Isoinitial Semantics for Logic Programs

Kung-Kiu Lau[1] and Mario Ornaghi[2]

[1] Dept. of Computer Science, University of Manchester, United Kingdom
kung-kiu@cs.man.ac.uk

[2] Dip. di Scienze dell'Informazione, Universita' degli studi di Milano, Italy
ornaghi@dsi.unimi.it

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 (program) proves (computes) only positive literals (atoms) in P, so it does not deal with negation. In this paper, we introduce *isoinitial* semantics for logic programs and show that it can provide a rich semantics for logic programs, which can deal with not just negation, but also incomplete information, parametricity and compositionality.

We dedicate this paper to the memory of the originator of isoinitial semantics:
Pierangelo Miglioli (1946–1999).

1 Introduction

The intended model of a definite logic program P is its Herbrand model H. It interprets P under the Closed World Assumption [16]. Among the class of all the models of P, H interprets P as an *initial* theory [9]. A distinguishing feature of an initial theory P is that, in general, it proves (computes) only positive literals in P, so it does not deal with negation. One way to handle negation is to consider program *closures* (e.g. program completion). In this paper, we introduce *isoinitial* semantics [2] for (definite) logic programs, and we show that isoinitial closures are better able to handle negation than initial closures, and are, in general, richer than initial closures from the point of view of *incomplete information*, *parametricity* and *compositionality*.

We will discuss full first-order theories in general, and consider definite programs as a particular case, with its own peculiarities. Since it admits any kind of axioms, our treatment also applies to *normal* and *disjunctive* programs (which for lack of space we will only briefly mention in the Conclusion). Finally, we shall assume familiarity with the general terminology for logic programming, and refer readers to standard works such as [12] for this terminology.

The results of Sections 2 and 3 adapt the general results of [2, 14] to logic programs (for the first time), while the other results are mainly new.

2 Initial and Isoinitial Models

In this section, we formally define, and exhibit examples of, theories with initial and isoinitial models. We show how we can characterise such models, and for isoinitial models, we state useful conditions for proving isoinitiality.

A Σ-theory is a set of Σ-sentences, where $\Sigma \equiv \langle F, R\rangle$ is a signature Σ with function symbols F and relation symbols R, 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]

[1] σ^n denotes a symbol σ with arity n.

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

Let $\Sigma \equiv \langle F, R\rangle$ be a signature. As usual, a *Σ-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 $f^{n\mathcal{M}} : D^n \to D$, 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 denote the value of a ground term t in $\mathcal{M}$, and $\mathcal{M} \models A$ will indicate that the sentence (sentences) A is (are) true in $\mathcal{M}$. A theory T is a set of sentences, and a *model* of T is a structure $\mathcal{M}$ such that $\mathcal{M} \models T$.

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.

Definition 1. (Isomorphic Embeddings). An *isomorphic embedding* $i : \mathcal{J} \to \mathcal{M}$ is a homomorphism between the structures $\mathcal{J}$ and $\mathcal{M}$ 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, $\mathcal{J}$ is isomorphic to a substructure of $\mathcal{M}$ (viz. the i-image of $\mathcal{J}$), i.e. $\mathcal{J}$ is 'isomorphically embedded' in $\mathcal{M}$.

Now we can define initial and isoinitial models of Σ-theories.

Definition 2. (Initial Models). Let T be a first-order Σ-theory, and $\mathcal{I}$ be a model of T. $\mathcal{I}$ is an *initial model* of T iff, for every model $\mathcal{M}$ of T, there is a unique homomorphism $h : \mathcal{I} \to \mathcal{M}$.

Definition 3. (Isoinitial Models). Let T be a first-order Σ-theory, and $\mathcal{J}$ be a model of T. $\mathcal{J}$ is an *isoinitial model* of T iff, for every model $\mathcal{M}$ of T, there is a unique isomorphic embedding $i : \mathcal{J} \to \mathcal{M}$.

Example 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.

In contrast, H is an isoinitial model of the theory $\{\neg a = b\}$. 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 $\mathcal{M}$.

In fact H is also an initial model of $\{\neg a = b\}$.

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

In the rest of the paper, we will consider *only* the particular case of *reachable* initial and isoinitial models. The treatment of non-reachable models requires concrete data [2] that are omitted for lack of space; isoinitiality entails initiality in the reachable case, whereas in the general case the two notions are independent.

Definition 4. (Reachable Models). A structure (model) $\mathcal{M} \equiv \langle D, F^{\mathcal{M}}, R^{\mathcal{M}} \rangle$ is *reachable* if, for every $\alpha \in D$, there is a ground term t such that $t^{\mathcal{M}} = \alpha$.

We can characterise reachable initial and isoinitial models as follows:

Theorem 1. *Let $\mathcal{J}$ be a reachable model of a Σ-theory T. Then $\mathcal{J}$ is an* initial *model of T if and only if the following* initiality condition *holds:*

for every ground atom A, $\mathcal{J} \models A$ iff ($\mathcal{M} \models A$, for every model $\mathcal{M}$ of T) (INI)

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

for every ground literal L, $\mathcal{J} \models L$ iff ($\mathcal{M} \models L$, for every model $\mathcal{M}$ of T). (ISO)

(The proof follows from the reachability hypothesis. We omit it for conciseness.)

Thus initial models represent *truth* of atomic formulas in every model, while isoinitial models represent both *truth* and *falsity* of atomic formulas in every model.

Now by the completeness theorem for first-order theories, we can prove:

Corollary 1. *In Theorem 1, we can replace* (INI) *and* (ISO) *by* (INI') *and* (ISO')*:*

for every ground atom A, $\mathcal{J} \models A$ iff $T \vdash A$. (INI')

for every ground literal L, $\mathcal{J} \models L$ iff $T \vdash L$. (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 could apply the above theoretical (characterisation) results. However, for isoinitial models, we can state a condition that is more useful in practice for proving isoinitiality:

Corollary 2. *In Theorem 1, we can replace* (ISO) *by the following* atomic completeness *condition: for every ground atom A, $T \vdash A$ or $T \vdash \neg A$.* (ATC)

In condition (ATC) models disappear altogether,[3] i.e., we have a purely proof-theoretic condition. This allows us to prove some interesting and useful results, that link classical and constructive proof theoretical properties with isoinitial models [14]. For example, we can prove the following theorem:

Theorem 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 symbol. Let H_F be the corresponding Herbrand structure, and let $CET(F)$ be Clark's Equality Theory for F. Then H_F is an isoinitial model of $CET(F)$.*

Proof. H_F is a model of $CET(F)$, and, 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 t and t' are different.

[3] However, models do not disappear completely from Theorem 1, because the existence of at least one reachable model $\mathcal{J}$ is always assumed.

3 Initial and Isoinitial Semantics for Closed Theories

In this section, we consider closed theories and their initial and isoinitial models. For conciseness, we will use the abbreviations *ini* for *initial semantics* (i.e. semantics based on initial models), *iso* for *isoinitial semantics* (i.e. semantics based on isoinitial models) and *sem* for a parameter standing for either *ini* or *iso*.

We will show that, for closed theories, *iso* is better able than *ini* to deal with *negation*. In the next section, we will show that for the special case of closed definite logic programs, *iso* also allows us to reason about *termination*.

Closed first-order Σ-theories are defined as follows:

Definition 5. (*Sem*-closed Σ-theories). A Σ-theory T is *sem-closed* if and only if it has a (reachable) *sem*-model.

Example 2. Consider the signature $K \equiv \langle F, \{\} \rangle$ (in Theorem 2) containing a non-empty set F of function and constant symbols, with at least one constant symbol. Let H_F be the corresponding Herbrand structure.

The empty theory $\emptyset$ is *ini*-closed, and H_F is an initial model of $\emptyset$. By contrast, $\emptyset$ is not *iso*-closed (it is not atomically complete). However, $CET(F)$ is an *iso*-closed theory, with isoinitial model H_F (see Theorem 2).

In general, *iso* is better equipped than *ini* to deal with *negation*: in Example 2, *iso* shows that $\emptyset$ lacks information with regard to negation, whereas *ini* does not show this. *ini* and *iso* correspond to two different ways of looking at negation, and, more generally, at the role of axioms. Indeed, as a corollary of (INI') and (ISO'), we get the following properties:

If $\mathcal{I}$ is an initial model, then for every ground atom A,

$$\mathcal{I} \models \neg A \text{ iff } T \nvdash A. \qquad \text{(INI'')}$$

If it is isoinitial, then for every ground atom A,

$$\mathcal{J} \models \neg A \text{ iff } T \vdash \neg A. \qquad \text{(ISO'')}$$

(INI'') generalises the Closed World Assumption (CWA) [16] to general first-order theories: a fact is *false* (in the intended reachable initial model) if and only if it cannot be proved.

(ISO'') corresponds to Constructive Negation (CN) [6]: a fact is *false* (in the intended model) if and only if its *negation* can be proved.

Thus *ini* and *iso* subscribe to different views. *Ini* corresponds to a principle of *economy*: we look for the smallest set of axioms that allows us to derive the *positive* facts. In contrast, *iso* semantics corresponds to a principle of *richness*: we look for rich theories, that allow us, at least, to treat *negation* constructively.

Another remarkable difference is shown by the following corollary of (INI') and (ISO'):

Corollary 3. *Let T be sem-closed, and let $\exists x Q(x)$ be an existential sentence such that $T \vdash \exists x Q(x)$.*

If sem is ini and $Q(x)$ is a positive *quantifier-free formula, then $T \vdash Q(t)$, for at least one ground t.*

If sem is iso and $Q(x)$ is any *quantifier-free formula, then $T \vdash Q(t)$, for at least one ground t.*

That is, an *iso*-closed theory T is rich enough to prove the answers $\neg A(t)$ of negative existential queries $\exists x \neg A(x)$ that hold in the isoinitial model, whereas this is not guaranteed, in general, for *ini*-closed theories: the latter are guaranteed to answer only *positive* existential queries. Moreover, as is well known, with CWA the set of false ground atoms may not be recursively enumerable,whereas reachable isoinitial models are guaranteed to be computable [2]. Negation and computability are not the only reasons that induced us to consider isoinitial semantics, however. As we will show in forthcoming sections, we will also use isoinitial semantics to deal with *incomplete information, parametrisation and modularity.*

4 Closed Logic Programs

Now, we consider closed logic programs as closed theories. For lack of space we will focus on *definite* programs. Consequently we do not discuss normal programs, but we will show that even for definite programs, *iso* has another advantage of being able to reason about *termination.*

For a signature $\Sigma = \langle F, R \rangle$ with at least one constant symbol, interpretations over the term-domain H_F will be called *Herbrand Σ-structures* (or *Σ-interpretations*). As usual, a Herbrand Σ-interpretation can be uniquely represented as a set of ground atoms. Thus we will consider Herbrand Σ-structures as term models or as sets of ground atoms, interchangeably.

Let P be a logic program, with signature $\Sigma_P \equiv \langle F_P, R_P \rangle$. We will use $Cdef(p)$ to denote the completed definition of p in P (see, e.g., [12]). We will also use $Cdef(P)$ to denote the set of $Cdef(p)$, for $p \in R_P$.

The completion of P, $Comp(P)$, is then the union $CET(F_P) \cup Cdef(P)$.

Example 3. Consider the usual program P_{sum}:

$$\begin{aligned} &sum(x, 0, x) \leftarrow \\ &sum(x, s(i), s(v)) \leftarrow sum(x, i, v) \end{aligned} \qquad (P_{sum})$$

for the sum of natural numbers (s is *successor*), with signature $\Sigma_{P_{sum}} \equiv \langle \{0^0, s^1\}, \{sum^3\} \rangle$. In this case, $CET(0^0, s^1)$ contains the axioms[4]

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

The completed definition of sum, $Cdef(sum)$, is (after some obvious simplifications): $\forall x, y, z \,.\, sum(x, y, z) \leftrightarrow (y = 0 \wedge z = x) \vee (\exists i, v \,.\, y = s(i) \wedge z = s(v) \wedge sum(x, i, v)))$.

The completion of P_{sum} is $Comp(P_{sum}) = CET(0^0, s^1) \cup Cdef(sum)$.

The minimum Herbrand model $\mathcal{M}(P)$ is defined in the usual way. We have the following theorem:

Theorem 3. *For a (definite) logic program P, $\mathcal{M}(P)$ 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 [9]. $\mathcal{M}(P)$ cannot be an isoinitial model of P, because P cannot be atomically complete (no negated formula is provable from it).

We might expect $\mathcal{M}(P)$ to be an isoinitial model of $Comp(P)$, but this is not necessarily so, as shown by the following example:

[4] Here $s^{(n)}$ denotes the iteration of s for n times.

Example 4. Consider the program P_1 :

$$p(a) \leftarrow q(a) \qquad (P_1)$$
$$q(a) \leftarrow p(a)$$

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

For $Comp(P_1)$ to have a reachable isoinitial model, 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 for $Comp(P_1)$.

On the other hand, $\mathcal{M}(P_1)$ (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.

Now consider the program P_2: $\quad p(a) \leftarrow q(a) \qquad (P_2)$

$Comp(P_2)$ is both *ini*- and *iso*-closed. Indeed, now $Cdef(q)$ is $\forall x \,.\, \neg q(x)$, and we can prove $\neg q(a)$ and $\neg p(a)$. $\mathcal{M}(P_2)$ is the same as $\mathcal{M}(P_1)$, but it is both initial and isoinitial for $Comp(P_2)$.

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

4.1 Termination

Example 4 suggests that termination (see [7] for a survey) and *iso*-closure are related. Indeed, we can prove the following result:

Definition 6. (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 4. *Let P be a definite program with a non-empty Herbrand universe. $Comp(P)$ is iso-closed if and only if P existentially ground-terminates.*

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

It follows that we can use termination analysis for stating isoinitiality.

Example 5. Consider the program P_{sum} in Example 3. We can prove that P_{sum} existentially ground-terminates. Therefore, we can conclude that its minimum Herbrand model is an isoinitial model of $Comp(P_{sum})$.

Moreover, the converse also holds, i.e. we can study existential ground-termination by studying isoinitiality, as the following example shows.

Example 6. Consider the following program P_{path}:

$$\left.\begin{array}{l} path(x,y) \leftarrow arc(x,y) \\ path(x,y) \leftarrow arc(x,z), path(z,y) \\ arc(1,1) \;\leftarrow \\ arc(2,2) \;\leftarrow \end{array}\right. \qquad (P_{path})$$

Its completion contains the axioms:

$$\neg 1 = 2$$
$$\forall x, y \,.\, path(x,y) \leftrightarrow arc(x,y) \vee \exists z \,.\, arc(x,z) \wedge path(z,y)$$
$$\forall x, y \,.\, arc(x,y) \leftrightarrow (x = 1 \wedge y = 1) \vee (x = 2 \wedge y = 2).$$

Instead of considering termination, we apply directly a model theoretic argument. We consider two term models $\mathcal{M}_1$ and $\mathcal{M}_2$. Both interpret *arc* in the same way, according to the axioms. In $\mathcal{M}_1$, the meaning of $path(x, y)$ is: "there is a finite path connecting x to y". In $\mathcal{M}_2$, the meaning of $path(x, y)$ is: "there is a finite path connecting x to y, or there is an infinite path starting from x". As we can see, both satisfy the axiom for *path*. In the first one, $path(1, 2)$ is false, while in the second one it is true. Therefore, we cannot have an isoinitial model. This entails that P_{path} does not existentially terminate for the goal $\leftarrow path(1, 2)$.

This model-theoretic argument applies to any graph, that is, we can define $\mathcal{M}_1$ and $\mathcal{M}_2$ that satisfy the axiom for *path* and interpret $path(a, b)$ in two different ways, if there is a cycle starting from a, but no finite path from a to b. In contrast, if the graph is finite and acyclic, or if it is infinite but *arc* is well founded, then both these interpretations interpret $path(a, b)$ in the same way for every a and b. On the other hand, for acyclic finite graphs, or infinite graphs with a well founded *arc*, the recursive clause for *path* existentially terminates.

5 Initial and Isoinitial Semantics for Open Theories

In this section we consider *sem*-open theories, namely theories without *sem*-models, and we will compare initial and isoinitial semantics from the point of view of *incomplete information*, *parametricity* and *compositionality*. We will show that *iso*-closures provide a rich semantics for these.

5.1 Incomplete Information

Definition 7. (*sem*-open Σ-theories). A Σ-theory T is *sem-open* if and only if it is consistent but has no *sem*-model.

We consider a *sem*-open Σ-theory as an incomplete axiomatisation of a *sem*-model, to be completed by adding new axioms and, possibly, new symbols to the signature Σ.

Here we will consider the simpler case where Σ is fixed, i.e., no new symbol is added. In this case, it is interesting to consider the *sem*-closures of a *sem*-open theory T, i.e., the theories T' that contain T,[5] and are *sem*-closed. The minimal *ini*-closures give rise to a poorer semantics compared to *iso*-closures, as we show in the following example.

Example 7. The theory $T_{disj} = \{p(a) \vee p(b)\}$ with signature $\langle\{a^0, b^0\}, \{p^1\}\rangle$ is both *ini*- and *iso*-open. In general, positive occurrences of $\vee$ and $\exists$ in the axioms give rise to theories that are both *ini* and *iso*-open.

T_{disj} has two minimal *ini*-closures, namely $T_{disj} \cup \{p(a)\}$ and $T_{disj} \cup \{p(b)\}$, and four minimal *iso*-closures, namely $T_{disj} \cup \{a = b\}$, $T_{disj} \cup \{\neg a = b, p(a), \neg p(b)\}$, $T_{disj} \cup \{\neg a = b, \neg p(a), p(b)\}$, $T_{disj} \cup \{\neg a = b, p(a), p(b)\}$.

This example shows that initial semantics allows more compact closures. However, (as we will see later in Example 8 in Section 6) isoinitial semantics is better at showing up missing or incomplete information. Dealing with incomplete information is an important issue for databases [8].

[5] In the sense that $Theorems(T') \supseteq Theorems(T)$.

5.2 Parametricity

In this section, we introduce parametrised theories as a particular case of open theories.

A *parametrised Σ-theory* $T(\Pi)$ is a theory with signature $\Sigma = \langle F, R \rangle$ and a set $\Pi \subseteq F \cup R$ of parameters. We denote by $\Sigma_\Pi = \langle F \cap \Pi, R \cap \Pi \rangle$ the subsignature of the parameters.

A Σ_Π-structure $\mathcal{P}$ can be seen as a kind of parameter passing. To do so, we have to consider interpretations over 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 *Σ-reduct* of a Σ'-structure $\mathcal{N}$ is the Σ-structure $\mathcal{N}|\Sigma$ that has the same domain as $\mathcal{N}$ and interprets each symbol s of Σ 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 *Σ'-expansion* of $\mathcal{M}$.

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

Now we can define the semantics of a parametrised theory $T(\Pi)$ by considering its $\mathcal{P}$-models:

Definition 8. ($\mathcal{P}$-models). Let $T(\Pi)$ be a Σ-theory, and $\mathcal{P}$ be a Σ_Π-interpretation. A *$\mathcal{P}$-model* of $T(\Pi)$ (if one exists) is a model $\mathcal{M}$ of T such that $\mathcal{M} \,|\, \Sigma_\Pi = \mathcal{P}$.

That is, $\mathcal{P}$-models are models that agree with the parameter passing $\mathcal{P}$. We can define $\mathcal{P}$-initial models and $\mathcal{P}$-isoinitial models in a similar manner to initial and isoinitial models. The difference is that here we use $\mathcal{P}$-homomorphisms and $\mathcal{P}$-isomorphisms.

Definition 9. ($\mathcal{P}$-homomorphism and $\mathcal{P}$-(isomorphic embedding)). Let $\Pi \subseteq \Sigma$ be two signatures, $\mathcal{P}$ be a Π-structure, and $\mathcal{N}$ and $\mathcal{M}$ be two Σ-expansions of $\mathcal{P}$. A *$\mathcal{P}$-homomorphism* $h : \mathcal{M} \to \mathcal{N}$ is a homomorphism such that $h(s^{\mathcal{M}}) = s^{\mathcal{M}} = s^{\mathcal{P}}$, for every symbol s of Π. A $\mathcal{P}$-(*isomorphic embedding*) is a *$\mathcal{P}$-homomorphism* that preserves the complements of the relations.

Thus $\mathcal{P}$-homomorphisms and isomorphisms completely preserve the parameter passing $\mathcal{P}$, i.e., they work as identity over the parameters.

Now we can define parametric theories:

Definition 10. Let $\Sigma = \langle F, R \rangle$ be a signature and $T(\Pi)$ be a parametric Σ-theory. $T(\Pi)$ is *ini-parametric* if and only if, for every Σ_Π-interpretation $\mathcal{P}$, the class $MOD_{\mathcal{P}}(T(\Pi))$ of the $\mathcal{P}$-models of $T(\Pi)$ contains a $\mathcal{P}$-initial model $\mathcal{I}_{\mathcal{P}}$. If $\mathcal{I}_{\mathcal{P}}$ is $\mathcal{P}$-isoinitial in $MOD_{\mathcal{P}}(T(\Pi))$, 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. 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 $\mathcal{P}$ has to be codified by Ax, in order to obtain a *sem*-complete theory $T(\Pi) \cup Ax$? It turns out that the use of constructive systems allows us to develop a proof theory for stating *iso*-parametricity (see [15]).

Another possibility is to characterise the minimal *iso*-closures, as we will see in Section 6.

5.3 Compositionality

Parametrised theories can be used for composing small well-defined theories into new larger theories [10]. Similarly, parametrised programs[6] can be used (and composed) as modules [4, 5]. In this section, we study only the consequences of initial and isoinitial semantics for composition.

In general, a parametrised theory O_1 leaves open the intended meaning of its parameters and, possibly, the intended domain. O_1 can be used to produce a larger, composite theory, by composing it successively with other (closed or parametrised) theories O_2, O_3, ... We can make sure that the final composite theory $O_1 O_2 O_3 \ldots$ is closed if we choose suitable closed sub-theories for the composition.

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

So we start from a *sem*-closed theory T, with *sem*-model $\mathcal{I}$, and consider adding new constant, function or relation symbols to T. In general, the new symbols are *open symbols*, since, in the new language, T is no longer *sem*-closed. In our previous discussion, these symbols can be closed by the axioms of some O_1, but the question is: what is preserved of the *sem*-model $\mathcal{I}$ of T, by the *sem*-model $\mathcal{I}'$ of $O_1 T$?

The following theorem provides a first answer:

Theorem 5. *Let $\Sigma \equiv \langle F, R \rangle$ be a signature, and T be a sem-closed Σ-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 $h : \mathcal{I} \to \mathcal{I}'|\Sigma$.*

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

The consequences of this theorem are the following:

Corollary 4. *Let T, T', $\mathcal{I}$, $\mathcal{I}'$ be as in Theorem 5. If sem is ini, then for every positive existential Σ-sentence $\exists(W)$, $\mathcal{I} \models \exists(W)$ entails $\mathcal{I}' \models \exists(W)$. If sem is iso, then for every existential Σ-sentence $\exists(Q)$, $\mathcal{I} \models \exists(Q)$ entails $\mathcal{I}' \models \exists(Q)$. Moreover, for a quantifier-free Σ-sentence, $\mathcal{I} \models Q$ iff $\mathcal{I}' \models Q$.*

The proof follows from Theorem 5, $\mathcal{I}'|\Sigma$ being a model of T.

This means that *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:

[6] Usually called open programs.

Corollary 5. *Let T, T', $\mathcal{I}$, $\mathcal{I}'$, h be as in Theorem 5. If sem is iso and h is surjective, then, for every Σ-sentence S, $\mathcal{I} \models S$ iff $\mathcal{I}' \models S$.*

The proof follows from the fact that a surjective isomorphic embedding is an isomorphism.

Corollary 5 does not hold for initial semantics, because surjective homomorphisms are not necessarily isomorphisms.

Corollary 4 applies when we add new constant or function symbols, while Corollary 5 applies when we add only new predicates. This suggests that it is useful to start from a large signature, that contains all the possible function and constant symbols. Another, more reasonable alternative is to introduce many-sorted theories. In general, we can show that Corollary 5 can be extended to the many-sorted case, and it holds whenever the domains interpreting the old sorts are preserved, even if the larger signature contains new sorts, interpreted as new domains.

6 Open Logic Programs

Now, we consider open logic programs as *sem*-open theories. In general, the domain and/or some predicates are open. Predicates may be open because they are incompletely axiomatised, or because they are used as parameters.

6.1 Incomplete Information

In a program with incomplete information, open predicates are incompletely axiomatised.

Example 8. Consider the following informal specification:

Every bird flies, if it is normal; tweety and pingu are birds.

We do not know the entire domain of birds, and we do not say whether or not *tweety* and *pingu* are normal, that is, we have incomplete information. We could codify this situation by the open program P_{bird}:

$$\begin{aligned} flies(x) &\leftarrow bird(x), normal(x) \\ bird(tweety) &\leftarrow \\ bird(pingu) &\leftarrow \end{aligned} \qquad (P_{bird})$$

We leave *normal* as an open symbol, without any clause for it, because we do not have any information on it. *bird* is only partially axiomatised, i.e., its clauses are intended to fix it only in the known universe. Nevertheless, in contrast, the first clause is intended to *completely* define $flies$, since this happens in our informal problem.

P_{bird} is *ini*-closed. On the other hand, P_{bird} is *iso*-open, but there are minimal *iso*-closures that are at variance with our intention that $flies$ is completely defined by the program. That is, initial semantics does not directly expose the presence of open symbols, while isoinitial semantics shows both that there is incomplete information and that the axiomatisation is too weak (in *iso*) to axiomatise our informal open problem. On the other hand, pure Horn clauses are not sufficiently expressive with respect to *iso*.

To get a more expressive language, we introduce the *open completion*, denoted by $Ocomp(P)$. Based on the idea that non-open predicates are completely defined by P, $Ocomp(P)$ contains the completed definition $Cdef(p)$ for every non-open predicate p, the clauses for open predicates q in P, and $CET(F_P)$ for the function and constant symbols of P.

Let P be a program with at least one open predicate, and with a non-empty Herbrand Universe. We can easily prove that $Ocomp(P)$ is *ini*-closed, but it is not *iso*-closed. That is, even using the open completion, initial semantics does not expose the fact that some information is missing, while isoinitial semantics does.

We consider the isoinitial models of the minimal *iso*-closures of $Ocomp(P)$ in a (possibly partial) domain as the *intended models* of the open program P in that domain. In this sense, $Ocomp(P)$ represents the interpretation of an open program P in the isoinitial semantics.

Example 9. In the informal problem, *bird* and *normal* are open. Thus, the open completion $Ocomp(P_{bird})$ of P_{bird} is: $\{\neg tweety = pingu,\ \forall x\,.\,flies(x) \leftrightarrow bird(x) \wedge normal(x),\ bird(tweety),\ bird(pingu)\}$. $Ocomp(P_{bird})$ is *ini*-closed, and in its initial model *tweety* and *pingu* do not fly. In contrast, $Ocomp(P_{bird})$ is *iso*-open. There are four minimal *iso*-closures, that is, four different ways of completing the information within the known domain $\{tweety, pingu\}$. One is:

$Ocomp(P_{bird}) \cup \{normal(tweety), \neg normal(pingu)\}$.

In its isoinitial model $\mathcal{I}_{bird}$, *tweety* flies, but *pingu* does not. By isoinitiality, if we add further individuals and further closed information on them, we get larger models, that contain (an isomorphic copy of) $\mathcal{I}_{bird}$ as a substructure.

6.2 Parametricity

In a parametric program P, *open* predicates occur in the body of clauses and act as parameters. P can be 'closed' by composition with different (closed, non-parametric) Q's, that compute the open predicates in different ways.

We assume that P completely specifies all its defined predicates, and parameters are *exactly* the predicates that are *not* defined by P. Moreover, the domain may be open, (i.e. we also consider the constant and function symbols (if any) as parameters), to be completed into a larger signature.

Let P be a parametric program with signature $\Sigma_P = \langle F_P, R_P \cup O\rangle$, where F_P are the constant and function symbols (if any) of P, and R_P are its defined predicates. We will use $\Pi_P = \langle F_P, O\rangle$ to denote the *parameter signature* of P.

Theorem 6. *A program P with parameter signature Π_P is ini-parametric.*

That is, for every Π_P-interpretation $\mathcal{P}$, there is a $\mathcal{P}$-initial model of P. It is easy to show that such a $\mathcal{P}$-initial model is the minimum $\mathcal{P}$-model[7] of P. This result holds for every interpretation $\mathcal{P}$, i.e., we do not necessarily require that $\mathcal{P}$ is reachable in the signature of program P.

Example 10. Consider the open program P_{times}:

$$\begin{aligned} times(x,0,z) &\leftarrow u(z) \qquad\qquad (P_{times})\\ times(x,s(y),z) &\leftarrow times(x,y,w), q(x,w,z) \end{aligned}$$

The parameters are $\Pi_{times} = \langle\{0^0, s^1\}, \{u^1, q^3\}\rangle$. If $\mathcal{P}_{sum}$ interprets $u(z)$ as $z = 0$ and $q(a,b,c)$ as $c = a + b$, then the $\mathcal{P}_{sum}$-initial model interprets $times(x,y,z)$

[7] It always exists, as shown in [11].

as $z = x * y$. If $\mathcal{P}_{prod}$ interprets $u(z)$ as $z = s(0)$ and $q(a,b,c)$ as $c = a * b$, then the $\mathcal{P}_{prod}$-initial model interprets $times(x,y,z)$ as $z = x^y$.

To introduce isoinitial semantics for parametric programs, we have to consider their open completion $Ocomp(P)$, as defined in the previous section. Here the open symbols are the parameters.

Example 11. The open completion $Ocomp(P_{times})$ contains $Cdef(times)$, i.e.

$$\forall x, v, z \,.\, times(x,v,z) \leftrightarrow (v = 0 \wedge u(z)) \vee$$
$$(\exists y, w \,.\, v = s(y) \wedge times(x,y,w) \wedge q(x,w,z))$$

and $CET(0,s)$.

While P and $Ocomp(P)$ are ini-parametric for every program P, this is no longer true for isoinitial semantics. As for closed programs, iso-parametricity of $Ocomp(P)$ depends on the termination properties of P. More precisely, we need to consider ground existential termination in a Π_P-interpretation $\mathcal{P}$ (formally defined in [11]). Informally, ground existential termination in $\mathcal{P}$ is defined in terms of the SLD-derivations and trees computed by an idealised $\mathcal{P}$-interpreter, that can solve the goals involving the parameters according to their interpretation in $\mathcal{P}$.

Theorem 7. *Let P be a program with parameter signature Π_P and open completion $Ocomp(P)$. $Ocomp(P)$ is iso-parametric in a class of Π_P-interpretations if and only if P ground existentially terminates in every interpretation of the class.*

That is, P ground existentially terminates in a Π_P-interpretation $\mathcal{P}$ if and only if there is a $\mathcal{P}$-isoinitial model of $Ocomp(P)$. The proof requires the results of [11] and the properties of $\mathcal{P}$-isoinitial models. We omit it for conciseness.

Example 12. Consider the open program P_{times}. $Ocomp(P_{times})$ is iso-parametric in the class of interpretations over the Herbrand structure corresponding to $CET(0,s)$. Indeed, in this class ground existential termination can be proved by the fact that $times(x, s(y), z)$ activates a recursive call with $y < s(y)$.

For an example where iso-parametricity fails, consider the following simple program P_p: $\quad p(a) \leftarrow q(a) \qquad p(b) \leftarrow p(b) \qquad (P_p)$

Its parameters are $\Pi_p = \langle \{a^0, b^0\}, \{q^1\} \rangle$. Its open completion $Ocomp(P_p)$ contains the axioms: $\neg a = b$, $\forall x \,.\, p(x) \leftrightarrow (x = a \wedge q(a)) \vee (x = b \wedge p(b))$.
A possible Π_p interpretation is the term interpretation $\mathcal{P}$ with domain $\{a, b\}$ and q true in a and false in b. Every $\mathcal{P}$-model has to interpret $p(a)$ as true, but $p(b)$ can be interpreted as true by some $\mathcal{P}$-models and as false by others. Thus, a $\mathcal{P}$-isoinitial model cannot exist. This is a consequence of the presence of the cyclic clause $p(b) \leftarrow p(b)$.

6.3 Compositionality

Let P be a *sem*-parametric program with parameter signature Π_P, and let $\mathcal{P}$ be a Π_P-interpretation. Let Q be a *sem*-closed program that computes the parameters of P correctly with respect to $\mathcal{P}$, and let us consider the composite program $P \cup Q$. As remarked in Section 5, there is a sufficient completeness problem here. For $P \cup Q$, this problem can be stated as follows: which kind of goals have to be solved by Q, in order that it can replace the idealised $\mathcal{P}$-interpreter considered in the previous section?

For initial semantics the answer is easy, and is given by Theorem 8, while isoinitial semantics can expose *termination* problems in program composition and requires a more complex analysis (see Theorem 9).

Theorem 8. *Let P be a program with signature $\Sigma_P = \langle F_P, R_P \cup O\rangle$ and parameters $\Pi_P = \langle F_P, O\rangle$. Let $\Delta = \langle F, O\rangle$ be a signature such that $F \supseteq F_P$, and $\mathcal{J}$ be a reachable Δ-interpretation. Let $\mathcal{H}_{\mathcal{J}}$ be the set of ground atoms true in $\mathcal{J}$. Then $P \cup \mathcal{H}_{\mathcal{J}}$ is an ini-closed expansion of P with a reachable initial model $\mathcal{I}$. Moreover, $\mathcal{I}$ is isomorphic to the $(\mathcal{J}\,|\,\Pi_P)$-initial model of P.*

Proof. Since P is *ini*-parametric, it has a $\mathcal{J} \mid \Pi_P$-initial model, that we will indicate by $\mathcal{J}_P$. Since every model of $P \cup \mathcal{H}_{\mathcal{J}}$ is a model of P, there is a unique homomorphism $h : \mathcal{J}_P \to \mathcal{I}$. By the initiality of $\mathcal{I}$, there is a unique homomorphism $h' : \mathcal{I} \to \mathcal{J}_P$. Since $F_P \subseteq F$, $\mathcal{I}$ and $\mathcal{J}_P$ are F-reachable. This allows us to show that h' is the inverse map of h, i.e., h is an isomorphism.

Let P, Π_P, Δ, $\mathcal{J}$, $\mathcal{H}_{\mathcal{J}}$ and $\mathcal{J}_P$ as in Theorem 8. As an easy corollary, we can prove that, for a program Q with signature Δ, if the success set of Q coincides with $\mathcal{H}_{\mathcal{J}}$, then $\mathcal{M}(P \cup Q)$ is isomorphic to $\mathcal{J}_P$.

Informally, this means that, to get a program Q such that $P \cup Q$ correctly computes the $(\mathcal{J}\,|\,\Pi_P)$-initial model $\mathcal{J}_P$ of P, it suffices that the success set of Q coincides with the set of ground atoms true in $\mathcal{J}$. This result easily extends to the case where the signature Δ of Q contains the open predicates O of P and other auxiliary predicates not in the signature of P.

For isoinitial semantics of open programs, we have to consider the open completion *Ocomp*(P). In this case, the answer to the sufficient completeness problem is more complex and involves termination, as we now explain.

Let us consider P, Π_P, Δ, $\mathcal{J}$, $\mathcal{H}_{\mathcal{J}}$ and $\mathcal{J}_P$ as in Theorem 8. As explained in the previous section, there is a $\mathcal{J}\,|\,\Pi_P$-isoinitial model of *Ocomp*(P) if and only if P ground existentially terminates in $\mathcal{J}$. We can show that this model is also $\mathcal{J}\,|\,\Pi_P$-initial, i.e., it is $\mathcal{J}_P$.

Now, let Q be an *iso*-closed program with success set $\mathcal{H}_{\mathcal{J}}$. By Theorem 8, $\mathcal{M}(P \cup Q)$ is isomorphic to $\mathcal{J}_P$. However, $\mathcal{M}(P \cup Q)$ is an isoinitial model of *Comp*$(P \cup Q)$ if and only if $P \cup Q$ ground existentially terminates. The ground existential termination of $P \cup Q$ is not guaranteed by the *iso*-closure of Q. This is due to the fact that, in general, Q must terminate also for suitable non-ground goals, to get existential termination of $P \cup Q$. We can link this termination problem to the sufficient completeness problem by Theorem 9 below.

An *goal axiom* is a closed formula of the form $\exists x\,.\,G$ or $\neg\exists x\,.\,G$, where G is a conjunction of atoms and $\exists x$ is a possibly empty sequence of existential quantifiers. If it is empty, we will simply write G. A *goal theory* is a (possibly infinite) set of goal axioms.

We say that a closed program Q *computes* a goal theory T if and only if, for every $\exists x G \in T$, the goal $\leftarrow G$ is successful in Q, and for every $\neg\exists x G \in T$, $\leftarrow G$ is finitely failed in Q. We have the following theorem.

Theorem 9. *Let P be a program with parameter signature $\Pi_P = \langle F_P, O\rangle$, such that $Ocomp(P)$ is iso-parametric. Let $\mathcal{J}$ be a reachable Δ-interpretation, where $\Delta = \langle F, O\rangle$, with $F \subseteq F_P$, and let $\mathcal{J}_P$ be the $\mathcal{J} \mid \Pi_P$-isoinitial model of $Ocomp(P)$. Let Q be a closed program with signature Δ. If there is a goal theory T such that $\mathcal{J} \models T$, Q computes T and $Ocomp(P) \cup T$ is iso-closed, then $Comp(P \cup Q)$ is iso-closed and $\mathcal{J}_P$ is an isoinitial model of it.*

Proof. Since $Ocomp(P) \cup T$ is *iso*-closed and $\mathcal{J}_P$ is reachable, then one can show that $\mathcal{J}_P$ is an isoinitial model of $Ocomp(P) \cup T$. $Comp(Q) \vdash H$ for every $H \in T$, because Q computes T, and $Ocomp(P) \cup Comp(Q) = Comp(P \cup Q)$, because Q is a closed Δ-program. Therefore, the models of $Comp(P \cup Q)$ are a subset of those of $Ocomp(P) \cup T$ and $\mathcal{J}_P$ is an isoinitial model of $Comp(P \cup Q)$.

This theorem shows that there is a link between termination properties of program composition and the minimal *iso*-closures of $Ocomp(P)$. This justifies our choice of *iso*-closures as an interesting semantics for open programs.

Example 13. Consider the open program P_{times} and the axiom $Cdef(times)$ of $Ocomp(P_{times})$ of Example 11. We can prove, for example:

$$\begin{array}{l} times(s0, s0, s0) \leftrightarrow \\ \exists w_1 \,.\, times(s0, 0, w_1) \wedge q(s0, w_1, s0) \leftrightarrow \\ \exists w_1 \,.\, u(w_1) \wedge q(s0, w_1, s0) \end{array}$$

Thus, if a goal theory T proves $\exists w_1 \,.\, u(w_1) \wedge q(s0, w_1, s0)$, then $Ocomp(P_{times}) \cup T \vdash times(s0, s0, s0)$. If $T \vdash \neg\exists w_1 \,.\, u(w_1) \wedge q(s0, w_1, s0)$, then $Ocomp(P_{times}) \cup T \vdash \neg times(s0, s0, s0)$.

In general, T has to decide existential formulas of the form $\exists w_1, \ldots, w_n \,.\, u(w_1) \wedge q(a, w_1, w_2) \wedge \ldots \wedge q(a, w_n, c)$, in order that $Ocomp(P_{times}) \cup T$ be *iso*-closed.

This means that, for a closed program Q with predicates u and q, $P \cup Q$ existentially ground terminates, if and only if Q computes a goal theory T of the above kind.

7 Conclusion

The traditional view of a definite logic program (e.g., [9]) 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, parametricity or compositionality. This view is therefore very much one of *programming-in-the-small*.

Our motivation is to 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 and compositionality in a uniform manner with respect to both closed and open programs. Moreover, constructive formal systems can help to formally prove isoinitiality and treat composition of *iso*-parametric theories [15]. Future work here will include a comparison with other semantics for modularity (e.g. [4, 5]) and compositionality (e.g. [3, 13]). Clearly such a uniform semantics is important if logic programming is to be used for large-scale software development. We have already used isoinitial semantics in our work in formal program development (e.g. [10]).

Finally, a brief comment on definite and normal programs. Under initial semantics, for every definite program D, both D and $Comp(D)$ are *ini-closed*, whereas for some normal programs N, both N and $Comp(N)$ are *ini-open*. Under isoinitial semantics, this asymmetry disappears if we consider $Comp(D)$ and $Comp(N)$ only: both definite and normal programs may be *iso*-open (and non-termination is one of the causes of *iso*-openness). For normal programs, other kinds of semantics have been proposed (for a survey see e.g. [1]). A comparison with these semantics is also one of our next steps. For example, we can link *iso*-closures and stable models by using open completion in a suitable way.

References

1. K.R. Apt and R. Bol. Logic programming and negation: a survey. *JLP* 19,20:9-71, 1994.
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. Bossi, M. Gabbrielli, G. Levi and M.C. Meo. A compositional semantics for logic programs. *TCS* 122:3-47, 1994.
4. A. Brogi, P. Mancarella, D. Pedreschi and F. Turini. Modular logic programming. *ACM TOPLAS* 16(4):1361–1398, 1994.
5. M. Bugliesi, E. Lamma and P. Mello. Modularity in logic programming. *JLP* 19,20:443–502, 1994.
6. D. Chan. Constructive negation based on the completed database. In *Proc. JICSLP'88*, pages 111-125, MIT Press, 1988.
7. D. De Schreye and S. Decorte. Termination of logic programs: The never-ending story. *JLP* 19,20:199-260, 1994.
8. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–385, 1991.
9. 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*:449–503, Oxford University Press, 1993.
10. J. Küster Filipe, K.-K. Lau, M. Ornaghi, and H. Yatsu. On dynamic aspects of OOD frameworks in component-based software development in computational logic. In *Proc. LOPSTR'99, LNCS* 1817:43-62, Springer-Verlag, 2000.
11. K.-K. Lau, M. Ornaghi, and S.-Å. Tärnlund. Steadfast logic programs. *JLP* 38(3):259-294, 1999.
12. J.W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, 2nd edition, 1987.
13. P. Lucio, F. Orejas, and E. Pino. An algebraic framework for the definition of compositional semantics of normal logic programs. *JLP* 40:89-123, 1999.
14. 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.
15. 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.
16. R. Reiter. On closed world data bases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322, Plenum Press, 1978.