extensional,*

$$p(\mathbf{t}) \in T' \textit{ iff } p(\mathbf{t}) \in T''.$$

Proof. (i) The if part follows from the fact that $T' \subseteq T''$. Only if: Assume that $(t_1 = t_2) \notin T'$. From property (γ) we can conclude that T' contains the instance $t_1 = t_2 \vee t_1 \neq t_2$

of axiom (12). By property (δ), it follows that T' contains $t_1 \neq t_2$ as well. Since T'' is a consistent superset of T', we can conclude that $(t_1 = t_2) \notin T''$. The proof of part (ii) is similar, using (13) instead of (12). $\square$

Lemma 8. *For any sentence of the form $\exists X G(X)$ over σ' there exists an object constant c in $C_{\text{sort}(X)}$ such that the formula*

$$\exists X G(X) \rightarrow G(c) \tag{19}$$

belongs to T''.

Proof. *Case 1:* $\exists X G(X) \in T''$. By Lemma (14), the sentence

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

is provable in *SQHT*$^=$. Consequently it belongs to T'. By (δ), T' contains one of its disjunctive terms. But the first disjunctive term cannot belong to T' because the consistent superset T'' of T' contains $\exists X G(X)$. Consequently $\neg\neg\exists X G(X)$ belongs to T'. By Lemma 4(iii), it follows that $\exists X \neg\neg G(X)$ belongs to T' as well. By condition (ϵ), it follows that there exists an object constant c from $C_{\text{sort}(X)}$ such that $\neg\neg G(c)$ belongs to T'. It remains to observe that T'' is a superset of T' closed under $\vdash_c$, and that (19) is a classical consequence of $\neg\neg G(c)$. *Case 2:* $\exists X G(X) \notin T''$. Since T'' is complete, it contains $\neg\exists X G(X)$; (19) is a classical consequence of this formula. $\square$

Lemma 9. *For any ground term t, t^I is the equivalence class of t.*

Proof. By induction on t. $\square$

Lemma 10. *For any sentence G over σ', $I \models G$ iff $G \in T''$.*

Proof. By induction on the size of the formula G. We will consider the three cases where reasoning is different than in the similar proof for intuitionistic logic (van Dalen 1986, Section 3): $t_1 = t_2$, $G \rightarrow H$, and $\forall X G(X)$.

1. To check that $I \models t_1 = t_2$ iff $t_1 = t_2 \in T''$, we show that each side is equivalent to $t_1 \approx t_2$. For the left-hand side, this follows from Lemma 9. For the right-hand side, this follows from the definition of $\approx$ and Lemma 7(i).

2. We want to show that $I \models G \rightarrow H$ iff $G \rightarrow H \in T''$. By the induction hypothesis,

$$I \models G \text{ iff } G \in T''$$

and

$$I \models H \text{ iff } H \in T''.$$

Then, since T'' is complete and consistent,

$$\begin{aligned}(G \rightarrow H) \in T'' &\text{ iff } \neg G \in T'' \text{ or } H \in T'' \\ &\text{ iff } I \not\models G \text{ or } I \models H \\ &\text{ iff } I \models G \rightarrow H.\end{aligned}$$

3. We want to show that

$$I \models \forall X G(X) \text{ iff } \forall X G(X) \in T''.$$

For the if part, assume that $\forall X G(X) \in T''$ and take any element d of $|I|^{\text{sort}(X)}$. By the definition of $|I|^s$, there exists a ground term t such that sort$(t) \preceq$ sort(X) and $t \in d$. Since T'' is closed under $\vdash$, $G(t) \in T''$. By the induction hypothesis, it follows that $I \models G(t)$. By Lemma 9, $t^I = d$. By Lemma 3, it follows that $I \models G(d^*)$. Thus $I \models \forall X\, G(X)$. To prove the only if part, take an object constant c in $C_{\text{sort}(X)}$ such that the sentence

$$\exists X \neg G(X) \rightarrow \neg G(c) \tag{20}$$

belongs to T'' (Lemma 8). Assume that $I \models \forall X G(X)$. Then $I \models G(c)$. By the induction hypothesis, it follows that $G(c)$ belongs to T''. It remains to observe that $\forall X G(X)$ is a classical consequence of (20) and $G(c)$. $\square$

Lemma 11. *For any sentence G over σ', $\langle \mathcal{H}, I\rangle \models_{ht} G$ iff $G \in T'$.*

Proof. By induction on the size of the formula G. We will consider the same three cases as in the previous proof.

1. To check that $\langle \mathcal{H}, I\rangle \models_{ht} t_1 = t_2$ iff $t_1 = t_2 \in T'$, we show that each side is equivalent to $t_1 \approx t_2$. For the left-hand side, this follows from the fact that for every ground term t, t^I is the equivalence class of $\approx$ that contains t (Lemma 9). The right-hand side is immediate from the definition of $\approx$.

2. We want to show that

$$\langle \mathcal{H}, I\rangle \models_{ht} G \rightarrow H \text{ iff } G \rightarrow H \in T'.$$

For the if part, assume that $(G \rightarrow H) \in T'$. Since T' is closed under $\vdash$, it follows that $G \notin T'$ or $H \in T'$. By the induction hypothesis,

$$\langle \mathcal{H}, I\rangle \models_{ht} G \text{ iff } G \in T'$$

and

$$\langle \mathcal{H}, I\rangle \models_{ht} H \text{ iff } H \in T'.$$

Consequently $\langle \mathcal{H}, I\rangle \not\models_{ht} G$ or $\langle \mathcal{H}, I\rangle \models_{ht} H$. Furthermore, $(G \rightarrow H) \in T' \subseteq T''$, so that $I \models G \rightarrow H$ (Lemma 10). Thus $\langle \mathcal{H}, I\rangle \models_{ht} G \rightarrow H$. For the only if part, assume that $\langle \mathcal{H}, I\rangle \models_{ht} G \rightarrow H$. By the induction hypothesis, it follows that

$$G \notin T' \text{ or } H \in T'. \tag{21}$$

On the other hand, by Lemma 10, we can conclude that

$$G \notin T'' \text{ or } H \in T''. \tag{22}$$

Case 1: $G \in T'$. Then, by (21), $H \in T'$ and consequently $(G \rightarrow H) \in T'$. *Case 2:* $\neg G \in T'$. Then $(G \rightarrow H) \in T'$ because $\neg G \vdash G \rightarrow H$. *Case 3:* $G \notin T'$ and $\neg G \notin T'$. From (14) we can conclude that T' contains $\neg G \vee \neg\neg G$. By property (δ) of T', it follows that $\neg\neg G \in T' \subseteq T''$. Then $G \in T''$ and, by (22), $H \in T''$. Since T'' is consistent and contains T', it follows that $\neg H \notin T'$. Since T' contains the instance $G \vee (G \rightarrow H) \vee \neg H$ of axiom schema (10), contains neither G nor $\neg H$, and satisfies (δ), we conclude that $(G \rightarrow H) \in T'$ in this case as well.

3. We want to show that

$$\langle \mathcal{H}, I\rangle \models_{ht} \forall X G(X) \text{ iff } \forall X G(X) \in T'.$$

For the if part, the reasoning is the same as in the proof of Lemma 10. For the only if part, consider the instance

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

of axiom schema (11). By condition (ϵ), there exists an object constant c in $C_{\text{sort}(X)}$ such that the formula

$$G(c) \to \forall X G(X) \tag{23}$$

belongs to T'. Assume that $\langle \mathcal{H}, I\rangle \models_{ht} \forall X G(X)$. Then $\langle \mathcal{H}, I\rangle \models_{ht} G\left(\left(c^I\right)^*\right)$; by Lemma 3, $\langle \mathcal{H}, I\rangle \models_{ht} G(c)$. By the induction hypothesis, this implies that $G(c) \in T'$. It remains to observe that $\forall X G(X)$ is an intuitionistic consequence of $G(c)$ and (23). $\square$

6.4 Proof of Theorem 4: Soundness

The deductive system *SQHT*$^\omega$ is the result of adding inference rule (15) to the system *SQHT*$^=$. We will extend the argument outlined in Section 6.2 by discussing the case corresponding to the additional rule.

Take an instance

$$\frac{\Gamma(X, \mathbf{Y}) \Rightarrow F(t, \mathbf{Y}) \text{ for all terms } t \text{ in } \omega(\text{sort}(X))}{\Gamma(X, \mathbf{Y}) \Rightarrow \forall X\, F(X, \mathbf{Y})} \tag{24}$$

of rule (15), where $\mathbf{Y}$ is the list of its free variables other than X. Take an ω-interpretation $\langle \mathcal{H}, I\rangle$ such that

$$\langle \mathcal{H}, I\rangle \models_{ht} \forall X\mathbf{Y}\big(\Gamma^\wedge(X, \mathbf{Y}) \to F(t, \mathbf{Y})\big) \tag{25}$$

for all terms t in $\omega(\text{sort}(X))$; we need to show that $\langle \mathcal{H}, I\rangle$ satisfies

$$\forall X\mathbf{Y}\big(\Gamma^\wedge(X, \mathbf{Y}) \to \forall X\, F(X, \mathbf{Y})\big). \tag{26}$$

Note first that

$$\langle \mathcal{H}, I\rangle \models_{ht} \forall X\mathbf{Y}\big(\Gamma^\wedge(X, \mathbf{Y}) \to F(d^*, \mathbf{Y})\big) \tag{27}$$

for every d in $|I|^{\text{sort}(X)}$. Indeed, take a term t in $\omega(\text{sort}(X))$ such that $t^I = d$; then $d^* = (t^I)^*$, and (27) follows from (25) by Lemma 3. Hence $\langle \mathcal{H}, I\rangle$ satisfies

$$\forall Z X\mathbf{Y}\big(\Gamma^\wedge(X, \mathbf{Y}) \to F(Z, \mathbf{Y})\big), \tag{28}$$

where Z is a fresh variable of the same sort as X. The goal (26) can be derived from (28) in *SQHT*$^=$ as follows. From (28),

$$\exists X\, \Gamma^\wedge(X, \mathbf{Y}) \Rightarrow \forall Z\, F(Z, \mathbf{Y}).$$

Then, by $\forall$-elimination and $\forall$-introduction,

$$\exists X\, \Gamma^\wedge(X, \mathbf{Y}) \Rightarrow \forall X\, F(X, \mathbf{Y}).$$

Using the sequent

$$\Gamma^\wedge(X, \mathbf{Y}) \Rightarrow \exists X\, F(X, \mathbf{Y})$$

and $\exists$-elimination, we further conclude

$$\Gamma^\wedge(X, \mathbf{Y}) \Rightarrow \forall X F(X, \mathbf{Y}),$$

and (26) follows by $\to$-introduction and $\forall$-introduction.

6.5 Omitting Types

The completeness part of the main theorem is derived in Section 6.6 from the omitting types theorem for the logic of here-and-there, stated below. In its statement,

- T is a theory over σ, and F is a sentence over σ such that $T \not\vdash F$;
- S is a subset of the set of sorts of σ,
- for every sort s in S, X^s is a variable of sort s, and Σ^s is a subset of the set of formulas that have no free variables other than X^s.

Omitting Types Theorem. *If for every sentence of the form $\exists X^s\, G(X^s)$ such that*

$$T,\, \exists X^s G(X^s) \not\vdash F$$

there exists a formula $H(X^s)$ in Σ^s such that

$$T, \exists X^s(G(X^s) \wedge H(X^s)) \not\vdash F$$

then T has an HT-model $\langle \mathcal{H}, I\rangle$ satisfying the following conditions:

(i) $\langle \mathcal{H}, I\rangle \not\models_{ht} F$;

(ii) for every s in S and every d in $|I|^s$ there exists a formula $H(X^s)$ in Σ^s such that $\langle \mathcal{H}, I\rangle \models_{ht} H(d^*)$.

In the following lemma, as in Section 6.3, σ' is the signature obtained from σ by adding, for every sort s, a countable set C_s of object constants of that sort.

Lemma 12. *If for every sentence of the form $\exists X^s G(X^s)$ such that*

$$T,\, \exists X^s G(X^s) \not\vdash F$$

there exists a formula $H(X^s)$ in Σ^s such that

$$T, \exists X^s(G(X^s) \wedge H(X^s)) \not\vdash F$$

then there exists a theory T' over σ' satisfying conditions (α)–(ϵ) from Lemma 6 and the condition

(ζ) for every sort s in S and every ground term t of sort s there exists a formula $H(X^s)$ in Σ^s such that $H(t) \in T'$.

Proof. Choose an enumeration of the union C of the sets C_s for all s in S. We define sets T_n, E_n, D_n recursively, as in the proof of Lemma 6, except that we distinguish between three cases, instead of two.

Case 1: $n = 3k - 2$. The sets T_{n+1}, E_{n+1}, D_{n+1} are defined as in Case 1 of the proof of Lemma 6.

Case 2: $n = 3k - 1$. The sets T_{n+1}, E_{n+1}, D_{n+1} are defined as in Case 2 of the proof of Lemma 6.

Case 3: $n = 3k$. Let c be the k-th constant in C, and let $\mathbf{c}$ be the list of all other constants from C that occur in T_n. (There are finitely many such constants, because T_n is the result of adding n formulas to T.) Then T_n can be represented as $T \cup \{G_1(c, \mathbf{c}), \dots, G_n(c, \mathbf{c})\}$ for some formulas $G_i(X^s, \mathbf{Y})$ over σ, where $s = \text{sort}(c)$. Let $G(X^s)$ be the formula $\exists \mathbf{Y}(G_1(X^s, \mathbf{Y}) \wedge \cdots \wedge G_n(X^s, \mathbf{Y}))$. The assumption that $T,\, \exists X^s G(X^s) \vdash F$ leads to a contradiction, because

$$T \subseteq T_n,\ T_n \vdash \exists X^s G(X^s), \text{ and } T_n \not\vdash F.$$

Thus T, $\exists X^s G(X^s) \not\vdash F$. Consequently there exists a formula $H(X^s)$ in Σ^s such that

$$T, \exists X^s(G(X^s) \wedge H(X^s)) \not\vdash F. \qquad (29)$$

Define

$$T_{n+1} = T_n \cup \{H(c)\}, \quad E_{n+1} = E_n, \quad D_{n+1} = D_n.$$

To show that the property $T_n \not\vdash F$ is preserved, assume that $T_{n+1} \vdash F$. Then

$$T, G_1(c, \mathbf{c}) \wedge \cdots \wedge G_n(c, \mathbf{c}), H(c) \vdash F.$$

Since the constants $\mathbf{c}$ occur neither in T nor $H(c)$ nor in F, it follows that

$$T, \exists \mathbf{Y}(G_1(c, \mathbf{Y}) \wedge \cdots \wedge G_n(c, \mathbf{Y})), H(c) \vdash F,$$

which can be written as T, $G(c)$, $H(c) \vdash F$. Since the constant c occurs neither in T nor in F, it follows that

$$T, \exists X^s(G(X^s) \wedge H(X^s)) \vdash F,$$

which contradicts (29).

Define T' as $\cup_{n \geq 0} T_n$. Then properties (α)–(ϵ) are proved in the same way as in the proof of Lemma 6. To prove property (ζ), take a term t of sort s and consider the formula $\exists X^s(X^s = t)$. It is provable in *SQHT*$^=$ and consequently belongs to T'. By property (ϵ), it follows that C_s contains a constant c such that $c = t$ belongs to T'. Take k such that c is the k-th constant in the set C. Then $H(c) \in T_{3k+1} \subseteq T'$, and consequently $H(t) \in T'$. ☐

To prove the Omitting Types Theorem, we define $\langle \mathcal{H}, I \rangle$ as in Section 6.3. Property (i) is established by the same reasoning as in the completeness proof above. To prove property (ii), take a sort s in S, an element d of $|I|^s$, and a term t in d. By Lemma 12, there exists a formula $H(X^s)$ in Σ^s such that $H(t) \in T'$. By Lemma 11, it follows that $\langle \mathcal{H}, I \rangle \models_{ht} H(t)$. By Lemma 9, $t^I = d = (d^*)^I$. By Lemma 3, it follows that $\langle \mathcal{H}, I \rangle \models_{ht} H(d^*)$.

6.6 Proof of Theorem 4: Completeness

Let F be a sentence that is not derivable in *SQHT*$^\omega$ from the axioms of a theory T. Our goal is to construct an ω-model of T that does not satisfy F.

Consider the set T' of sentences over σ that can be derived from the axioms of T in *SQHT*$^\omega$. We will apply Omitting Types Theorem (Section 6.5) to the theory T', with the set $\{X^s = t : t \in \omega(s)\}$ as Σ^s for all $s \in S$. To use the theorem, we need to show that for every sentence of the form $\exists X^s\, G(X^s)$ such that

$$T', \exists X^s\, G(X^s) \not\vdash F \qquad (30)$$

there exists a term t in $\omega(s)$ such that

$$T', \exists X^s(G(X^s) \wedge X^s = t) \not\vdash F.$$

Assume that this not the case, so that for all t in $\omega(s)$

$$T', \exists X^s(G(X^s) \wedge X^s = t) \vdash F.$$

Then

$$T', G(t) \vdash F \qquad (t \in \omega(s))$$

and consequently

$$T' \vdash G(t) \to F \qquad (t \in \omega(s)),$$
$$T' \vdash_\omega \forall X^s(G(X^s) \to F),$$

and

$$\forall X^s(G(X^s) \to F) \in T',$$

because T' is closed under $\vdash_\omega$. This conclusion contradicts (30).

By the Omitting Types Theorem, T' has an HT-model $\langle \mathcal{H}, I \rangle$ such that

(i) $\langle \mathcal{H}, I \rangle \not\models_{ht} F$;

(ii) for every s in S and every d in $|I|^s$ there exists a term t in $\omega(s)$ satisfying the condition $\langle \mathcal{H}, I \rangle \models_{ht} d^* = t$.

The last condition is equivalent to $d = t^I$. Consequently (ii) asserts that I is an ω-interpretation.

Conclusion

The main result of this paper is an ω-completeness theorem for the many-sorted logic of here-and-there. It is derived from a types omission theorem for that logic. Using the main theorem, we showed that the strong equivalence relation on mini-GRINGO programs can be characterized as the possibility of deriving rules, rewritten as first-order formulas, in the deductive system *SQHT*$^\omega$. Extending the last result to more expressive languages of answer set programming is a topic for future work.

Acknowledgements

Many thanks to Michael Gelfond and to the anonymous referees for helping us improve previous versions of this paper.

References

Bagheri, S.-M., and Pourmahdian, M. 2011. Omitting types in an intermediate logic. *Studia Logica: An International Journal for Symbolic Logic* 97:319–328.

Fandinno, J.; Lifschitz, V.; Lühne, P.; and Schaub, T. 2020. Verifying tight logic programs with Anthem and Vampire. *Theory and Practice of Logic Programming* 20.

Ferraris, P.; Lee, J.; and Lifschitz, V. 2011. Stable models and circumscription. *Artificial Intelligence* 175:236–263.

Ferraris, P. 2005. Answer sets for propositional theories. In *Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, 119–131.

Gebser, M.; Harrison, A.; Kaminski, R.; Lifschitz, V.; and Schaub, T. 2015. Abstract Gringo. *Theory and Practice of Logic Programming* 15:449–463.

Gebser, M.; Kaminski, R.; Kaufmann, B.; Lindauer, M.; Ostrowski, M.; Romero, J.; Schaub, T.; and Thiele, S. 2019. Potassco User Guide. https://github.com/potassco/guide/releases/.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In Kowalski, R., and Bowen, K., eds., *Proceedings of International Logic Programming Conference and Symposium*, 1070–1080. MIT Press.

Harrison, A.; Lifschitz, V.; Pearce, D.; and Valverde, A. 2017. Infinitary equilibrium logic and strongly equivalent logic programs. *Artificial Intelligence* 246:22–33.

Henkin, L. 1954. A generalization of the concept of ω-consistency. *The Journal of Symbolic Logic* 19:183–196.

Hosoi, T. 1966. The axiomatization of the intermediate propositional systems S_n of Gödel. *Journal of the Faculty of Science of the University of Tokyo* 13:183–187.

Kiesler, H. J. 1977. Fundamentals of model theory. In Barwise, J., ed., *Handbook of Mathematical Logic*. Amsterdam: North-Holland. 47–105.

Lifschitz, V.; Lühne, P.; and Schaub, T. 2019. Verifying strong equivalence of programs in the input language of gringo. In *Proceedings of the 15th International Conference on Logic Programming and Non-monotonic Reasoning*.

Lifschitz, V.; Morgenstern, L.; and Plaisted, D. 2008. Knowledge representation and classical logic. In van Harmelen, F.; Lifschitz, V.; and Porter, B., eds., *Handbook of Knowledge Representation*. Elsevier. 3–88.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic* 2:526–541.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2007. A characterization of strong equivalence for logic programs with variables. In *Procedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, 188–200.

Lifschitz, V. 2021. Here and there with arithmetic. *Theory and Practice of Logic Programming*.

Marković, Z. 1979. An intuitionistic omitting types theorem. *Publications de l'Institut Mathématiques, Nouvelle série* 25(40):167–169.

Marković, Z. 1995. Omitting types in Kripke models. *Filomat* 9:803–807.

Mendelson, E. 1987. *Introduction to Mathematical Logic*. Wadsworth & Brooks. Third edition.

Mints, G. 2000. *A Short Introduction to Intuitionistic Logic*. Kluwer.

Pearce, D., and Valverde, A. 2004. Towards a first order equilibrium logic for nonmonotonic reasoning. In *Proceedings of European Conference on Logics in Artificial Intelligence (JELIA)*, 147–160.

Truszczynski, M. 2012. Connecting first-order ASP and the logic FO(ID) through reducts. In Erdem, E.; Lee, J.; Lierler, Y.; and Pearce, D., eds., *Correct Reasoning: Essays on Logic-Based AI in Honor of Vladimir Lifschitz*. Springer. 543–559.

van Dalen, D. 1986. Intuitionistic logic. In Gabbay, D., and Guenther, F., eds., *Handbook of Philosophical Logic, Volume III: Alternatives in Classical Logic*. Dordrecht: D. Reidel Publishing Co.