# Isoinitial Models for Logic Programs: A Preliminary Study

Kung-Kiu Lau, Mario Ornaghi

**Abstract**

The Herbrand model $H$ of a definite logic program $P$ is an initial model among the class of all the models of $P$, interpreting $P$ as an initial theory. Such a theory proves (computes) only positive literals (atoms) in $P$, so it does not deal with negation. In this paper, we introduce *isoinitial* models of logic programs. We show that isoinitial semantics deals with negation, and works in a uniform way for definite and normal logic programs. Moreover, the lack of an isoinitial model signals the absence of information. Thus it also provides a unifying semantics for closed and open logic programs.

*Keywords: Semantics, isoinitial models, negation*

## 1 Introduction

The intended model of a standard (Horn clause) logic program $P$ is its Herbrand model $H$. It interprets $P$ under the Closed World Assumption [11]. Considering the class of all the models of $P$, $H$ interprets $P$ as an initial (Horn) theory [5]. A distinguishing feature of an initial theory $P$ is that, in general, it proves (computes) only positive literals (atoms) in $P$, so it does not deal with negation.

In this paper, we introduce *isoinitial* models [2] of logic programs. If the completion $Comp(P)$ of a program $P$ has an isoinitial model, then $H$ is also such a model of $Comp(P)$. For a definite program $P$, $H$ is always an initial model of $Comp(P)$, but $Comp(P)$ may have no isoinitial models. Far from being a drawback, this is in fact an advantage: the lack of an isoinitial model is a symptom of termination problems with respect to finite failure. That is, isoinitial semantics works in a uniform way for definite and normal logic programs; the lack of an isoinitial model always exposes some circularity or lack of information in $Comp(P)$, with respect to the provability of negated atoms.

This capability of signaling absence of information makes isoinitial semantics a good candidate for capturing *open*, i.e. incomplete, programs. Indeed isoinitial semantics provides a unifying semantics for closed and open logic programs. It also deals with negation. We have used isoinitial semantics in our work in formal program development in computational logic (e.g. [7, 8]), since it provides a suitable semantic framework.

In this paper we consider mainly definite programs, and we compare initial and isoinitial semantics for such programs. Normal programs, for which other kinds of semantics should be considered, are only briefly addressed at the end of Section 3.

First Author is with Dept. of Computer Science, University of Manchester, Manchester M13 9PL, UK. E-mail: kung-kiu@cs.man.ac.uk

Second Author is with Dip. di Scienze dell'Informazione, Universita' di Milano, Via Comelico 39, 20135 Milano, Italy. E-mail: ornaghi@dsi.unimi.it

# 2 Initial and Isoinitial Theories

A $\Sigma$-theory is a set of $\Sigma$-sentences, where $\Sigma \equiv \langle F, R\rangle$ is a signature $\Sigma$ with function symbols $F$ and relation symbols $R$ over the domain $D$, where each symbol has an associated arity. For example, Peano Arithmetic is a $Nat$-theory, where $Nat \equiv \langle\{0^0, s^1, +^2, *^2\}, \{=^2\}\rangle$.[1]

We shall work in first-order logic with identity, i.e. identity $=$ and the usual identity axioms will be always understood. For example, we can introduce $Nat$ as the signature $Nat \equiv \langle\{0^0, s^1, +^2, *^2\}, \{\}\rangle$, $=$ being understood.

Let $\Sigma \equiv \langle F, R\rangle$ be a signature. As usual, a $\Sigma$-*structure* is a triple $\mathcal{M} \equiv \langle D, F^{\mathcal{M}}, R^{\mathcal{M}}\rangle$, where $F^{\mathcal{M}}$ is a $F$-indexed set of functions interpreting $F$, and $R^{\mathcal{M}}$ is an $R$-indexed set of relations interpreting $R$. Of course, the interpretation of a function symbol $f^n$ is an $n$-ary function $f^{n\mathcal{M}} : D^n \rightarrow D$, and the interpretation of a relation symbol $r^m \in R$ is an $m$-ary relation $r^{m\mathcal{M}} \subseteq D^m$. When no confusion can arise, we may omit the arity, i.e. we write $f^{\mathcal{M}}$ instead of $f^{n\mathcal{M}}$ and $r^{\mathcal{M}}$ instead of $r^{m\mathcal{M}}$.

In a structure $\mathcal{M}$, terms and formulas are interpreted in the usual way. $t^{\mathcal{M}}$ will indicate the value of a ground term $t$ in $\mathcal{M}$, and $\mathcal{I} \models A$ will indicate that the sentence (sentences) $A$ is (are) true in $\mathcal{I}$.

Finally, *homomorphisms*, *isomorphisms* and *isomorphic embeddings* are defined in the usual way. Since the latter are less popularly used in the literature than homomorphisms and isomorphisms, we briefly recall them here.

An *isomorphic embedding* $i : \mathcal{J} \rightarrow \mathcal{M}$ is a homomorphism that preserves the complements of relations, i.e.: $(\alpha_1, \ldots, \alpha_n) \notin r^{\mathcal{J}}$ entails $(i(\alpha_1), \ldots, i(\alpha_n)) \notin r^{\mathcal{M}}$. Therefore, $\alpha \neq \beta$ entails $i(\alpha) \neq i(\beta)$, i.e. isomorphic embeddings are injective. Moreover, the $i$-image of $\mathcal{J}$ is isomorphic to a substructure of $\mathcal{M}$, i.e. $\mathcal{J}$ is 'isomorphically embedded' in $\mathcal{M}$.

Now we can define initial and isoinitial models of $\Sigma$-theories.

**Definition 2.1** (Initial Models) Let $T$ be a first-order $\Sigma$-theory, and $\mathcal{I}$ be a model of $T$. $\mathcal{I}$ is an *initial model* of $T$ iff, for every other model $\mathcal{M}$ of $T$, there is a unique homomorphism $h : \mathcal{I} \rightarrow \mathcal{M}$.

**Definition 2.2** (Isoinitial Models) Let $T$ be a first-order $\Sigma$-theory, and $\mathcal{J}$ be a model of $T$. $\mathcal{J}$ is an *isoinitial model* of $T$ iff, for every other model $\mathcal{M}$ of $T$, there is a unique isomorphic embedding $i : \mathcal{I} \rightarrow \mathcal{M}$.

**Example 2.1** Consider the simple signature $K \equiv \langle\{a^0, b^0\}, \{\}\rangle$, containing just two constant symbols $a$ and $b$. The corresponding Herbrand Interpretation $H$ is defined by $D = \{a, b\}$, $a^H = a$ and $b^H = b$.[2]

$H$ is an initial model of the empty theory $\emptyset$. Indeed, for every other model $\mathcal{M}$, the map $h$ defined by $(h(a) = a^{\mathcal{M}}, h(b) = b^{\mathcal{M}})$ is the unique homomorphism from $H$ into $\mathcal{M}$. The empty theory does not prevent interpretations where $a = b$.

$H$ is not an isoinitial model of $\emptyset$ however. Indeed, there is no isomorphic embedding from $H$ into models $\mathcal{M}$ such that $a^{\mathcal{M}} = b^{\mathcal{M}}$, since isomorphic embeddings have to preserve inequality.

$H$ is an isoinitial model of $\neg a = b$. Indeed, for every model $\mathcal{M}$ of $\neg a = b$, we have $a^{\mathcal{M}} \neq b^{\mathcal{M}}$, and the map $i$ such that $(i(a) = a^{\mathcal{M}}, i(b) = b^{\mathcal{M}})$ is the unique isomorphic embedding of $H$ into $\mathcal{M}$.

---

[1] $\sigma^n$ indicates a symbol $\sigma$ with arity $n$.

[2] The standard interpretation of $=$ is understood.

It is worthwhile noting that $H$ is both an initial and an isoinitial model of $\neg a = b$.

We will consider the particular case of *reachable* initial and isoinitial models: a structure $\mathcal{M} \equiv \langle D, F^{\mathcal{M}}, R^{\mathcal{M}} \rangle$ is reachable if, for every $\alpha \in D$, there is a ground term $t$ such that $t^{\mathcal{M}} = \alpha$.

**Theorem 2.1** Let $\mathcal{J}$ be a reachable model of a $\Sigma$-theory $T$. Then $\mathcal{J}$ is an initial model of $T$ if and only if the following *initiality condition* (INI) holds:

$$\text{Let } A \text{ be a ground atom and } \mathcal{M} \text{ be a model of } T. \text{ Then } \mathcal{J} \models A \text{ entails } \mathcal{M} \models A. \qquad (\textsc{ini})$$

while it is an isoinitial model of $T$ if an only if the following *isoinitiality condition* (ISO) holds:

$$\text{Let } A \text{ be a ground atom and } \mathcal{M} \text{ be a model of } T. \text{ Then } \mathcal{J} \models A \text{ iff } \mathcal{M} \models A. \qquad (\textsc{iso})$$

*Proof.* The non-trivial direction follows from the reachability hypothesis. Here we omit the proof for conciseness. □

That is, initial models represent truth of atomic formulas in every model, while isoinitial models represent both truth and falsity of atomic formulas in every model. By the completeness theorem for first-order theories, we can prove:

**Corollary 2.1** In Theorem 2.1, we can replace (INI) and (ISO) by:

$$\text{For every ground atom } A,\ \mathcal{J} \models A \text{ iff } T \vdash A. \qquad (\textsc{ini'})$$

$$\text{For every ground literal } L,\ \mathcal{J} \models L \text{ iff } T \vdash L. \qquad (\textsc{iso'})$$

That is, initial models represent provability of atomic formulas, while isoinitial models represent provability of literals, i.e. they behave properly with respect to negation of atomic formulas.

Of course, in general, a first-order theory may have no initial or isoinitial models. To state the existence of such models, we can apply the previous results. For isoinitial models, we have a more manageable condition, that allows us to introduce a proof-theoretical tool for stating isoinitiality.

**Corollary 2.2** In Theorem 2.1, we can replace (ISO) by the following *atomic completeness* condition:

$$\text{For every ground atom } A,\ T \vdash A \text{ or } T \vdash \neg A. \qquad (\textsc{atc})$$

In condition (ATC) models disappear altogether,[3] i.e., we have a purely proof-theoretic condition. As a first application, let us prove the following theorem:

**Theorem 2.2** Let $K \equiv \langle F, \{\} \rangle$ be a signature containing a non-empty set $F$ of function and constant symbols, with at least one constant. Let $H$ be the corresponding Herbrand structure. Let $CET(F)$ be Clark's Equality Theory for $F$. Then $H$ is an isoinitial model of $CET(F)$.

[3]However, models do not disappear completely from Theorem 2.1, because the existence of at least one reachable model $\mathcal{J}$ is always assumed.

*Proof.* $H$ is a model of $CET(F)$. Being a term-model, it is trivially reachable. Atomic completeness follows from the fact that, for every ground atomic formula $t = t'$, $\emptyset \vdash t = t'$ if $t$ and $t'$ coincide, and $CET(F) \vdash \neg t = t'$, if they are different. □

As a second application, we introduce constructive logics, as tools for formally proving isoinitiality.

**Definition 2.3** (Constructive Logics) Let $S$ be a classically consistent logic contained in $CL$. $S$ is *constructive* with respect to a theory $T$ if and only if:

- For every ground formula $A \vee B$, $T \vdash_S A \vee B$ entails $T \vdash_S A$ or $T \vdash_S B$.
- For every ground formula $\exists x \,.\, F(x)$, if $T \vdash_S \exists x \,.\, F(x)$, then there is a ground term $t$ such that $T \vdash_S F(t)$.

We recall that $S$ is classically consistent if $T \vdash \mathit{false}$ entails $T \vdash_S \mathit{false}$. Intuitionistic logic $\mathit{int}$ is not classically consistent, while logics containing $\mathit{int} + K$ are classically consistent, where $K$ is the Kuroda Principle $\forall x \neg\neg A(x) \to \neg\neg \forall x A(x)$. We can prove the following corollary:

**Corollary 2.3** Let $S$ be constructive with respect to $T$. Then, in Theorem 2.1, we can replace (ISO) by the following *constructive atomic completeness* condition:

$$T \vdash_S \forall x_1, \ldots, x_n \,.\, r(x_1, \ldots, x_n) \vee \neg r(x_1, \ldots, x_n), \text{ for every } r^n \in R. \qquad (\text{CATC})$$

*Proof.* Since $S$ is constructive, (CATC) entails atomic completeness. □

An interesting example is the logic $Kat = \mathit{int} + K + \mathit{at}$, where $\mathit{at}$ is $\neg\neg A \to A$, for $A$ atomic.[4] $Kat$ is constructive in any axiomatisation $T$ containing Harrop-axioms and possible first-order induction schemas. Moreover, in $(T + Kat)$-proofs we can use classical proofs of Harrop formulas, without enlarging the class of $(T + Kat)$-theorems.

We do not enter into further details here, because this paper is mainly on semantics of isoinitial models, and we use some proof-theoretical tools only in some examples. For a discussion, see [10, 9], where semi-constructive systems are also introduced.

# 3 Closed Logic Programs

We first introduce closed first-order theories. For the sake of conciseness, we will use *ini* for *initial semantics*, i.e. semantics based on initial models, *iso* for *isoinitial semantics*, i.e. semantics based on isoinitial models, and *sem* as a parameter standing for one of the two. Moreover, *ini*-models will be reachable initial models, and *iso*-models will be reachable isoinitial models.

Closed first-order $\Sigma$-theories are defined as follows:

**Definition 3.1** (*sem*-closed $\Sigma$-theories) A theory $T$ is *sem-closed* if and only if it has a *sem*-model.

[4] $Kat$ is non-standard. Indeed, in $\mathit{at}$, $A$ must be atomic and hence $\mathit{at}$ is not closed under substitution.

Let us consider a signature $K \equiv \langle F, \{\}\rangle$ with at least a constant symbol. We are interested in axiomatising the corresponding Herbrand Structure $H$.

The empty theory $\emptyset$ is *ini*-closed, and $H$ is an *ini*-model of $\emptyset$. By contrast, $\emptyset$ is not *iso*-closed. However, $CET(F)$ is an *iso*-closed theory, with *iso*-model $H$ (see Theorem 2.2).

Therefore, *iso* is better equipped to deal with negation: *iso* shows that $\emptyset$ lacks information with respect to negation, whereas *ini* does not show this. This advantage is further elucidated as follows:

**Definition 3.2** Let $T$ be a *sem*-closed theory with a *sem*-model $\mathcal{I}$. We say that $T$ is *closed* with respect to ($\mathcal{I}$ and) a sentence $F$ if and only if:

$$\mathcal{I} \models F \;\; \textit{iff} \;\; T \vdash F$$

$\exists(Q)$ is an existential formula if $\exists$ is a possibly empty list of existential quantifiers and $Q$ is quantifier-free. If $Q$ is positive, then $\exists(Q)$ is a positive existential formula. We have:

**Theorem 3.1** If $T$ is *ini*-closed, then it is closed with respect to all the positive existential sentences. Moreover, if $T \vdash \exists x \,.\, P(x)$, where $\exists x \,.\, P(x)$ is a positive existential sentence, then $T \vdash P(t)$, for at least a ground $t$.

If $T$ is *iso*-closed, then it is closed with respect to all the existential sentences. Moreover, if $T \vdash \exists x \,.\, Q(x)$, where $\exists x \,.\, Q(x)$ is an existential sentence, then $T \vdash Q(t)$, for at least a ground $t$.

*Proof.* By induction on the structure of formulas, using the reachability hypothesis and Corollary 2.1. □

That is, for *iso*-closed theories, $Q$ may contain negation, whereas $P$ cannot contain negation for *ini*-closed theories, unless the latter are also *iso*-closed.

Now we consider closed logic programs as closed first-order theories.

Let $P$ be a logic program, with signature $\Sigma_P \equiv \langle F_P, R_P\rangle$. We will separately consider the axioms of $CET(F_P)$, the universal closure $\forall P$ of the clauses of $P$, and, for every predicate $p \in R_P$, the only-if parts of the completed definition $Cdef^-(p)$ of $p$ in $P$. For the latter, we will use the counterposition $\forall x \,.\, \neg p(x) \leftarrow \neg D(x)$, instead of the usual $\forall x \,.\, p(x) \rightarrow D(x)$, because the former is an Harrop formula and we can use it in constructive reasoning. By $Cdef^-(P)$, we will indicate the set of the $Cdef^-(p)$, for $p \in R_P$. The completion of $P$, written $Comp(P)$, is the union of $CET(F_P)$, $P$ and $Cdef^-(P)$.

**Example 3.1** Let us consider the usual program $SumP$ for the sum of natural numbers, with signature $\Sigma_{SumP} \equiv \langle \{0^0, s^1\}, \{sum^3\}\rangle$. Now $CET(0^0, s^1)$ contains the axioms

$$\{\forall x \,.\, \neg 0 = s(x),\;\; \forall x, y \,.\, s(x) = s(y) \rightarrow x = y\} \cup \{\forall x \,.\, \neg s^n(x) = x | n > 0\}^5$$

$\forall SumP$ is:

$$\forall x \,.\, sum(x, 0, x),\;\; \forall x, i, v \,.\, sum(x, s(i), s(v)) \leftarrow sum(x, i, v)$$

$Cdef^-(sum)$ is (after some obvious simplifications):

$$\forall x, y, z \,.\, \neg sum(x, y, z) \leftarrow \neg(y = 0 \wedge z = x \vee (\exists i, v \,.\, y = s(i) \wedge z = s(v) \wedge sum(x, i, v)))$$

The completion is $Comp(SumP) = CET(0^0, s^1) \cup \forall SumP \cup Cdef^-(sum)$.

[5] $n$ is not the arity: it indicates the iteration of $s$ for $n$ times.

$H_P$ will indicate the Herbrand Structure corresponding to $F_P$. As usual, a Herbrand Model is an interpretation of the predicates of $R_P$ over the domain of $H_P$. The Minimum Herbrand Model $\mathcal{M}(P)$ is defined in the usual way.

**Theorem 3.2** $\mathcal{M}(P)$ is an initial model of $P$ and of $Comp(P)$, but it is not an isoinitial model of $P$.

*Proof.* The initiality of $\mathcal{M}(P)$ is well-known [5]. $\mathcal{M}(P)$ cannot be an isoinitial model of $P$, because no negated formula is provable from $P$ (that is, $P$ is not and cannot be atomically complete). □

One would expect that $\mathcal{M}(P)$ is an isoinitial model of $Comp(P)$. However, this is not necessarily true, as shown by the following example:

**Example 3.2** Consider the program $P_1$ :

$$\begin{array}{lcl} p(a) & \leftarrow & q(a) \\ q(a) & \leftarrow & p(a) \end{array} \qquad (P_1)$$

with signature $\Sigma_1 \equiv \langle \{a^0\}, \{p^1, q^1\} \rangle$. $CET(a)$ is empty. $Cdef^-(p)$ is $\forall x \,.\, \neg p(x) \leftarrow \neg(x = a \wedge q(a))$, and $Cdef^-(q)$ is $\forall x \,.\, \neg q(x) \leftarrow \neg(x = a \wedge p(a))$.

Atomic completeness requires that $Comp(P_1) \vdash p(a)$ or $Comp(P_1) \vdash \neg p(a)$, and $Comp(P_1) \vdash q(a)$ or $Comp(P_1) \vdash \neg q(a)$. However these requirements are not met and, therefore, no reachable isoinitial model can exist. On the other hand, the minimum Herbrand Model of $P_1$ (where $p(a)$ and $q(a)$ are false) is an initial model of $Comp(P_1)$. Therefore $P_1$ and $Comp(P_1)$ are *ini*-closed, but not *iso*-closed.

Let us consider the program $P_2$

$$p(a) \leftarrow q(a) \qquad (P_2)$$

$Comp(P_2)$ is both *ini*- and *iso*-closed. Indeed, now $Cdef^-(q)$ is $\forall x \,.\, \neg q(x)$, and we can prove $\neg q(a)$ and $\neg p(a)$. The Minimum Herbrand Model of $P_2$ is the same as that of $P_1$, but now it is both initial and isoinitial.

Finally, while $P_1$ does not terminate with respect to the goals $\leftarrow p(a)$ and $\leftarrow q(a)$, $P_2$ finitely fails for both.

This example suggests that termination and *iso*-closure are related. Indeed, we can prove the following result:

**Definition 3.3** (Existential Ground-termination) Let $P$ be a definite program. $P$ *existentially ground-terminates* if and only if its Herbrand Universe is not empty and, for every ground goal $\leftarrow A$, either there is a refutation of $\leftarrow A$, or $\leftarrow A$ finitely fails.

**Theorem 3.3** Let $P$ be a definite program with a non-empty Herbrand universe. $Comp(P)$ is *iso*-closed if and only if it existentially ground-terminates.

*Proof.* If $Comp(P)$ is *iso*-closed, it is atomically complete. By completeness of $SLDNF$-resolution for definite programs, $P$ existentially ground-terminates. If $P$ atomically ground-terminates, then $Comp(P)$ is atomically complete. □

This theorem indicates that we can use termination analysis for stating isoinitiality. Moreover, we can also do the converse, i.e. we can derive existential ground-termination by stating isoinitiality.

**Example 3.3** Consider the program $SumP$ in Example 3.1. We can prove that $SumP$ existentially ground-terminates. Therefore, we can conclude that its minimum Herbrand Model is an isoinitial model of $Comp(SumP)$.

However, we can also proceed in a different way. Since the axioms of $Comp(SumP)$ are Harrop formulas, $Kat$ is semi-constructive with respect to it. By induction on $n$, we can prove that, for every $s^n(O)$:

$$Comp(SumP) \vdash sum(x, s^n(O), z_1) \wedge sum(x, s^n(O), z_2) \rightarrow z_1 = z_2 \tag{1}$$

$$Comp(SumP) \vdash_{Kat} \exists z \,.\, sum(x, s^n(O), z) \tag{2}$$

From (1) and (2) we get

$$Comp(SumP) \vdash_{Kat} sum(x, s^n(O), z) \vee \neg sum(x, s^n(O), z) \tag{3}$$

In (1), classical logic $CL$ is used. As we already said, this is permitted because the proved formula is a Harrop one.

By constructiveness, we get that $Comp(SumP)$ is atomically complete. Therefore, its (trivially reachable) minimum Herbrand model is an isoinitial one.

As a corollary, we also get that $SumP$ existentially ground-terminates.

By Theorem 3.2, for a definite program $P$, $P$ and $Comp(P)$ are always *ini*-closed, but they may not be *iso*-closed, as we have shown. Local variables are a remarkable source of possible absence of *iso*-closure, as is shown by the following example:

**Example 3.4** Let us assume that there is a program $P_r$ for computing a relation $r(x, y)$ over some domain. Let $P_q$ be the following program:

$$\forall x \,.\, q(x) \leftarrow \exists y \,.\, r(x, y)$$

The completed definition $Cdef^-(q)$ is

$$\forall x \,.\, \neg q(x) \leftarrow \neg(\exists y \,.\, r(x, y))$$

Let us assume that $Comp(P_r)$ is *iso*-closed. This does not guarantee that $Comp(P_r \cup P_q)$, i.e $Comp(P_r) \cup Cdef^-(q)$, is *iso*-closed. However, it is *ini*-closed, because *ini*-closure is always guaranteed, for definite programs. That is, initial semantics does not have the capability to expose a possible lack of information, with respect to the decidability of $q$.

As we can see, $Comp(P_r \cup P_q)$ is *iso*-closed if and only if $Comp(P_r)$ is complete with respect to all the formulas $\neg(\exists y \,.\, r(t, y))$ such that $t$ is a ground term. For example, if the program $P_r$ is

$$r(a, b) \leftarrow$$

then $Comp(P_r) \vdash \exists y \,.\, r(a, y))$ and $Comp(P_r) \vdash \neg(\exists y \,.\, r(b, y))$, i.e. $Comp(P_r \cup P_q) \vdash q(a)$ and $Comp(P_r \cup P_q) \vdash \neg q(b)$, i.e. it is *iso*-closed.

Let us conclude with a quick look at normal programs, even though they are not the main concern of this paper. For a normal program $P$,[6] $\forall P$ and $Comp(P)$ may be not *ini* and not *iso*-closed.

[6] Here we consider only consistent normal programs.

**Example 3.5** Consider the program $P_3$:

$$\begin{array}{rcl} p(a) & \leftarrow & \neg q(a) \\ q(a) & \leftarrow & \neg p(a) \end{array} \qquad (P_3)$$

$\forall P_3$ and $Comp(P_3)$ are not *ini*- and not *iso*-closed. Indeed $P_3$ has three Herbrand models: $\{p(a)\}$, $\{q(a)\}$ and $\{p(a), q(a)\}$, and $Comp(P_3)$ has two Herbrand models: $\{p(a)\}$ and $\{q(a)\}$.

The reason for the non-closure of $P_3$ is circularity: $p$ is defined in terms of $q$, and $q$ in terms of $p$, in a non-well-founded way. Circularity is exposed by both initial and isoinitial semantics.

The same kind of circularity also occurs in the program $P_1$ of Example 3.2. In that case, however, circularity was exposed only by isoinitial semantics. In initial semantics, circular definite programs have an intended model, namely the empty model.

It is worthwhile to remark that we are interested in logic programs as first-order theories. In particular, we study the intended models of normal logic programs, but we do not deal with the soundness or completeness problems of $SLDNF$-resolution. For example, in an *iso*-closed program $P$, $Comp(P) \vdash \exists x \,.\, \neg A(x)$ entails $Comp(P) \vdash \neg A(t)$ for at least one ground term $t$ (see Theorem 3.1), while, with $SLDNF$-resolution, the open goal $\leftarrow \neg A(X)$ gives rise to the well-known problem of floundering.

# 4 Isoinitial Semantics for Open Logic Programs

A theory $T$ is *open* with respect to *sem*, or *sem*-open for short, if it is consistent and has no *sem*-model. We consider a *sem*-open theory as an incomplete axiomatisation of a *sem*-model, to be completed by adding new axioms and, possibly, new symbols to the signature.

Open theories are needed if we want to compose small well-defined theories to build new larger theories [7, 6]. In the case of programs, open programs are needed if we want to use and compose them as modules [3, 4]. In this paper, we do not treat the problem of theory or program composition. Rather, we study only the consequences of initial and isoinitial semantics in open theories and programs.

Dealing with open theories, we have to consider different signatures. A signature $\Sigma \equiv \langle F, R \rangle$ is a subsignature of $\Sigma' \equiv \langle F', R' \rangle$, written $\Sigma \subseteq \Sigma'$, if and only if $F \subseteq F'$ and $R \subseteq R'$.

For $\Sigma \subseteq \Sigma'$, we have the well known notions of *reduct* and *expansion*. The $\Sigma$-*reduct* of a $\Sigma'$-structure $\mathcal{N}$ is the $\Sigma$-structure $\mathcal{N}|\Sigma$ that has the same domain as $\mathcal{N}$ and interprets each symbol $s$ of $\Sigma$ in the same way as $\mathcal{N}$, i.e., $s^{\mathcal{N}|\Sigma} = s^{\mathcal{N}}$. Conversely, if $\mathcal{M} = \mathcal{N}|\Sigma$, $\mathcal{N}$ is said to be a $\Sigma'$-expansion of $\mathcal{M}$.

A well known property of reducts is that, for every $\Sigma$-formula $F$, $\mathcal{N} \models F$ iff $\mathcal{N}|\Sigma \models F$.

Now we can discuss open theories. In general, an open theory $O_1$ leaves open the intended meaning of some symbols and, possibly, the intended domain. $O_1$ is composed with another closed or open theory $O_2$, giving rise to a composite theory $O_1O_2$. The resulting theory may be further composed with $O_3$, and so on, until a final closed theory is obtained. This is what happens, for example, when theories are logic programs. Composing them yields a final closed program.

If composition is associative, i.e. $(O_1O_2)O_3 = O_1(O_2O_3)$, each sequence $O_1O_2 \cdots O_n$ is equivalent to a two-step sequence $O_1T$, with $T = O_2 \cdots O_n$. For example, composition of logic programs is associative. Moreover, in many interesting cases, if $O_1T$ is closed, then so is $T$. This is the case, for example, for program composition without mutual recursion. In the sequel we consider cases where we assume $T$ to be *sem*-closed.

Let us start from our *sem*-closed theory $T$, with *sem*-model $\mathcal{I}$. Let us assume that we add new constant, function or relation symbols. In general, the new symbols become *open symbols*, since, in the new language, $T$ is no longer *sem*-closed. In our previous discussion, they are closed by the axioms of $O_1$. The problem is: if we close $T$ by some $O_1$, what is preserved of the *sem*-model $\mathcal{I}$ of $T$? A first answer is:

**Theorem 4.1** Let $\Sigma \equiv \langle F, R\rangle$ be a signature, and $T$ be a *sem*-closed $\Sigma$-theory, with *sem*-model $\mathcal{I} \equiv \langle D, F^{\mathcal{I}}, R^{\mathcal{I}}\rangle$. Let $\Sigma' \equiv \langle F', R'\rangle$ be a larger signature, and $T'$ be a *sem*-closed theory containing $T$, with *sem*-model $\mathcal{I}'$. Then there is a unique *sem*-morphism $h : \mathcal{I} \to \mathcal{I}'|\Sigma$.

*Proof.* The proof follows easily from the fact that $T \subseteq T'$ and, hence, $\mathcal{I}'|\Sigma \models T$. □

Let us briefly consider the consequences of this theorem:

**Corollary 4.1** Let $T$, $T'$, $\mathcal{I}$, $\mathcal{I}'$ be as in Theorem 4.1. If *sem* is *ini*, then for every positive existential $\Sigma$-sentence $\exists(P)$, $\mathcal{I} \models \exists(P)$ entails $\mathcal{I}' \models \exists(P)$.

If *sem* is *iso*, then for every existential $\Sigma$-sentence $\exists(Q)$, $\mathcal{I} \models \exists(Q)$ entails $\mathcal{I}' \models \exists(Q)$. Moreover, for a quantifier-free $\Sigma$-sentence, $\mathcal{I} \models Q$ iff $\mathcal{I}' \models Q$.

*Proof.* The proof follows from Theorem 3.1, $\mathcal{I}'|\Sigma$ being a model of $T$. □

As we can see, *iso*-closed theories preserve truth and falsity of quantifier-free formulas. This corresponds to the fact that $\mathcal{I}$ is isomorphically embedded into $\mathcal{I}'|\Sigma$. In initial semantics, $\mathcal{I}$ is only guaranteed to be homomorphic.

When the domain is preserved, *iso*-closure has a strong consequence:

**Corollary 4.2** Let $T, T', \mathcal{I}, \mathcal{I}', h$ be as in Theorem 4.1. If *sem* is *iso* and $h$ is surjective, then, for every $\Sigma$-sentence $F$, $\mathcal{I} \models F$ iff $\mathcal{I}' \models F$.

*Proof.* A surjective isomorphic embedding is an isomorphism. □

Corollary 4.2 does not hold for initial semantics, because surjective homomorphisms are not necessarily isomorphisms.

For logic programs, Corollary 4.1 applies when we add new constant or function symbols, while Corollary 4.2 applies when we add only new predicates. This suggests that it is useful to start from a signature larger than the one of the program at hand, and containing all the possible function and constant symbols. In this way, if we work with *iso*-closed programs, then the strong property of Corollary 4.2 is guaranteed.[7]

However, fixing in advance the whole domain is too restrictive. A more reasonable alternative is to introduce many-sorted programs.

**Example 4.1** Let us consider the program *SumP* of Example 3.1, and let us introduce the following program for computing the sum of a list of natural numbers:

$$sumlist(nil, 0) \leftarrow, \quad \forall n, l, a, b \,.\, sumlist(n.l, a) \leftarrow sumlist(l, b) \wedge sum(n, b, a)$$

We have to add the list constructor $\cdot$ and the constant *nil*. If we stay in the one-sorted case, the predicate *sum* is automatically extended to the larger domain. Since *Comp*(*SumP*) is *iso*-closed, by the previous results we have that the old Minimum Herbrand Model is isomorphically embedded into the new one.[8] Therefore, truth of

[7] To apply Corollary 4.2 correctly, we have to check that the new *Comp*($P'$) entails the old one.

[8] In this case the new theory is *iso*-closed, but this is not guaranteed in general.

quantifier-free and existential closed formulas is preserved. However, this *iso*-complete axiomatisation may be incomplete with respect to formulas like $\neg exists(Q)$, i.e., the latter may be true in the original model but false in the new one. A possible solution is to introduce lists as a new sort. In this case the domain interpreting the sort of natural numbers does not change, *sum* cannot be applied to lists, and we are in a situation where Corollary 4.2 applies.

In general, we can show that Corollary 4.2 can be extended to the many-sorted case, and it holds whenever the domains interpreting the old sorts are preserved. This shows that introducing sorts is very useful, when dealing with program composition.

Parametrised theories are a particular case of open theories. A parametrised theory $T(P)$ is a theory with signature $\Sigma = \langle F, R\rangle$ and a set $P \subseteq F \cup R$ of parameters. Let us indicate by $\Sigma_P = \langle F \cap P, R \cap P\rangle$ the subsignature of the parameters. A $\Sigma_P$-structure $\mathcal{P}$ can be seen as a kind of parameter passing, and we can consider the $\mathcal{P}$-models of $T(P)$ determined by it.

**Definition 4.1** Let $T(P)$ be a $\Sigma$-theory, and $\mathcal{P}$ be a $\Sigma_P$-interpretation. A $\mathcal{P}$-*model* of $T(P)$ (if one exists) is a model $\mathcal{M}$ of $T$ such that $\mathcal{M}|\Sigma_P = \mathcal{P}$.

That is, $\mathcal{P}$-models are models that agree with the parameter passing $\mathcal{P}$.

**Definition 4.2** Let $\Pi \subseteq \Sigma$ be two signatures, $\mathcal{P}$ be a $\Pi$-structure, and $\mathcal{N}$ and $\mathcal{M}$ be two $\Sigma$-expansions of $\mathcal{P}$. A $\mathcal{P}$-*homomorphism* $h : \mathcal{M} \to \mathcal{N}$ is a homomorphism such that $h(s^{\mathcal{M}}) = h(s^{\mathcal{N}}) = h(s^{\mathcal{P}})$, for every symbol $s$ of $\Pi$. A $\mathcal{P}$-(*isomorphic embedding*) is a $\mathcal{P}$-*homomorphism* that preserves the complements of the relations.

That is, $\mathcal{P}$-homomorphisms and isomorphisms completely preserve the parameter passing $\mathcal{P}$ (i.e., they work as the identity over the parameters). $\mathcal{P}$-initial models and $\mathcal{P}$-isoinitial models are defined like initial and isoinitial models. The difference is that they use $\mathcal{P}$-homomorphisms and $\mathcal{P}$-isomorphisms.

**Definition 4.3** Let $\Sigma(\Pi) = \langle F, R\rangle$ be a signature and $T(P)$ be a parametric $\Sigma$-theory. $T(P)$ is *ini-parametric* if and only if, for every $\Sigma_P$-interpretation $\mathcal{P}$, the class $MOD_{\mathcal{P}}(T(P))$ of the $\mathcal{P}$-models of $T(P)$ contains a $\mathcal{P}$-initial model $\mathcal{I}_{\mathcal{P}}$. If $\mathcal{I}_{\mathcal{P}}$ is $\mathcal{P}$-isoinitial in $MOD_{\mathcal{P}}(T(P))$, then $T$ is *iso-parametric*.

All the model-theoretic results that we have shown for initial and isoinitial models extend to $\mathcal{P}$-initial and $\mathcal{P}$-isoinitial models, considering the class of $\mathcal{P}$-models of a theory. Here reachability is not required, since the domain of the $\mathcal{P}$-models is completely left to $\mathcal{P}$.

With respect to provability, an open theory in general does not prove any ground atomic formula, since relation symbols are left open. For example, an open program often 'always finitely fails'. We have to complete the theory, by adding a set $Ax$ of new axioms, that characterises a parameter passing $\mathcal{P}$. Here we have the following *sufficient completeness problem*: how much of $\mathcal{P}$ is to be codified by $Ax$, in order to obtain a *sem*-complete theory $T(P) \cup Ax$? Indeed, in general, *sem*-completeness of $Ax$ does not suffice. It turns out that the use of constructive systems allows us to develop a proof theory for stating *iso*-parametricity, based on Corollary 2.3 (see [10]).

Moreover, the above notion of parametricity may be too restrictive. There are cases where it suffices to consider only particular classes of $\mathcal{P}$. For example, if we have already in mind a reachable domain where we want to interpret the relation symbols of $T(P)$, we can consider only those $\mathcal{P}$ that have this domain. However, these aspects are not the main concern of this paper, so we will not discuss further details about parametricity.

Now, we consider open logic programs. If a predicate $q(x)$ is not definite[9] in a program $P$, then we will consider it an *open* predicate. To handle such predicates, we introduce the *open completion* $Ocomp(P)$, that contains $Cdef^-(p)$ for every defined predicate $p$, but does not contain $Cdef^-(q) = \forall x \neg q(x)$, for the open predicates.

**Example 4.2** Let us consider the program $P_{even}$

$$even(0), \forall x\, .\, even(s(x)) \leftarrow odd(x)$$

It is reasonable to consider *odd* as an open symbol. The open completion contains $Cdef^-(even)$, i.e. $\forall x\, .\, \neg even(s(x)) \leftarrow \neg odd(x)$, but it does not contain $\forall x\, .\, \neg odd(x)$. More precisely, the open completion contains $P_{even}$, $Cdef^-(even)$ and $CET(0, s)$.

We can show that $P_{even}$ is parametric with respect to *odd*, at least in the class of interpretations of *odd* over the Herbrand structure corresponding to $CET(0, s)$. Thus we have a parametric theory $Ocomp(P_{even})(odd)$. The signature of the parameters is $\Sigma_P = \langle \{\}, \{odd\} \rangle$.

**Theorem 4.2** Let $P$ be a program with at least one open predicate, and with a non-empty Herbrand Universe. Then $Ocomp(P)$ is *ini*-closed, but it is not *iso*-closed.

We omit the easy proof. This theorem shows that, while initial semantics does not expose the fact that some information is missing, isoinitial semantics does: $Ocomp(P)$ is not *iso*-complete, hence it is not *iso*-closed.

For example, $Ocomp(P_{even})(odd)$ is *iso*-open. We can close it by $\forall x\, .\, odd(s(x)) \leftarrow even(x)$. Here mutual recursion is well-founded, i.e. non-circular. Circularity would be exposed by the lack of an *iso*-initial model, since *iso*-initiality is related to existential ground-termination.

The next example discusses the problem of sufficient completeness:

**Example 4.3** Consider the open completion of the program $P_q$ of Example 3.4:

$$\forall x, y\, .\, q(x) \leftarrow r(x, y), \forall x\, .\, \neg q(x) \leftarrow \neg(\exists y\, .\, r(x, y))$$

The parametric $Ocomp(P_q)(r)$ is *ini*-closed, and the empty model is its initial model. It is not *iso*-closed. It is *ini*-parametric and *iso*-parametric. Indeed, the signature of the parameters is $\Pi = \langle \{\}, \{r^2\} \rangle$, and for every $\Pi$-interpretation $\mathcal{P}$ there is one $\mathcal{P}$-initial and $\mathcal{P}$-isoinitial model of $Ocomp(P)$. For example, if the domain of $\mathcal{P}$ is $\{a, b\}$ and $\exists y\, .\, r(x, y)$ holds in $\mathcal{P}$ if and only if $x = a$, then the corresponding $\mathcal{P}$-isoinitial (and $\mathcal{P}$-initial) model is the expansion of $\mathcal{P}$ interpreting $q(a)$ as true and $q(b)$ as false.

Now, let us assume that we have a program $Q$. As already remarked in Example 3.4, to obtain an *ini*-complete $Comp(P \cup Q)$, it suffices that $Q$ is *ini*-complete. However, if we want *iso*-completeness, we need $Q$ to be complete with respect to the sentences $\neg \exists y\, .\, q(t, y)$, called *sufficient completeness requirements*. Therefore, with respect to parametricity, initial semantics also works in a larger class of cases, but yields less information.

[9]I.e. it does not occur in the head of any clause.

# 5 Conclusion

Starting from the general theory of isoinitial models [2], we have presented some preliminary results on initial and isoinitial theories and models. They are oriented towards our approach to modular program synthesis, but, we believe, they are interesting in general. The traditional view of a definite logic program (e.g., [5]) treats it as an initial theory. This in our opinion is too restrictive because it basically takes the Closed World view and does not provide a uniform semantics for negation and open programs, or parametricity. This view is therefore very much one of *programming-in-the-small*. For normal programs, other kinds of semantics have been proposed (for a survey see e.g. [1]). A comparison with these semantics is one of our next steps.

Our (preliminary) results on isoinitial theories and models are motivated by a search for a suitable uniform semantics for both *programming-in-the-large* and *programming-in-the-small*. We believe that isoinitial semantics fits the bill, for logic programs. It handles not only negation but also parametricity in a uniform manner with respect to both closed and open programs. Moreover, constructive formal systems can help to formally prove isoinitiality.

Clearly such a uniform semantics is important if logic programming is to be used for large-scale software development. Indeed we believe this semantics is a unique feature and a great advantage of the logic programming paradigm. We have already used isoinitial semantics in our work in formal program development, and more recently in component-based software development, in computational logic. It is within this context that we plan to continue our study of isoinitial theories and models.

## Acknowledgements

We wish to thank the referees for their comments that have helped us to improve this paper.

# References

[1] K.R. Apt and R. Bol. Logic programming and negation: a survey. *J. Logic Programming* 19,20:9-71, 1994. Special issue: Ten years of logic programming.

[2] A. Bertoni, G. Mauri and P. Miglioli. On the power of model theory in specifying abstract data types and in capturing their recursiveness. *Fundamenta Informaticae* VI(2):127–170, 1983.

[3] A. Brogi, P. Mancarella, D. Pedreschi and F. Turini. Modular logic programming. *ACM TOPLAS* 16(4):1361–1398, 1994.

[4] M. Bugliesi, E. Lamma and P. Mello. Modularity in logic programming. *J. Logic Programming* 19,20:443–502, 1994. Special issue: Ten years of logic programming.

[5] W. Hodges. Logical features of horn clauses. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson. editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, *Volume 1: Logical Foundations*, pages 449–503, Oxford University Press, 1993.

[6] C. Kreitz, K.K. Lau, and M. Ornaghi. Formal reasoning about modules, reuse and their correctness. In D.M. Gabbay and H.J. Ohlbach, editors, *Proc. Int. Conf. on Formal and Applied Practical Reasoning*, *LNAI* 1085, pages 384–399, Springer-Verlag, 1996.

[7] K.-K. Lau and M. Ornaghi. Towards an object-oriented methodology for deductive synthesis of logic programs. In M. Proietti, editor, *Proc. LOPSTR 95*, *LNCS* 1048:152–169, Springer-Verlag, 1996.

[8] K.-K. Lau and M. Ornaghi. A formal approach to deductive synthesis of constraint logic programs. In J.W. Lloyd, editor, *Proc. 1995 Int. Logic Programming Symp.*, pages 543–557, MIT Press, 1995.

[9] P. Miglioli, U. Moscato and M. Ornaghi. Constructive theories with abstract data types for program synthesis. In D.G. Skordev, editor, *Mathematical Logic and its Applications*, pages 293–302, Plenum Press, 1986.

[10] P. Miglioli, U. Moscato and M. Ornaghi. Abstract parametric classes and abstract data types defined by classical and constructive logical methods. *J. Symb. Comp.* 18:41–81, 1994.

[11] R. Reiter. On closed world data bases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, 1978.