From failure to success : comparing a denotational and a declarative semantics for Horn clause logic

Citation for published version (APA):

Document status and date:
Published: 01/01/1990

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 06. Oct. 2019
Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

F.S. de Boer, J.N. Kok, C. Palamidessi, J.J.M.M. Rutten

From failure to success: comparing a denotational and a declarative semantics for Horn clause logic

Computer Science/Department of Software Technology Report CS-R9019 May 1990
From failure to success: comparing a denotational and a declarative semantics for Horn clause logic
The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).
From Failure to Success: Comparing a Denotational and a Declarative Semantics for Horn Clause Logic *

F.S. de Boer
Department of Computer Science, Technical University Eindhoven,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

J.N. Kok
Department of Computer Science, University of Utrecht,
P.O. Box 80089, 3508 TB Utrecht, The Netherlands

C. Palamidessi
Dipartimento di Informatica, Università di Pisa,
Corso Italia, 40, 56125 Pisa, Italy

J.J.M.M. Rutten
Centre for Mathematics and Computer Science,
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Abstract
The main purpose of the paper is to relate different models for Horn Clause Logic: operational, denotational, declarative. We study their relationship by contrasting models based on interleaving, on the one hand, to models based on maximal parallelism, on the other. We make use of complete metric spaces as an important mathematical tool, both in defining and in comparing the various models.

Key words and phrases: operational semantics, denotational semantics, declarative semantics, parallelism, logic languages, correctness, complete metric spaces, partial orders.

1985 Mathematics Subject Classification: 68Q55, 68Q10.

1 Introduction
The most basic example of a (parallel) logic programming language is Horn Clause Logic (HCL). An HCL program is a finite set of definite clauses of the form $H \leftarrow B$, where $H$ is an atom and $B$ is a finite sequence of atoms. We shall introduce three different types of models for HCL: operational, denotational, and declarative. The first and the latter were already introduced elsewhere (see below). In addition to the definition of two denotational models for HCL, the contribution of this paper consists of a systematic comparison of the different models. In particular, we shall establish a precise relationship between the denotational and the declarative models. Although

*Part of this work was carried out in the context of ESPRIT Basic Research Action (3020) Integration. The research of C. Palamidessi was partially supported by the Dutch REX (Research and Education in Concurrent Systems) project.

Report CS-R9019
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands
we have been recently investigating various models for more advanced parallel logic languages like GHC and Parlog [dBKPR89a, dBKPR89b], which contain constructs like the commit operator and annotations for communication, it is necessary to understand the precise relationship of these models first at the basic level of HCL.

1.1 The operational models

We shall consider two operational models, which are both based on a transition system (in the so-called SOS style [HP79]). The first one, called \( \mathcal{O}_\mathbb{FT} \) (FI for fair interleaving), corresponds to the standard (sequential) operational semantics of HCL based on SLD resolution (like in \([vEK76, Llo87]\)); it uses a fair derivation rule (reduction from left to right) in order to model also failure behavior. From \( \mathcal{O}_\mathbb{FT} \) we can deduce the three sets that are classically used to describe the operational behavior of an HCL program: the success set, the finite failure set, and the infinite failure set.

The second operational semantics, \( \mathcal{O}_\mathbb{MP} \), models maximal parallelism; the derivation rule used here amounts to executing in parallel one resolution step for each atom in a goal. (In this way, fairness is automatically ensured.) Then a goal, consisting of several atoms, can do one step by composing all local substitutions of the individual atoms in parallel by means of a parallel composition operator \( \delta \) for substitutions (introduced in \([Pal89]\)). It has two effects: it tests whether these substitutions are mutually compatible and, if so, it takes the union of all the bindings. This model is of interest because it could serve as a basis for a parallel implementation of HCL based languages; furthermore, it can be seen as a starting point for the formalization of additional features such as atomic unification (Cf. \([Sha83]\)). Technically, \( \mathcal{O}_\mathbb{MP} \) will play a role (or, more precisely, the denotational model corresponding to it) as an intermediate in establishing the correspondence between \( \mathcal{O}_\mathbb{FT} \) and the declarative model, to be presented in a minute.

For both operational models, we shall introduce corresponding denotational models. Their main characteristic is compositionality: the meaning of the conjunction of two goals will be computed by composing the meanings of the separate goals. (Note that we do not study compositionality with respect to the union of programs; this we consider to be a separate issue.)

1.2 The denotational models

In order to be compositional, the denotational models are considerably more complicated than their operational counterparts. This is mainly due to the difficulty of describing failure behavior in a compositional manner. The denotational model corresponding to \( \mathcal{O}_\mathbb{FT} \) will be called \( \mathcal{D}_\mathbb{FT} \). In order to allow for the definition of an operator for parallel composition, corresponding to the conjunction of goals, the codomain of this model (also called its semantic universe) will be more complicated than the operational one. Both for \( \mathcal{O}_\mathbb{FT} \) and \( \mathcal{O}_\mathbb{MP} \) it suffices to consider sets of sequences (or words) of substitutions. Here, we need sets of sequences (or vectors) of sequences (or words) of pairs of substitutions. We shall prove the correctness of \( \mathcal{D}_\mathbb{FT} \) with respect to \( \mathcal{O}_\mathbb{FT} \) by showing that the latter equals the composition of an abstraction operation with the former.

Next a second denotational model, called \( \mathcal{D}_\mathbb{MP} \), is introduced, which equals the operational semantics \( \mathcal{O}_\mathbb{MP} \). Its semantic universe is the same as the one of \( \mathcal{O}_\mathbb{MP} \), which is simpler than that of \( \mathcal{D}_\mathbb{FT} \). The semantic operator for the parallel composition (conjunction) of two goals is the operator \( \delta \) described above, but now extended to sets of sequences of substitutions.

1.3 The declarative model

The third type of model we describe is the declarative semantics. We recall the definition of the declarative semantics \( \mathcal{D}_{Dec} \) as introduced in \([FLPM89]\). The term declarative means that the program is seen as a set of first order formulas and that the semantics is intended in the model-theoretic sense, i.e., characterizing the set of logical consequences of the program. This semantics is obtained as the least fixed-point of a continuous transformation \( T \) on the interpretations of the program, the so-called immediate consequence operator. An important distinction between the denotational models above and the declarative semantics is that the latter describes the success
set only, whereas the denotational semantics additionally model (finite and infinite) failure. The first declarative semantics for HCL was proposed by van Emden and Kowalski in [vEK76] (see also [Llo87]). In their approach, interpretations are sets of ground atoms and the least fixed-point, which is equivalent to the least Herbrand model of the program, characterizes the validity of the ground atoms only. The construction in [FLPM89] extends this approach in that interpretations may also contain non-ground atoms. Therefore Dec can also express the validity of so-called generic atoms, i.e., atoms of the form $p(\bar{x})$.

### 1.4 The mathematical tools

We work mainly in the framework of complete metric spaces, in which we follow the tradition initiated by De Bakker and Zucker in [dBZ82]. The metric approach is particularly useful in those situations where (sets of) sets of sequences occur, since these can be supplied with a standard metric. This is the case in the operational and (all but one) denotational models, since they describe in addition to success behavior also (finite and infinite) failure behavior, for which the use of sequences seems natural. The metric structure of our semantic universa is exploited in two ways: first, it enables us to introduce both our models and our semantic operators as the (by Banach’s theorem) unique fixed-points of so-called contractions. Secondly, this uniqueness implies that in order to prove the equality of two models, it is sufficient to show that they are both a fixed-point of the same contraction. It is in particular this second point that distinguishes between the metric and the more usual partial order (or lattice) approach: a continuous operator on a complete partial order has a (least) fixed-point but may have more than one. Therefore it is there more involved to prove such equalities. (Cf. [dV90].) We shall use ordered structures in those cases where we want to describe only success behavior, such as the declarative semantics.

### 1.5 Comparing the models

After having introduced all these models, we shall make a precise and complete comparison. The two operational models are related to the corresponding two denotational models, as just mentioned. The main result of the paper consists in establishing a connection between the first denotational model, $D_{FI}$, and the declarative model Dec. This is done in two steps.

First we shall relate $D_{FI}$ and $D_{MP}$. To this end, an intermediate denotational model $I$ is introduced, to which both are then related. Secondly, and this is the more difficult part, $D_{MP}$ and Dec are compared. Again an intermediate denotational semantics, called $D_{CS}$ (CS for computed substitutions) is introduced. It is essentially a model for maximal parallelism, like $D_{MP}$, but does not deliver sets of sequences of substitutions, but sets of single substitutions only. As a consequence, it only models success behavior. The relationship between $D_{MP}$ and $D_{CS}$ is fairly easy; the only technical problem is that the first model is defined as the fixed-point of a contraction on a complete metric space, whereas the latter is given as the least fixed-point of a continuous function on a complete lattice. Finally, $D_{CS}$ and Dec are related. Although their connection is intuitively obvious, it takes some (technical) effort to make this precise.

At the end of our paper, we mention some consequences that can be deduced from the various relations between the different models. The most important of these is that we can easily establish a proof of the soundness and completeness of the declarative semantics with respect to the success set (which was derived from $D_{FI}$). In this way, we find a fairly transparent alternative to the equivalence proof given in [FLPM89], the latter being quite complicated. The main problem is the contrast between the bottom-up and (maximally) parallel nature of the declarative semantics and the top-down and interleaving nature of the operational semantics. The intermediate models that we have introduced above allow for a decomposition of this proof into several steps, and thus give some insight into the contrasting concepts involved.
2 Mathematical preliminaries

We assume the following notions to be known: complete metric space, continuous function on a metric space, compact subset of a metric space. (The reader might consult, e.g., [Eng77].) We shall also use the following notions from order theory: complete partial order (CPO), complete lattice, continuous function on a CPO.

Let \((M_1, d_1)\) and \((M_2, d_2)\) be two complete metric spaces. A function \(f : M_1 \rightarrow M_2\) is called non-expansive if for all \(x, y \in M_1\)

\[d_2(f(x), f(y)) \leq d_1(x, y)\]

It is called contracting (or a contraction) if there exists \(\epsilon \in [0, 1)\) such that for all \(x, y \in M_1\)

\[d_2(f(x), f(y)) \leq \epsilon \cdot d_1(x, y)\]

Non-expansive and contracting functions are continuous. The following fact is known as Banach's Theorem: let \((M, d)\) be a complete metric space and \(f : M \rightarrow M\) a contraction. Then \(f\) has a unique fixed point, that is, there exists a unique \(x \in M\) such that \(f(x) = x\).

The set \(M_1 \rightarrow M_2\) is the set of all functions from \(M_1\) to \(M_2\). It can be turned into a complete metric space by taking as a metric

\[d(f_1, f_2) = \sup_{x \in M_1} \{d_2(f_1(x), f_2(x))\}\]

(All our metrics will have \([0, 1]\) as their range.) Let

\[\mathcal{P}_{\text{nc}}(M) = \{X : X \subseteq M \land X \text{ is non-empty and compact}\}\]

We can turn \(\mathcal{P}_{\text{nc}}(M)\) into a complete metric space by defining a metric \(d_H\), called the Hausdorff distance induced by \(d\) (the metric on \(M\)), as follows: For every \(X, Y \in \mathcal{P}_{\text{nc}}(M)\)

\[d_H(X, Y) = \max\{\sup_{x \in X} \{d(x, Y)\}, \sup_{y \in Y} \{d(y, X)\}\}\]

where \(d(x, Z) = \inf_{z \in Z} \{d(x, z)\}\) for every \(Z \subseteq M\), \(x \in M\).

We shall often use the following notation: we write \((x, y \in) X\) when introducing a set \(X\) with typical elements \(x\) and \(y\).

A typical example of a complete metric space that we shall often use is the set \((w_1, w_2 \in) A^\infty = A^* \cup A^\omega\) of all finite and infinite words over an alphabet \(A\), supplied with a metric \(d\) given by

\[d(w_1, w_2) = 2^{-\sup \{k : w_1(k) = w_2(k)\}}\]

where \(w(k)\) denotes the prefix of the word \(w\) of length \(k\). We denote the usual concatenation of two words by \(w_1 \cdot w_2\).

3 The language HCL

We only give an informal introduction to the language HCL. For further details we refer to [Llo87, Apt87].

The sets \(\text{Term}\) of terms, \((A, B, H \in) \text{Atom}\) of atomic formulas (or atoms), and \((\varnothing, \sigma, \gamma \in) \text{Subst}\) of substitutions are defined as usual. Elementary atoms \((E\text{Atom})\) are of the form \(p(\overline{x})\), where \(p\) is a predicate and \(\overline{x}\) is a tuple of distinct variables. A definite clause is a construct of the form \(H \leftarrow B_1, \ldots, B_n\) \((n \geq 0)\), where \(H\) and each \(B_i\) is an atom; \(H\) is called the head and \(B_1, \ldots, B_n\) (also denoted by \(\overline{B}\)) the body of this clause. An HCL program \(W\) is a finite set of definite clauses. A goal statement (or goal) is a construct of the form \(-A_1, \ldots, A_n\) \((n \geq 0)\), where each \(A_i\) is an atom. If \(n > 0\) we denote \(-A_1, \ldots, A_n\) also by \(-\overline{A}\). If \(n = 0\) we have the so-called empty goal, and we write \(\emptyset\). The set of all goals is denoted by \(\text{Goal}\).

We have the usual notion of most general unifier of two atoms \(A\) and \(H\), denoted by \(\text{mgu}(A, H)\). For the composition of two substitutions we write \(\varnothing_1 \varnothing_2\). For technical convenience,
we shall throughout this paper consider only idempotent substitutions, i.e., satisfying $\varnothing \varnothing = \varnothing$ (see [Pal89] for some discussion on this point). The set of variables occurring in the atom $A$ is indicated by $\text{Var}(A)$. For an atom $A$ and a substitution $\varnothing$ we write $\varnothing|_A$ for the restriction of $\varnothing$ to $\text{Var}(A)$. The empty substitution is denoted by $\epsilon$.

The classical operational semantics of HCL programs is based on the notion of refutation. Let $G \leftarrow A_1, \ldots, A_n$ be a goal and let $H \leftarrow B_1, \ldots, B_m$ be a (properly renamed variant of a) clause in the program $W$. Assume that $A_i$ and $H$ are unifiable with most general unifier $\varnothing$. Then the goal

$$
\leftarrow (A_1, \ldots, A_{i-1}, B_1, \ldots, B_m, A_{i+1}, \ldots, A_n)\varnothing
$$

is derivable from $G$ by one resolution step. A repeated application of such a resolution step is called a derivation. A derivation is successful (and called a refutation) if it ends with the empty goal $\varnothing$; it is failing if no further reductions are possible while the empty goal has not been reached; and it is infinite otherwise. A selection rule is a function that gives for each goal the atom to be reduced. A derivation according to a certain selection rule is called an SLD-derivation. A selection rule is fair if and only if all the atoms in all the possible goals generated in SLD-derivations are eventually selected. Classically, the (operational) semantics of an HCL program is defined by three sets:

- the success set ($\text{OSS}$), containing all the atoms that have a refutation, instantiated by the last substitution (the so-called computed answer substitution).
- the finite failure set ($\text{OFFS}$), containing all the atoms for which all the fair SLD-derivations are failing (see [AvE82]).
- The infinite failure set ($\text{OIFS}$), containing all atoms, for which there are no successful derivations and there is at least one fair infinite derivation.

The notion of success set given above is not completely satisfactory for characterizing the operational behavior of a logic program. In the present paper, we use a different notion of success set: we take the one introduced in [FLPM88, FLPM89], which contains all the elementary atoms that have a refutation, instantiated by the computed answer substitution (see the next section and that on the declarative semantics).

4 Operational semantics

We present two operational semantics for HCL, which will both be based on a labelled transition system (in the style of [HP79]). The first one models interleaving and uses a breath-first selection rule, which is fair. The second operational semantics, in which all the atoms occurring in a goal are reduced at the same time, describes maximal parallelism. Throughout the rest of this paper, we assume the program $W$ to be fixed.

4.1 Interleaving

First we introduce a labelled transition system for fair interleaving, on which our first operational semantics will be based.

**Definition 4.1** Let $(\text{Goal}, \text{Subst}, \rightarrow)$ be the labelled transition system, of which the transition relation $\rightarrow \subseteq \text{Goal} \times \text{Subst} \times \text{Goal}$ is defined as the smallest relation satisfying the following axiom:

$$
A, \overline{A} \xrightarrow{\varnothing} \overline{A}\varnothing, \overline{B}\varnothing
$$

(As usual, we write $\leftarrow \overline{A} \xrightarrow{\varnothing} \overline{B}$ rather than $(\leftarrow \overline{A}, \varnothing, \leftarrow \overline{B}) \in \rightarrow$.) Here $\varnothing = \text{mgw}(A, H)$ and $H \leftarrow \overline{B}$ is a clause of $W$. We assume this clause to be renamed such that $A, \overline{A}$ and $H$ have no variables in common.
Note that in the above axiom, a breath-first selection rule is used. In this way, fairness is automatically ensured. This left-to-right strategy does not impose any restrictions; we still get all possible fair behaviors. This can be proved by making use of the so-called switching (or square) lemma (see [Llo87]). Another feature of the above transition system is the fact that the computed substitution (above the arrow) is applied to the goal at the right of the arrow. This ensures that all subsequently computed substitutions will be consistent with (i.e., extensions of) the current one.

Based on this transition system we define an operational semantics \( \ast Sem : \text{Goal} \rightarrow \mathcal{P}_{\text{ST}}, \) which associates with a goal a set of sequences of substitutions. The semantic universe \((X, Y) \in \mathcal{P}_{\text{ST}} \) (ST is an abbreviation for streams) is given by

\[
\mathcal{P}_{\text{ST}} = \mathcal{P}_{\text{seco}}(\text{Subst}_5^\varepsilon)
\]

where \( \text{Subst}_5^\varepsilon \), the set of finite, infinite and deadlocking sequences (or words, or streams), is defined by

\[
(v, w, z \in \text{Subst}_5^\varepsilon) \quad \text{Subst}_5^\varepsilon = \text{Subst}^+ \cup \text{Subst}^\omega \cup \text{Subst}^* \cdot \delta
\]

As a metric on \( \mathcal{P}_{\text{ST}} \) we take the Hausdorff metric induced by the standard metric on sequences (see the preliminaries). The empty sequence is denoted by \( \lambda \) and the concatenation of two sequences \( w_1 \) and \( w_2 \) by \( w_1 \text{ · } w_2 \). To denote failure we have added to the set of substitutions a special element \( \delta \). We postulate for any substitution \( \theta \) that \( \delta \theta \), the composition of \( \theta \) and \( \delta \), equals \( \delta \); for any sequence of substitutions \( v \) we have that \( \delta \cdot v \), the concatenation of \( \delta \) and \( v \), is equal to \( \delta \). Each sequence represents a particular computation that corresponds to a specific choice of clauses. The elements of such a sequence represent the partial results of the computation. Finite sequences not ending in \( \delta \) (elements of \( \text{Subst}^+ \)) correspond to successfully terminating computations (refutations). Sequences ending in \( \delta \) (in \( \text{Subst}^* \cdot \delta \)) represent failing computations. Infinite sequences (in \( \text{Subst}^\omega \)) are associated with infinitely failing computations.

Definition 4.2 Let \( \Phi_{\text{FI}} \) be the unique fixed point of the contracting operator \( \Phi_{\text{FI}} : (\text{Goal} \rightarrow \mathcal{P}_{\text{ST}}) \rightarrow (\text{Goal} \rightarrow \mathcal{P}_{\text{ST}}) \), which is given by

\[
\Phi_{\text{FI}}(F)[\square] = \{ \varepsilon \}
\]

\[
\Phi_{\text{FI}}(F)[\leftarrow \vec{A}] = \cup \{ \theta \cdot (\theta \leftarrow \leftarrow \vec{F}[\vec{A}]') : \leftarrow \vec{A} \leftarrow \leftarrow \vec{A}' \} \cup \{ \delta : \leftarrow \vec{A} \leftarrow \}
\]

Here \( \leftarrow \leftarrow \vec{F} \) : \( \text{Subst}^+ \times \mathcal{P}_{\text{ST}} \rightarrow \mathcal{P}_{\text{ST}} \) is defined by \( \theta \leftarrow \leftarrow \vec{F} X = \{ \theta \leftarrow \leftarrow \vec{F} x : x \in X \} \), with

\[
\theta \leftarrow \leftarrow \vec{F} \lambda = \lambda
\]

\[
\theta \leftarrow \leftarrow \vec{F} (\sigma \cdot z) = (\theta \sigma) \cdot (\theta \leftarrow \leftarrow \vec{F} z)
\]

The contractivity of \( \Phi_{\text{FI}} \) in the above definition is straightforward. The compactness of \( \Phi_{\text{FI}}(F)[\leftarrow \vec{A}] \) follows from the fact that only finitely many transitions are possible from \( \leftarrow \vec{A} \).

The definition of \( \Phi_{\text{FI}}(\square) \) is obvious. For a non-empty goal \( \leftarrow \vec{A} \) we have that \( \Phi_{\text{FI}}(\leftarrow \vec{A}] \) equals \{ \delta \} if there are no transitions possible from \( \leftarrow \vec{A} \) (indicated by \( \leftarrow \vec{A} \not\leftarrow \)). Otherwise, \( \Phi_{\text{FI}}(\leftarrow \vec{A}] \) contains all sequences that start with \( \theta \) and continue with a sequence stemming from \( \Phi_{\text{FI}}(\leftarrow \vec{A}] \), in which every element is composed with \( \theta \). The latter is caused by the application of \( \theta \leftarrow \leftarrow \vec{F} \) to \( \Phi_{\text{FI}}(\leftarrow \vec{A}] \), which is added because we want to collect the total effect of all intermediate substitutions.

The definition has been presented in a fixed-point format, because this will ease the comparison of \( \Phi_{\text{FI}} \) with other models still to come. We could, however, have given a more direct definition based on transition sequences. A second remark concerns the use of the somewhat abstract operation \( \theta \leftarrow \leftarrow \vec{F} \). This could have been avoided as well by using a different type of transition system, in which a configuration \( \leftarrow \vec{A}, \sigma \) would consist of both a goal and a substitution. The latter could then be used to store all the bindings found sofar. The axiom corresponding to the one above would be

\[
\langle \leftarrow \vec{A}, \sigma \rangle \rightarrow \langle \leftarrow \vec{A}, \vec{B}, \sigma \theta \rangle
\]
with $B$ and $\vartheta$ as above.

The following counter example shows that $\mathcal{OFI}$ is not compositional. Consider the following program

\begin{align*}
{p(x) \leftarrow s_1(x), p(x) \leftarrow s_2(x), q(x) \leftarrow s_3(x)} \\
{\begin{cases} 
\delta_1(a) \leftarrow, \delta_2(b) \leftarrow, \delta_3(c) \leftarrow, \tau(a) \leftarrow 
\end{cases}}
\end{align*}

It is easy to see that with respect to this program $\mathcal{OFI}[\neg p(x)] = \mathcal{OFI}[\neg q(x)]$. But, on the other hand, we have $\{x/a\} \in \mathcal{OFI}[\neg \tau(x), p(x)] \setminus \mathcal{OFI}[\neg \tau(x), q(x)]$.

### 4.2 Success, finite failure and infinite failure sets

From the operational semantics $\mathcal{OFI}$ we can derive the success set, the finite failure set, and the infinite failure set in the following way:

\begin{align*}
\mathcal{OSS} &= \{p(x)\vartheta : p(x) \in E\text{Atom} \land \vartheta \in \text{last}(\mathcal{OFI}[\neg p(x)] \cap \text{Subst}^+)\} \\
\mathcal{OFPS} &= \{A : \mathcal{OFI}[\neg A] \subseteq \text{Subst}^* \cdot \delta\} \\
\mathcal{OIFS} &= \{A : \mathcal{OFI}[\neg A] \cap \text{Subst}^* = \emptyset \land \mathcal{OFI}[\neg A] \cup \text{Subst}^\omega \neq \emptyset\}
\end{align*}

In the first set, the function $\text{last}$ takes from a set of sequences the last elements (not equal to $\delta$). Those elements represent the computed answer substitution for successful refutations. The notion of success set we consider here is introduced in [FLPM89, FLPM88] and extends the standard one given in [Llo87, vEK76]. (See also the section on the declarative semantics for some more discussion.) The second set, $\mathcal{OFPS}$, contains those atoms that give rise to only failing computations, i.e., sequences of substitutions that end in $\delta$. The last set, $\mathcal{OIFS}$, contains the so-called infinitely failing atoms; those give rise to no successful computations and at least one fair infinite one.

### 4.3 Maximal parallelism

The next execution model we consider for our language is called maximally parallel. Each step in the execution of a goal consists conceptually of two stages: first, all atoms present in the goal perform one step independently. Secondly, the substitutions resulting from these local computations are composed in order to obtain the global outcome of the computation. For this composition we introduce a new operator on substitutions called parallel composition. (Sometimes it is called reconciliation operator; Cf. [Jac89].) It is defined as follows.

**Definition 4.3** We define the parallel composition of two substitutions $\vartheta$ and $\sigma$, denoted by $\vartheta \circ \sigma$, by

$$
\vartheta \circ \sigma = \begin{cases} 
\text{mgu}(S(\vartheta) \cup S(\sigma)) & \text{if it exists} \\
\delta & \text{otherwise}
\end{cases}
$$

where $S(\vartheta) = \{< x, t > : x/t \in \vartheta \}$. Furthermore we define $\vartheta \circ \delta = \delta \circ \vartheta = \delta$. (Note that the notion of $\text{mgu}$ is extended to sets of pairs of terms.)

This operator tests whether the two substitutions are compatible and, if this is the case, yields the minimal substitution containing the same information (bindings) as these substitutions. Otherwise it yields $\delta$. It is straightforward to show that $\circ$ is commutative, associative, and idempotent (modulo the renaming of variables).

The proof of the correspondence of the interleaving and the maximally parallel semantics will make use of the following property of this operator.

**Lemma 4.4** For all substitutions $\vartheta_1$ and $\vartheta_2$

$$
\vartheta_1 \circ \vartheta_2 = \vartheta_1 \text{mgu}(S(\vartheta_2)\vartheta_1)
$$

where $S(\vartheta_2)\vartheta_1 = \{< x\vartheta_1, t\vartheta_1 > : x, t \in S(\vartheta_2)\}$. 

7
For the proof of this lemma and additional discussion of \( \hat{o} \) we refer to [Pal89]. The definition of \( \hat{o} \) is illustrated by the following example.

**Example 4.5** Let \( \vartheta_1 = \{x/f(y,a), z/g(b)\} \) and \( \vartheta_2 = \{x/f(b,w), z/g(y)\} \). Then

\[
\vartheta_1 \hat{o} \vartheta_2 = \text{mgu}\{<x,f(y,a)>, <z,g(b)>, <x,f(b,w)>, <z,g(y)>\} = \{x/f(b,a), z/g(b), y/b, w/a\}
\]

If we take \( \vartheta_1 \) as before and \( \vartheta_2 = \{x/f(a,w), z/g(y)\} \) we have

\[
\vartheta_1 \hat{o} \vartheta_2 = \text{mgu}\{<x,f(y,a)>, <z,g(b)>, <x,f(a,w)>, <z,g(y)>\} = \hat{o}
\]

Next we introduce a transition relation for maximal parallelism. It is specified by the following axiom and rule.

**Definition 4.6** We define

1. \[ \leftarrow A \xrightarrow{\theta} \leftarrow B \]
   where \( \theta = \text{mgu}(A, H) \) and \( H \leftarrow B \) is an (appropriately renamed) clause of \( W \).

2. \[
\leftarrow \bar{A} \xrightarrow{\theta} \leftarrow \bar{A}', \quad \leftarrow \bar{B} \xrightarrow{\sigma} \leftarrow \bar{B}'
\]
   \[
\leftarrow \bar{A}, \bar{B} \xrightarrow{\theta \circ \sigma} \leftarrow \bar{A}', \bar{B}'
\]

Note that in the conclusion of the rule above, we can have that \( \theta \circ \sigma \) equals \( \hat{o} \). This means that the two substitutions are not compatible.

**Definition 4.7** The operational semantics \( \text{OMP} \) is defined as the fixed point of the contraction \( \Phi_{\text{MP}} : \text{Goal} \rightarrow \text{PsT} \rightarrow \text{Goal} \rightarrow \text{PsT} \), given by

\[
\Phi_{\text{MP}}(F)[\square] = \{\}
\]

\[
\Phi_{\text{MP}}(F)[\leftarrow \bar{A}] = \cup\{\theta : (\theta \sim_{\text{MP}} F[\leftarrow \bar{A}]) : \leftarrow \bar{A} \xrightarrow{\theta} \bar{A}' \} \cup \{\delta : \leftarrow \bar{A} \not\rightarrow\}
\]

Here \( \sim_{\text{MP}} : \text{Subst} \times \text{PsT} \rightarrow \text{PsT} \) is defined by \( \theta \sim_{\text{MP}} X = \{\theta \sim_{\text{MP}} x : x \in X\} \), with

\[
\theta \sim_{\text{MP}} \lambda = \lambda
\]

\[
\theta \sim_{\text{MP}} \sigma \cdot z = \begin{cases} 
\hat{o} & \text{if } \theta \circ \sigma = \hat{o} \\
\theta \circ \sigma \cdot (\theta \sim_{\text{MP}} z) & \text{otherwise}
\end{cases}
\]

The definition of \( \text{OMP} \) is very similar to that of \( \text{OFI} \). Two differences should be noticed here. First, the transition relation that is used is different from the one in the definition of \( \text{OFI} \); secondly, the definition of the function \( \sim_{\text{MP}} \) differs from the function \( \sim_{\text{FI}} \). It composes \( \theta \) in parallel with the elements of \( \text{OMP}[\leftarrow \bar{A}] \), as opposed to \( \theta \sim_{\text{FI}} \), which uses ordinary composition. Here we use the parallel composition, because in the transition system above, the substitution above the arrow is not applied to the atom at its right-hand side. Therefore, the next computation step will not take this substitution into account and the next substitution that is computed has to be reconciled with the previous one.
5 Denotational semantics for interleaving

In this section, we develop a denotational semantics $D_{FI}$ for the operational interleaving semantics $C_{FI}$. We start by introducing the complete metric space $P_{FI}$, which is defined by

$$P_{FI} = \mathcal{P}_{\text{acc}}(((\text{Subst} \times \text{Subst})^+)^\infty)$$

with a metric on $P_{FI}$ similar to the one on $P_{ST}$. It consists of sets of (finite and infinite) sequences of finite sequences of pairs of substitutions. Such a sequence (called a vector) we denote by $<v_1, \ldots, v_n, \ldots>$, where each $v_i$ is a finite sequence of pairs of substitutions. We shall use the following prefixing operator, which composes a vector containing one pair of substitutions, $<(\vartheta_1, \vartheta_2)>$ and a vector $<v_1, v_2, \ldots>$, and is defined by

$$<(\vartheta_1, \vartheta_2) \cdot v_1 \cdot v_2, \ldots> = <(\vartheta_1, \vartheta_2), v_1, v_2, \ldots>$$

We use pairs of substitutions to represent the basic (unification) steps in the computation. The first substitution of a pair is called the input substitution and can be seen as an assumption on the behavior of the environment or, in other words, the computation that has taken place so far. The second one, called the output substitution, denotes the result of this computation step. As we shall see below, it will be the substitution resulting from a unification. Failure of such a unification is denoted by $\delta$. (An alternative would have been to use functions from substitutions to substitutions. This would yield a semantics that is less abstract, i.e., more discriminating.)

Next we explain why we use vectors (instead of just sequences of pairs of substitutions). When we define a compositional semantics we introduce a semantic merge operator $||_{FI}$. Operationally, a goal is executed by performing from left to right one step of each atom in the goal. The operator $||_{FI}$ is defined such that it mimics this strategy. If we had sequences of pairs of substitutions in our basic domain we would not be able to do this: we would not know how many processes (atoms) contributed to this goal. Vectors have this kind of information. The intuition is that the $n$-th element of a vector represents the $n$-th left to right swap of the goal. Hence the operator $||_{FI}$ combines two vectors by concatenating their elements, i.e., their sequences of substitutions, component-wise.

**Definition 5.1** We define $||_{FI} : P_{FI} \times P_{FI} \rightarrow P_{FI}$, for every $X, Y \in P_{FI}$, by

$$X ||_{FI} Y = \bigcup \{x ||_{FI} y : x \in X, y \in Y \}$$

where

$$<v_1, v_2, \ldots> ||_{FI} <w_1, w_2, \ldots> = <v_1 \cdot w_1, v_2 \cdot w_2, \ldots>$$

$$<v_1, \ldots, v_n> ||_{FI} <w_1, w_2, \ldots> = <v_1 \cdot w_1, \ldots, v_n \cdot w_n, v_{n+1}, \ldots>$$

Now we are ready to give the definition of the denotational semantics $D_{FI}$.

**Definition 5.2** We define $D_{FI} : \text{Goal} \rightarrow P_{FI}$:

$$D_{FI}[\square] = \{\lambda\}$$

$$D_{FI}[\leftarrow A] = \{<(\vartheta, \text{mgu}(A\vartheta, H))> \cdot D_{FI}[\leftarrow B] : \vartheta \in \text{Subst}, H \leftarrow B \in W\} \cup \{<(\vartheta, \delta)> \cdot \forall H \leftarrow B \in W[\text{mgu}(A\vartheta, H) \text{ does not exist}]\}$$

$$D_{FI}[\leftarrow A_1, A_2] = D_{FI}[\leftarrow A_1] ||_{FI} D_{FI}[\leftarrow A_2]$$

This recursive definition can be justified with the use of contractions in the standard way. (See Definition 6.2 for an example.)

In section 8, the correctness of $D_{FI}$ with respect to $C_{FI}$ will be proved.
6 Denotational semantics for maximal parallelism

We next introduce a denotational variant, named $\mathcal{D}_{MP}$, of the operational model $\mathcal{O}_{MP}$ for maximal parallelism. Unlike the case of fair interleaving, we need not introduce a new semantic universe; we can again take $P_{ST}$. Recall that $P_{ST}$ is defined as

$$P_{ST} = \mathcal{P}_{\text{seq}}(\text{Subst}^s)$$

Before we introduce the model $\mathcal{D}_{MP}$, we first extend the parallel composition operator $\circ$ to a parallel operator $||_{MP}$ defined on sets of sequences of substitutions.

**Definition 6.1** We define $||_{MP}: P_{ST} \times P_{ST} \rightarrow P_{ST}$ by, for all $X$ and $Y$ in $P_{ST}$,

$$X ||_{MP} Y = \bigcup \{ x \mid_{MP} y : x \in X, y \in Y \}$$

Here $x \mid_{MP} y$ is defined by the following cases.

$$\begin{align*}
(\sigma_1 \cdot z_1) \mid_{MP} (\sigma_2 \cdot z_2) &= \begin{cases} 
\delta & \text{if } \sigma_1 \circ \sigma_2 = \delta \\
(\sigma_1 \circ \sigma_2) \cdot (z_1 \mid_{MP} z_2) & \text{otherwise}
\end{cases} \\
\sigma_1 \mid_{MP} (\sigma_2 \cdot z) = (\sigma_2 \cdot z) \mid_{MP} \sigma_1 &= \begin{cases} 
\delta & \text{if } \sigma_1 \circ \sigma_2 = \delta \\
(\sigma_1 \circ \sigma_2) \cdot (\sigma_1 \mid_{MP} z) & \text{otherwise}
\end{cases}
\end{align*}$$

Note that $||_{MP}$ is recursively defined. Formally, we can introduce it as the unique fixed point of a suitably defined contraction.

Now we can introduce the semantics $\mathcal{D}_{MP}$. It turns out to be equal to $\mathcal{O}_{MP}$, which will be proved in section 8.

**Definition 6.2** Let the function $\Psi_{MP}: \text{Goal} \rightarrow P_{ST}$ be the unique fixed point of the contraction $\Psi_{MP}: (\text{Goal} \rightarrow P_{ST}) \rightarrow (\text{Goal} \rightarrow P_{ST})$, given by

$$\begin{align*}
\Psi_{MP}(F)[\Box] &= \{ e \} \\
\Psi_{MP}(F)[\leftarrow A] &= \bigcup \{ \text{mgu}(A,H) \rightarrow_{MP} F[\leftarrow \bar{B}] : H \leftarrow \bar{B} \in W \} \cup \\
& \{ \delta : \forall H \leftarrow \bar{B} \in W \text{ mgu}(A,H) \text{ does not exist} \} \\
\Psi_{MP}(F)[\leftarrow \bar{A}_1, \bar{A}_2] &= \Psi_{MP}(F)[\leftarrow \bar{A}_1] \mid_{MP} \Psi_{MP}(F)[\leftarrow \bar{A}_2]
\end{align*}$$

It is not difficult to show that $\Psi_{MP}$ in the above definition is contracting; a proof would make use of the fact that $||_{MP}$ is non-expansive, an observation that on its turn is rather straightforward.

7 Declarative semantics

In this section, we recall the definition of the declarative semantics $\mathcal{D}_{Dec}$ introduced in [FLPM89].

The term *declarative* means that the program is seen as a set of first order formulas and that the semantics is intended in the model-theoretic sense, i.e., characterizing the set of logical consequences of the program. This semantics is obtained as the least fixed-point of a continuous transformation $T$ on the interpretations of the program. Such a transformation is called *immediate consequence operator* because for an interpretation $I$, the set $T(I)$ contains all the (atomic) consequences obtained from the (atomic) formulas that are true in $I$ by a one step inference from the program. The first declarative semantics for HCL was proposed by van Emde and Kowalski in [vEK76]. In their approach, interpretations are sets of ground atoms and the least fixed-point, shown equivalent to the least Herbrand model of the program, characterizes the validity of the ground atoms only. The construction in [FLPM89] extends this approach in that interpretations contain also non ground atoms and therefore the least fixed-point allows to express validity for so-called generic atoms.

Next we give the construction of [FLPM89] in more detail. We refer to that paper for the proofs of the results we mention here. For Theorem 7.7 a proof will be presented in section 9.
Definition 7.1 The partially ordered set of (extended) interpretations, with typical element I, is defined as $(P_{\text{Dec}}, \preceq)$, where $P_{\text{Dec}} = \mathcal{P}(\text{Atom})$.

Proposition 7.2 $(P_{\text{Dec}}, \preceq)$ is a complete lattice.

Definition 7.3 The (extended) immediate consequence operator $T : P_{\text{Dec}} \rightarrow P_{\text{Dec}}$ is defined by

\[ T(I) = \{ \text{Hmgu}(\bar{B}, \bar{B}') : H \leftarrow \bar{B} \in W, \bar{B}' \in I \} \]

Proposition 7.4 The operator $T$ is continuous.

Since $T$ is continuous, its least fixed-point $lfp(T)$ exists; moreover, $lfp(T) = \bigcup_{n \geq 0} T^n(\emptyset)$, where $T^n(I)$ is defined by

\[ T^0(I) = I, \quad T^{n+1}(I) = T(T^n(I)) \]

The declarative semantics is defined as follows.

Definition 7.5 $\text{Dec} = lfp(T)$

The next theorem gives the relation between the model-theoretic semantics of $W$ and $\text{Dec}$.

Theorem 7.6 For every atom $A$,

\[ W \models A \ (\text{i.e., } A \text{ is a logical consequence of } W) \text{ if and only if} \]

\[ \exists A' \in \text{Dec} \exists \theta \in \text{Subst} [A' \theta = A]. \]

Finally, the following result expresses the relation between $\text{Dec}$ and the success set.

Theorem 7.7 $\text{Dec} = \text{Oss}$

8 The relations between the models

8.1 The relations between the denotational and the operational models

8.1.1 Relating $\text{OP}_1$ and $\text{DF}_1$

We start with the relation between $\text{OP}_1$ and $\text{DF}_1$, the operational and denotational semantics based on interleaving. They will be connected by the following abstraction operator.

Definition 8.1 The operator $\beta_{\text{FI}} : \text{Subst}_\epsilon \rightarrow P_{\text{FI}} \rightarrow P_{\text{FI}}$ is defined by $\beta_{\text{FI}}(\delta)(X) = \{ \lambda \}$, and for $\theta \neq \delta$, by

\[ \beta_{\text{FI}}(\theta)(X) = \bigcup \{ \vartheta_1 : \beta_{\text{FI}}(\vartheta_1)(X(\vartheta, \vartheta_1)) : X(\vartheta, \vartheta_1) \neq \emptyset \} \]

Here $X(\vartheta, \vartheta_1)$ is defined by $X(\vartheta, \vartheta_1) = \{ \langle v_1, v_2, \ldots \rangle : (\vartheta, \vartheta_1) \cdot v_1, v_2, \ldots \in X \}$.

(The well-definedness of $\beta_{\text{FI}}$ can be established in the familiar way: it can be given as the fixed-point of a contraction.) The abstraction operator $\beta_{\text{FI}}$ first selects from the set $X$ the connected sequences, that is, those sequences such that the output substitution of a pair equals the input substitution of the following pair. From such a connected sequence it takes all the output substitutions.

We have the following theorem relating $\text{OP}_1$ and $\text{DF}_1$. (Recall that $\epsilon$ is the empty substitution.)

Theorem 8.2 For every goal $\leftarrow \bar{A}$ we have $\beta_{\text{FI}}(\epsilon) \circ D_{\text{FI}}[\leftarrow \bar{A}] = O_{\text{FI}}[\leftarrow \bar{A}]$. 

11
Proof We prove \( \beta_{\mathcal{P}l}(e) \circ \mathcal{D}_{\mathcal{P}l} = \mathcal{C}_{\mathcal{P}l} \) by showing that \( \beta_{\mathcal{P}l}(e) \circ \mathcal{D}_{\mathcal{P}l} \) is a fixed-point of the contraction \( \Phi_{\mathcal{P}l} \). Then the equality follows from Banach’s theorem. We omit the deadlock case, which can be taken care of straightforwardly.

\( \Phi_{\mathcal{P}l}(\beta_{\mathcal{P}l}(e) \circ \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow A, \vec{A} \right] ) = \)

\[
\bigcup \{ \vartheta : (\vartheta \circ_{\mathcal{P}l} (\beta_{\mathcal{P}l}(e) \circ \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow A \vartheta, \vec{B} \vartheta ])) : H \leftarrow \vec{B} \in W \text{ and } \vartheta = \text{mgu}(A, H) \} =
\]

(remark 1)

\[
\bigcup \{ \vartheta : (\beta_{\mathcal{P}l}(\vartheta) \circ \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow A, \vec{B} \right] ) : H \leftarrow \vec{B} \in W \text{ and } \vartheta = \text{mgu}(A, H) \} =
\]

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (\beta_{\mathcal{P}l}(\vartheta) \circ \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow A, \vec{B} \right] ) : H \leftarrow \vec{B} \in W \text{ and } \vartheta = \text{mgu}(A, H) \} =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (\beta_{\mathcal{P}l}(\vartheta) \circ \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow A \right] ) : H \leftarrow \vec{B} \in W \text{ and } \vartheta = \text{mgu}(A, H) \} =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (\beta_{\mathcal{P}l}(\vartheta) \circ \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow \vec{A} \right] ) : H \leftarrow \vec{B} \in W \text{ and } \vartheta = \text{mgu}(A, H) \} =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (\beta_{\mathcal{P}l}(\vartheta) \circ \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow A, \vec{B} \right] ) : H \leftarrow \vec{B} \in W \text{ and } \vartheta = \text{mgu}(A, H) \} =
\)

Remark 1 The identity \( \beta_{\mathcal{P}l}(e) \circ \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow A \vartheta, \vec{B} \vartheta ] = \beta_{\mathcal{P}l}(\vartheta) \circ \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow \vec{A}, \vec{B} \right] \) is justified by the following observations. Let \( < v_1, \ldots > \in \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow \vec{A}, \vec{B} \right] \) be a connected sequence with its first pair of the form \( (\vartheta, \vartheta') \), for some \( \vartheta' \). It follows that \( v_1 = < (\vartheta, \vartheta_1), \ldots, (\vartheta_1 \ldots \vartheta_{n-1}, \vartheta_1 \ldots \vartheta_n) > \), with \( \vartheta_i = \text{mg}(A_i \vartheta \ldots \vartheta_i, H_i) \), for some \( H_i \leftarrow \vec{B}_i \). Here we have \( A_1, \ldots A_n = \vec{A}, \vec{B} \). So for \( v'_1 = < (\vartheta, \vartheta_1), \ldots, (\vartheta_1 \ldots \vartheta_{n-1}, \vartheta_1 \ldots \vartheta_n) > \) there exists a sequence \( < v'_1, \ldots > \in \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow \vec{A} \vartheta, \vec{B} \vartheta ] \). Now each pair occurring in \( < v_2, \ldots > \in \mathcal{D}_{\mathcal{P}l} \left[ \leftarrow \vec{B}_1, \ldots, \vec{B}_n \right] \) is of the form \( (\vartheta \vartheta', \vartheta \vartheta'' \vartheta') \), where \( \vartheta'' = \text{mg}(B \vartheta \vartheta', H) \), for some atoms \( B \) and \( H \). But due to the renaming mechanism, which we implicitly assume, we have that \( \vartheta \) does not affect the variables of \( B \). So we have that \( \vartheta' = \text{mg}(B \vartheta' \vartheta', H) \) implying that we can eliminate \( \vartheta \) from the sequence \( < v_2, \ldots > \). This argument could be formalized by the introduction of an explicit renaming mechanism.

Remark 2 We show that \( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] = \beta_{\mathcal{P}l}(\vartheta) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] \) \) (For convenience, we write \( < v_n >_n \) for \( < v_1, v_2, \ldots > \).

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] \) =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] \) =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] \) =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] \) =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] \) =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] \) =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] \) =
\)

\( \beta_{\mathcal{P}l}(e) \bigcup \{ \vartheta : (X \left[ \leftarrow \mathcal{F}_l Y] \) =
\)

12
8.1.2 Relating $O_{MP}$ and $D_{MP}$

Next we prove the identity of the operational model $O_{MP}$ and the denotational model $D_{MP}$ for maximal parallelism.

**Theorem 8.3** $O_{MP} = D_{MP}$

**Proof** Similarly to the proof of Theorem 8.2, it can be shown that $D_{MP}$ is a fixed-point of the contraction $\Phi_{MP}$, from which the theorem follows.

8.2 Relating $D_{FI}$ and $D_{MP}$

In order to relate $D_{FI}$ and $D_{MP}$, we introduce an intermediate semantics $I : Goal \rightarrow P_{1}$, with $P_{1} = P_{hoo}((Subst_{s}^{+})^{\infty})$, as the fixed-point of the contraction $\Psi : (Goal \rightarrow P_{1}) \rightarrow (Goal \rightarrow P_{1})$ defined as follows.

**Definition 8.4** We define

\[
\Psi(F)[[\square]] = \{\epsilon\}
\]
\[
\Psi(F)[[\leftarrow A]] = \begin{cases} \text{\bigcup} \{< \text{mgu}(A, H) \times F(\tilde{B}) : H \leftarrow \tilde{B} \in W\} & \\
\text{\bigcup} \{\delta : \forall H \leftarrow \tilde{B} \in W \text{mgu}(A, H) \text{ does not exist}\}
\end{cases}
\]
\[
\Psi(F)[[\leftarrow \tilde{A}_{1}, \tilde{A}_{2}]] = \Psi(F)[[\leftarrow \tilde{A}_{1}]] \cup \Psi(F)[[\leftarrow \tilde{A}_{2}]]
\]

Here $\|$ is defined in a similar way as $\|_{FI}$.

Now $D_{FI}$ and $I$ are related by the following abstraction operator.

**Definition 8.5** We define $\alpha : P_{FI} \rightarrow P_{1}$ by

\[
\alpha(X) = \{< \vartheta_{1} \cdots \vartheta_{k}, \vartheta_{k+1} \cdots \vartheta_{n}> : (\epsilon, \vartheta_{1}) \cdots (\epsilon, \vartheta_{k}) \cdots (\epsilon, \vartheta_{n}) \cdots (\epsilon, \vartheta_{n}) \in X\}
\]

(We have omitted the case that $X$ contains finite sequences.)

This abstraction operator selects from each set those sequences that make no assumptions on the environment, i.e., of which all pairs have $\epsilon$ (the empty substitution) as the first element.

**Theorem 8.6** $I = \alpha \circ D_{FI}$

**Proof** It can be shown that $\alpha \circ D_{FI}$ is a fixed-point of $\Psi$.

We continue the equivalence proof of $D_{FI}$ and $D_{MP}$ by relating $I$ and $D_{MP}$. For this purpose we again need an abstraction operator.

**Definition 8.7** We define $\alpha_{MP} : P_{1} \rightarrow P_{ST}$ by $\alpha_{MP}(< s_{1}, s_{2}, \ldots >) = (\delta s_{1}) \cdots (\delta (s_{1} \cdots s_{2})) \cdots$, where $s_{i} \in Subst_{s}^{+}$ and $\delta \vartheta_{1} \cdots \vartheta_{n} = \vartheta_{1} \cdots \vartheta_{n}$.

This operator takes for each word $\vartheta_{1} \cdots \vartheta_{n} \in Subst_{s}^{+}$ the parallel composition, thus turning it into one maximally parallel step. Further, it passes through the result of previous steps to the next one to be considered. This mimics the behavior of the $\sim_{MP}$ operator in the definition of $D_{MP}$. Now we can establish the following theorem.

**Theorem 8.8** $D_{MP} = \alpha_{MP} \circ I$

**Proof** Again it can easily be shown that $\alpha_{MP} \circ I$ is a fixed-point of $\Psi_{MP}$.

Combining the two above theorems yields the following corollary.

**Corollary 8.9** $D_{MP} = \alpha_{MP} \circ \alpha \circ D_{FI}$
8.3 Relating $D_{MP}$ and Dec: an intermediate model $D_{CS}$

We introduce an intermediate denotational semantics $D_{CS}$ (CS is an abbreviation for computed substitutions), to which both $D_{MP}$ and Dec will be related. It can be seen as a denotational variant of Dec, which yields for every goal the set of computed answer substitutions; since it delivers a set of substitutions, rather than a set of sequences of substitutions, it models only success behavior. Like $D_{MP}$ it is a model for maximal parallelism. Formally, $D_{CS}$ is introduced as the least fixed-point of a continuous function on a complete lattice, which we introduce next.

**Definition 8.10** The set $P_{CS}$, with typical element $F$ is given by $P_{CS} = \text{Goal} \rightarrow \mathcal{P}(\text{Subst})$.

The set $\mathcal{P}(\text{Subst})$ of sets of substitutions, is a complete lattice with respect to set inclusion. Thus $P_{CS}$ is also a complete lattice, when supplied with the inclusion relation induced by the one on $\mathcal{P}(\text{Subst})$: $f_1 \subseteq f_2$ iff $\forall \leftarrow \hat{A}[f_1(\leftarrow \hat{A}) \subseteq f_2(\leftarrow \hat{A})]$. Since we do not need to consider sequences, a lattice structure, rather than a metric one, suffices as a domain for $D_{CS}$.

The least upper bound of a set $F \subseteq P_{CS}$, denoted by $\bigcup_{F \in F}$, is defined by

$$\left( \bigcup_{F \in F} \right)[\leftarrow \hat{A}] = \bigcup_{F \in F} F[\leftarrow \hat{A}]$$

Before giving the definition of $D_{CS}$ we first extend the definition of $\circ$, the operator for the parallel composition of substitutions, to sets of substitutions. We put, for $X, Y \in \mathcal{P}(\text{Subst})$,

$$X \circ Y = \{ \theta \circ \sigma : \theta \in X, \sigma \in Y, \theta \circ \sigma \neq \emptyset \}$$

The following lemma states that it is continuous, a fact that we shall need in the definition below.

**Lemma 8.11** Let $\{X_m\}_{m \geq 0}, \{Y_n\}_{n \geq 0}$ be chains in $\mathcal{P}(\text{Subst})$ (for $X_k \subseteq X_{k+1} \land Y_k \subseteq Y_{k+1}$). Then $\bigcup_{k \geq 0} (X_k \circ Y_k) = (\bigcup_{m \geq 0} X_m) \circ (\bigcup_{n \geq 0} Y_n)$.

Next we introduce $D_{CS}$:

**Definition 8.12** Let $D_{CS} : \text{Goal} \rightarrow \mathcal{P}(\text{Subst})$ be the least fixed-point of the continuous (with respect to the lattice structure on $P_{CS}$) operator $\Psi_{CS} : (\text{Goal} \rightarrow \mathcal{P}(\text{Subst})) \rightarrow (\text{Goal} \rightarrow \mathcal{P}(\text{Subst}))$, given by

$$\Psi_{CS}(F)[\emptyset] = \{ \epsilon \}$$

$$\Psi_{CS}(F)[\leftarrow A] = \bigcup\{(\text{mg}(A, H) \circ F(\leftarrow B))|_{\text{Var}(A)} : H \leftarrow B \in W\}$$

$$\Psi_{CS}(F)[\leftarrow \hat{A}_1, \hat{A}_2] = \Psi_{CS}(F)[\leftarrow \hat{A}_1] \circ \Psi_{CS}(F)[\leftarrow \hat{A}_2]$$

The continuity of $\Psi_{CS}$ is a direct consequence of Lemma 8.11.

8.4 Relating $D_{MP}$ and $D_{CS}$

The relation between the models $D_{MP}$ and $D_{CS}$ is described by the abstraction operator

$$\alpha_{CS} : FST \rightarrow \mathcal{P}(\text{Subst}) \text{ defined by } \alpha_{CS}(X) = \text{last}(X \cap \text{Subst}^+)$$

(The function last used above yields for a set of finite sequences the set of their last elements.) We have the following theorem.

**Theorem 8.13** $D_{CS} = \alpha_{CS} \circ D_{MP}$

The theorem is immediate from the following two lemmas, which can be proved by induction on $n$. Let the functions $\perp$ and $f_{ev}$ be defined by $\perp (\leftarrow \hat{A}) = \emptyset$ and $f_{ev}(\leftarrow \hat{A}) = \{e\}$, for all $\leftarrow \hat{A}$. 14
Lemma 8.14 For all n: \( \Psi^n_{CS}(\bot) = (\alpha_{CS} \circ \Psi^n_{MP})(f_{\bot}) \)

Lemma 8.15 For all n and \( \bar{B} \): \( (\alpha_{CS} \circ \Psi^n_{MP})(f_{\bot})(\bar{B}) \subseteq (\alpha_{CS} \circ \Psi^{n+1}_{MP})(f_{\bot})(\bar{B}) \)

Proof of Theorem 8.13: For any \( \bar{B} \) we have

\[
(\alpha_{CS} \circ D_{MP})(\bar{B}) = \alpha_{CS}\left( \lim_{n \to \infty} \Psi^n_{MP}(f_{\bot})(\bar{B}) \right)
\]

\[
= (\text{Lemma 8.15}) \alpha_{CS}\left( \bigcup_n \Psi^n_{MP}(f_{\bot})(\bar{B}) \right)
\]

\[
= \bigcup_n \alpha_{CS}(\Psi^n_{MP}(f_{\bot})(\bar{B}))
\]

\[
= (\text{Lemma 8.14}) \bigcup_n \Psi^n_{CS}(\bot)(\bar{B})
\]

\[
= D_{CS}(\bar{B}).
\]

\[\square\]

8.5 Relating \( D_{CS} \) and \( Dec \)

Next we shall compare the denotational semantics modeling the computed answer substitutions, on the one hand, and the declarative semantics, on the other. The relation will be given by defining two uniform functions, \( \nu \) and \( \mu \) and by showing that \( Dec = \nu(D_{CS}) \) and \( D_{CS} = \mu(Dec) \). Here uniform means that these two functions do not depend upon the specific program \( W \).

The sketch of the proof is the following: first we consider a sub-domain \( P \) of the domain of \( \Psi_{CS} \) such that \( \mu \) and \( \nu \) make \( T \) and \( \Psi_{CS} \) to commute on this domain, namely: \( \Psi_{CS}(\mu(I)) = \mu(T(I)) \), and \( \nu(\Psi_{CS}(F)) = T(\nu(F)) \) for all \( F \in P \). Then we show that \( \nu \) allows to simulate step by step the fixed-point construction of \( \Psi_{CS} \) by \( T \) and vice-versa, namely: for each \( n \geq 0, T^n(\theta) = \nu(\Psi^n_{CS}(F_0)) \) and \( \Psi^n_{CS}(F_0) = \mu(T^n(\theta)) \) (where \( F_0 \) is the minimal element of \( P \)). Finally, by continuity of \( \nu \) and \( \mu \), we can commute also the least upper bounds of these chains, so that \( \text{lfp}(T) = \bigcup_{n \geq 0} T^n(\theta) = \nu(\bigcup_{n \geq 0} \Psi^n_{CS}(F_0)) = \nu(\text{lfp}(\Psi_{CS})) \) and \( \text{lfp}(\Psi_{CS}) = \bigcup_{n \geq 0} \Psi^n_{CS}(F_0) = \mu(\bigcup_{n \geq 0} T^n(\theta)) = \mu(\text{lfp}(T)) \).

We use the following notation: \( \text{Var}(A) \) is the set of variables occurring in \( A \). \( \text{Dom}(\theta) \) (the domain of \( \theta \)) is the set \( \{ x : x \theta \neq x \} \). \( \text{Cod}(\theta) \) (the codomain of \( \theta \)) is the set \( \bigcup_{x \in \text{Dom}(\theta)} \text{Var}(x \theta) \). If \( X \) is a set of substitutions and \( A \) is an atom, then \( X_A \) is the set \( \{ \theta_A : \theta \in X \} \), where \( \theta_A \) is any renaming of \( \theta \) with respect to \( A \), i.e., such that \( \forall x[\text{Var}(x \theta_A) \cap \text{Var}(A) = \emptyset] \).

Definition 8.16 \( P \) is the subset of \( P_{CS} = \text{Goal} \to \mathcal{P} \text{(Subst)} \) of all elements \( F \) that satisfy the following properties.

1. \( F[\emptyset] = \{ \} \)
2. \( \forall \theta [\text{Dom}(\theta) \subseteq \emptyset \Rightarrow (\theta \circ \bar{F}[\bar{\varphi}]_{p(\bar{\varphi})=p(\bar{\varphi})})_{p(\bar{\varphi})=p(\bar{\varphi})} = \bar{F}[\bar{\varphi}]_{p(\bar{\varphi})=p(\bar{\varphi})}] \)
3. \( F[\bar{A}, \bar{A}_2] = F[\bar{A}] \circ F[\bar{A}_2] \)
4. \( \forall \bar{A} \forall \theta \in F[\bar{\varphi}]_{\text{Dom}(\theta) \subseteq \text{Var}(\bar{A})} \)

The motivation of these restrictions is of a technical nature: the set \( P \) will turn out to be isomorphic to the set \( P_{Dec} \). The isomorphism pair, \( \langle \nu, \mu \rangle \), will be defined later. \( R3 \) requires the information given by \( F \) about generic goals to be obtainable by the information about atomic goals. This corresponds to the compositional nature of interpretations in \( P_{Dec} \): the meaning of a conjunction is declaratively defined in terms of its conjuncts. \( R2 \) also reflects a kind of compositionality: the possibility to obtain the information about an instantiated atom from the uninstantiated one. \( R1 \) and \( R4 \) impose a sort of minimality on the information associated to a goal.

The set \( P \) is a complete partial order with respect to the ordering it inherits from \( P_{CS} \). This we prove next.
Proposition 8.17 \((P, \subseteq)\) is a complete partial order; the least upper bound of a chain \((F_n)\) is given by \((\bigcup_n F_n)\). In other words, \((P, \subseteq)\) is a sub CPO of \((\text{PCs}, \subseteq)\).

Proof We have to show that for any chain \((F_n)\) in \(P\), \(\bigcup_n F_n\) preserves the properties R1-R4. R1, R2 and R4 are obvious. R3 follows by lemma 8.11. \(\square\)

Definition 8.18
- The function \(\nu : P \to P_{\text{Dec}}\) is defined by
  \[\nu(F) = \{p(\overline{x})\theta : \theta \in F[\overline{\theta}] \land p(\overline{x}) \in \text{EAtom}\}\]
- The function \(\mu : P_{\text{Dec}} \to P\) is defined by
  \[
  \begin{align*}
  \mu(I)[\emptyset] & = \{\epsilon\} \\
  \mu(I)[\overline{A}] & = \{\operatorname{mgu}(A, A')_I : A' \in I \land \operatorname{Var}(\overline{A'}) \cap \operatorname{Var}(\overline{A}) = \emptyset\} \\
  \mu(I)[\overline{A}, \overline{A'}] & = \mu(I)[\overline{A}] \circ \mu(J)[\overline{A'}]
  \end{align*}
  \]

Remark 8.19 The function \(\mu\) is well defined, i.e., \(\forall I \in P_{\text{Dec}} [\mu(I) \in P]\).

Proof R1, R3 and R4 are trivial, and R2 is an immediate consequence of the following lemma.

Lemma 8.20 Let \(\theta\) be an idempotent substitution, and assume \(\text{Dom}(\theta) \subseteq \overline{x}\). Let \(A\) be an atom such that \(\operatorname{Var}(A) \cap \{\overline{x}\} = \emptyset\) and \(\operatorname{Var}(A) \cap \operatorname{Var}([\overline{x} \theta]) = \emptyset\). Then
  \[\theta \circ \operatorname{mgu}(p(\overline{x})\theta, A)[p(\overline{x})\phi].\]

Proof The proof uses some elementary properties of idempotent substitutions (see [Pal89]).

The following facts can be readily established.

Proposition 8.21 The functions \(\nu\) and \(\mu\) are continuous.

Proposition 8.22 \(P\) is closed with respect to \(\Psi_{\text{CS}}\), i.e., \(\forall F \in P [\Psi_{\text{CS}}(F) \in P]\).

The following result shows that \(\mu\) and \(\nu\) commute the functions \(\Psi_{\text{CS}}\) and \(T\) on \(P\).

Lemma 8.23
1. If \(F \in P\) then \(\nu(\Psi_{\text{CS}}(F)) = T(\nu(F))\)
2. If \(I \in P_{\text{Dec}}\) then \(\Psi_{\text{CS}}(\mu(I)) = \mu(T(I))\)

The functions \(\nu\) and \(\mu\) allow to simulate, step by step, the fixed-point construction of \(\mathcal{D}_{\text{CS}}\) in \(\text{Dec}\), and vice-versa. There is only one difficulty: the fixed-point construction of \(\Psi_{\text{CS}}\) starts from the minimal element of \(P_{\text{CS}}\), that is the function \(F_1\) such that for every \(A, F_1[\overline{A}] = \emptyset\). Unfortunately, \(F_0\) is not the minimal element of \(P\), in fact \(F_0 \not\in P\). The minimal element of \(P\) is the function \(F_0\) such that
  \[
  F_0[\overline{A}] = \begin{cases} 
  \{\epsilon\} & \text{iff } \overline{A} = \emptyset \\
  \emptyset & \text{otherwise}
  \end{cases}
  \]

However, the fixed-point of \(\Psi_{\text{CS}}\) can be also obtained by starting from \(F_0\), as the following remark shows.

Remark 8.24 We have \(F_0 = \mu(\emptyset)\) and \(F_0 = \Psi_{\text{CS}}(F_1)\).

Lemma 8.25
1. \(\forall n \geq 0 [T^n(\nu(F_0)) = \nu(\Psi_{\text{CS}}(F_0))]\)
Proof By induction on $n$.

Finally, we show the correspondence between $\mathcal{D}_{CS}$ and $\mathcal{D}_{\text{Dec}}$

**Theorem 8.26**

1. $\mathcal{D}_{\text{Dec}} = \nu(\mathcal{D}_{CS})$
2. $\mathcal{D}_{CS} = \mu(\mathcal{D}_{\text{Dec}})$

**Proof**

1. $\mathcal{D}_{\text{Dec}} = \nu(\mathcal{D}_{CS})$
   
   $= \bigcup_{n \geq 0} T^n(\emptyset)$
   
   $= \bigcup_{n \geq 0} T^n(\nu(F_0))$ (by Remark 8.24, part 1)
   
   $= \bigcup_{n \geq 0} \nu(\Psi^n_{CS}(F_0))$ (by Lemma 8.25, part 1)
   
   $= \nu(\bigcup_{n \geq 0} \nu(\Psi^n_{CS}(F_0)))$ (by continuity of $\nu$)
   
   $= \nu(\nu(\Psi_{CS}))$ (by Remark 8.24, part 2)
   
   $= \nu(\mathcal{D}_{CS})$. \hfill $\square$

2. Similar to the previous one.

**9 Collecting the results**

After the long and exhausting previous section, the reader might be comforted by a schematic overview of the relationships that were established. We have the following equalities.

- $\mathcal{O}_{FI} = \beta_{FI} \circ \mathcal{D}_{FI}$
- $\mathcal{I} = \alpha \circ \mathcal{D}_{FI}$
- $\mathcal{O}_{MP} = \mathcal{D}_{MP} = \alpha_{MP} \circ \mathcal{I}$
- $\mathcal{D}_{CS} = \alpha_{CS} \circ \mathcal{D}_{MP}$
- $\mathcal{D}_{CS} = \mu(\mathcal{D}_{\text{Dec}})$
- $\mathcal{D}_{\text{Dec}} = \nu(\mathcal{D}_{CS})$

In Figure 9, these equalities are graphically represented. Moreover, it contains some arrows between $\mathcal{O}_{FI}$ and the sets $\mathcal{O}_{SS}$, $\mathcal{O}_{FFS}$ and $\mathcal{O}_{IFS}$, indicating that the definition of these sets is based on that of $\mathcal{O}_{FI}$.

Combining some of the equalities above, we find

$$\mathcal{D}_{\text{Dec}} = \nu(\alpha_{CS} \circ \alpha_{MP} \circ \alpha \circ \mathcal{D}_{FI}),$$

a maybe somewhat complicated but precise relationship between the declarative semantics $\mathcal{D}_{\text{Dec}}$ and the denotational semantics $\mathcal{D}_{FI}$. From this the following theorem, which establishes the soundness and completeness of the declarative semantics, is fairly immediate. Thus an alternative is given for the quite complicated proof that is given in [FLPM89]. The fact that here the relationship between $\mathcal{D}_{\text{Dec}}$ and $\mathcal{D}_{FI}$ and, hence, between $\mathcal{D}_{\text{Dec}}$ and $\mathcal{O}_{SS}$ has been decomposed into several steps makes the proof below more transparent.

**Theorem 9.1** $A \in \mathcal{O}_{SS} \iff A \in \mathcal{D}_{\text{Dec}}$

**Proof**
We deduce from the equalities above a second fact, which says that \( \mathcal{O}_{SS}, \mathcal{O}_{FFS} \) and \( \mathcal{O}_{IFS} \) can be characterized in terms of \( \mathcal{O}_{MP} \) (\( = \mathcal{D}_{MP} \)), instead of \( \mathcal{O}_{PI} \). In other words, for the semantics of an HCL program, it does not matter whether we consider an interleaving or a maximally parallel model. Although this might seem not very surprising, it is not completely straightforward, since \( \mathcal{O}_{MP} \) and \( \mathcal{O}_{PI} \) have a different deadlock behavior: the former delivers deadlock for more goals than the latter. (See the counter example at the end of section 4.1.)

**Theorem 9.2** We have the following equalities.

\[
\begin{align*}
\mathcal{O}_{SS} &= \{ p(x) \theta : p(x) \in E\text{Atom} \land \theta \in \text{last}(\mathcal{O}_{MP}[\leftarrow p(x)] \cap \text{Subst}^+) \} \\
\mathcal{O}_{FFS} &= \{ A : \mathcal{O}_{MP}[\leftarrow A] \subseteq \text{Subst}^* \cdot \delta \} \\
\mathcal{O}_{IFS} &= \{ A : \mathcal{O}_{MP}[\leftarrow A] \cap \text{Subst}^* = \emptyset \land \mathcal{O}_{MP}[\leftarrow A] \cap \text{Subst}^w \neq \emptyset \}
\end{align*}
\]

**Proof** Similar to that of the previous theorem.
Acknowledgements

We thank Jean-Marie Jacquet, Peter Knijnenburg and Erik de Vink for their detailed comments on a draft of this paper.

References


