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

De Boer, F.S., J.N. Kok, C. Palamidessi and J.J.M.M. Rutten, Theoretical Computer Science 101 (1992) 239-263.

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.

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 $H \leftarrow \bar{B}$, where H is an atom and $\bar{B}$ is a finite sequence of atoms. We shall introduce

* 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.

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 we have been recently investigating various models for more advanced parallel logic languages like GHC and Parlog [4, 5], which contain constructs such as 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 [10]). The first one, called $\mathcal{O}_{\mathrm{FI}}$ (FI for fair interleaving), corresponds to the standard (sequential) operational semantics of HCL based on SLD resolution (like in [15, 12]); it uses a fair derivation rule (reduction from left to right) in order to model also failure behavior. From $\mathcal{O}_{\mathrm{FI}}$ 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, $\mathcal{O}_{\mathrm{MP}}$, 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 $\hat{\circ}$ for substitutions (introduced in [13]). 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. [14]). Technically, $\mathcal{O}_{\mathrm{MP}}$ (or, more precisely, the denotational model corresponding to it) will play a role as an intermediate in establishing the correspondence between $\mathcal{O}_{\mathrm{FI}}$ 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 $\mathcal{O}_{\mathrm{FI}}$ will be called $\mathcal{D}_{\mathrm{FI}}$. 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 $\mathcal{O}_{\mathrm{FI}}$ and $\mathcal{O}_{\mathrm{MP}}$ 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 $\mathcal{D}_{\mathrm{FI}}$ with respect to $\mathcal{O}_{\mathrm{FI}}$ by showing that the latter equals the composition of an abstraction operation with the former.

Next a second denotational model, called $\mathcal{D}_{\mathrm{MP}}$, is introduced, which equals the operational semantics $\mathcal{O}_{\mathrm{MP}}$. Its semantic universe is the same as the one of $\mathcal{O}_{\mathrm{MP}}$, which is simpler than that of $\mathcal{D}_{\mathrm{FI}}$. The semantic operator for the parallel composition (conjunction) of two goals is the operator $\hat{\circ}$ 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 $\mathcal{D}ec$ as introduced in [9]. 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 [15] (see also [12]). 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 [9] extends this approach in that interpretations may also contain non-ground atoms. Therefore $\mathcal{D}ec$ 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 [3]. 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. [6].) 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 that we have 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, $\mathscr{D}_{\mathrm{FI}}$, and the declarative model $\mathscr{D}ec$. This is done in two steps.

First we shall relate $\mathscr{D}_{\mathrm{FI}}$ and $\mathscr{D}_{\mathrm{MP}}$. To this end, an intermediate denotational model $\mathscr{I}$ is introduced, to which both will then be related. Secondly—and this is the more difficult part—$\mathscr{D}_{\mathrm{MP}}$ and $\mathscr{D}ec$ are compared. Again an intermediate denotational semantics, called $\mathscr{D}_{\mathrm{CS}}$ (CS for computed substitutions) is introduced. It is essentially a model for maximal parallelism, like $\mathscr{D}_{\mathrm{MP}}$, 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 $\mathscr{D}_{\mathrm{MP}}$ and $\mathscr{D}_{\mathrm{CS}}$ 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, $\mathscr{D}_{\mathrm{CS}}$ and $\mathscr{D}ec$ 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 $\mathscr{O}_{\mathrm{FI}}$). In this way, we find a fairly transparent alternative to the equivalence proof given in [9], 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., [7].) 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 $f: M_1 \to M_2$ is called *nonexpansive* 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 $\varepsilon \in [0, 1)$ such that for all $x, y \in M_1$

$$d_2(f(x), f(y)) \leq \varepsilon \cdot d_1(x, y).$$

Nonexpansive and contracting functions are continuous. The following fact is known as Banach's Theorem: let (M, d) be a complete metric space and $f: M \to M$ 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 $M_1 \to M_2$ 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

$$\mathcal{P}_{\text{nco}}(M) = \{X : X \subseteq M \wedge X \text{ is nonempty and compact}\}.$$

We can turn $\mathcal{P}_{\text{nco}}(M)$ 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 $X, Y \in \mathcal{P}_{\text{nco}}(M)$

$$d_{\text{H}}(X, Y) = \max\{\sup_{x \in X}\{d(x, Y)\}, \sup_{y \in Y}\{d(y, X)\}\}$$

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

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 [12, 1].

The sets *Term* of terms, $(A, B, H \in)$ *Atom* of atomic formulas (or atoms), and $(\theta, \sigma, \gamma \in)$ *Subset* of substitutions are defined as usual. Elementary atoms (*EAtom*) are of the form $p(\bar{x})$, where p is a predicate and $\bar{x}$ 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 $\bar{B}$) 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 $\leftarrow A_1, \ldots, A_n$ $(n \geq 0)$, where each A_i is an atom. If $n > 0$ we denote $\leftarrow A_1, \ldots, A_n$ also by $\leftarrow \bar{A}$. If $n = 0$ we have the so-called empty goal, and we write $\square$. The set of all goals is denoted by *Goal*.

We have the usual notion of *most general unifier* of two atoms A and H, denoted by $mgu(A, H)$. For the composition of two substitutions we write $\theta_1\theta_2$. For technical convenience, we shall throughout this paper consider only idempotent substitutions, i.e., satisfying $\theta\theta = \theta$ (see [13] for some discussion on ths point). The set of variables occurring in the atom A is indicated by $Var(A)$. For an atom A and a substitution θ we write $\theta_{|A}$ for the restriction of θ to $Var(A)$. The empty substitution is denoted by ε.

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

$$\leftarrow (A_1, \ldots, A_{i-1}, B_1, \ldots, B_n, A_{i+1}, \ldots, A_m)\theta$$

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 $\square$; 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 ($\mathcal{O}_{\mathrm{SS}}$), containing all the atoms that have a refutation, instantiated by the last substitution (the so-called computed answer substitution);
- the finite failure set ($\mathcal{O}_{\mathrm{FFS}}$), containing all the atoms for which all the fair SLD-derivations are failing (see [2]);
- the infinite failure set ($\mathcal{O}_{\mathrm{IFS}}$), 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 this paper, we use a different notion of success set: we take the one introduced in [8, 9], which contains all the elementary atoms that have a refutation, instantiated by the computed answer substitution (see the next section and the one on the declarative semantics).

4. Operational semantics

We present two operational semantics for HCL, both based on a labelled transition system (in the style of [10]). 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 $(Goal, Subst, \rightarrow)$ be the labelled transition system, whose transition relation $\rightarrow \subseteq Goal \times Subst \times Goal$ is defined as the smallest relation satisfying the following axiom:

$$\leftarrow A, \bar{A} \xrightarrow{\theta} \leftarrow \bar{A}\theta, \bar{B}\theta.$$

(As usual, we write $\leftarrow \bar{A} \xrightarrow{\theta} \leftarrow \bar{B}$ rather than $(\leftarrow \bar{A}, \theta, \leftarrow \bar{B}) \in \longrightarrow$.) Here $\theta = mgu(A, H)$ and $H \leftarrow \bar{B}$ is a clause of W. We assume this clause to be renamed such that A, $\bar{A}$ 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 [12]). 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 $\mathcal{O}_{\mathrm{FI}}: Goal \rightarrow P_{\mathrm{ST}}$, which associates with a goal a set of sequences of substitutions. The semantic universe $(X, Y \in)\, P_{\mathrm{ST}}$ (ST is an abbreviation for streams) is given by

$$P_{\mathrm{ST}} = \mathcal{P}_{\mathrm{nco}}(Subst^{\mathrm{st}}_{\delta}),$$

where $Subst^{\mathrm{st}}_{\delta}$, the set of finite, infinite and deadlocking sequences (or words, or streams), is defined by

$$(v, w, z \in)\; Subst^{\mathrm{st}}_{\delta} = Subst^{*} \cup Subst^{\omega} \cup Subst^{*} \cdot \delta.$$

As a metric on P_{ST} we take the Hausdorff metric induced by the standard metric on sequences (see the preliminaries). The empty sequence is denoted by λ and the concatenation of two sequences w_1 and w_2 by $w_1 \cdot w_2$. To denote failure we have added to the set of substitutions a special element δ. We postulate for any substitution θ that $\theta\delta$, the composition of θ and δ, equals δ; for any sequence of substitutions v we have that $\delta \cdot v$, the concatenation of δ and v, is equal to δ. 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 δ (elements of $Subst^{+}$) correspond to successfully terminating computations (refutations). Sequences ending in δ (in $\mathit{Subst}^* \cdot \delta$) represent failing computations. Infinite sequences (in Subst^ω) are associated with infinitely failing computations.

Definition 4.2. Let $\mathcal{O}_{\mathrm{FI}}$ be the unique fixed point of the contracting operator $\Phi_{\mathrm{FI}} : (\mathit{Goal} \to P_{\mathrm{ST}}) \to (\mathit{Goal} \to P_{\mathrm{ST}})$, which is given by

$$\Phi_{\mathrm{FI}}(F)[\![\square]\!] = \{\varepsilon\}$$

$$\Phi_{\mathrm{FI}}(F)[\![\leftarrow \bar{A}]\!] = \bigcup \{\theta \cdot (\theta \rightsquigarrow_{\mathrm{FI}} F[\![\bar{A}']\!]) : \leftarrow \bar{A} \xrightarrow{\theta} \leftarrow \bar{A}'\}$$
$$\cup \{\delta : \leftarrow \bar{A} \not\longrightarrow\}$$

Here $\rightsquigarrow_{\mathrm{FI}} : \mathit{Subst} \times P_{\mathrm{ST}} \to P_{\mathrm{ST}}$ is defined by $\theta \rightsquigarrow_{\mathrm{FI}} X = \{\theta \rightsquigarrow_{\mathrm{FI}} x : x \in X\}$, with

$$\theta \rightsquigarrow_{\mathrm{FI}} \lambda = \lambda,$$

$$\theta \rightsquigarrow_{\mathrm{FI}} (\sigma \cdot z) = (\theta\sigma) \cdot (\theta \rightsquigarrow_{\mathrm{FI}} z).$$

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

The definition of $\mathcal{O}_{\mathrm{FI}}[\![\square]\!]$ is obvious. For a nonempty goal $\leftarrow \bar{A}$ we have that $\mathcal{O}_{\mathrm{FI}}[\![\leftarrow \bar{A}]\!]$ equals $\{\delta\}$ if there are no transitions possible from $\leftarrow \bar{A}$ (indicated by $\leftarrow \bar{A} \not\longrightarrow$). Otherwise, $\mathcal{O}_{\mathrm{FI}}[\![\leftarrow \bar{A}]\!]$ contains all sequences that start with θ and continue with a sequence stemming from $\mathcal{O}_{\mathrm{FI}}[\![\leftarrow \bar{A}']\!]$, in which every element is composed with θ. The latter is caused by the application of $\theta \rightsquigarrow_{\mathrm{FI}}$ to $\mathcal{O}_{\mathrm{FI}}[\![\leftarrow \bar{A}']\!]$, 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 $\mathcal{O}_{\mathrm{FI}}$ 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 $\theta \rightsquigarrow_{\mathrm{FI}}$. This could have been avoided as well by using a different type of transition system, in which a configuration $\langle \leftarrow \bar{A}, \sigma \rangle$ would consist of both a goal and a substitution. The latter could then be used to store all the bindings found so far. The axiom corresponding to the one above would be

$$\langle (\leftarrow A, \bar{A}), \sigma \rangle \longrightarrow \langle (\leftarrow \bar{A}, \bar{B}), \sigma\theta \rangle$$

with $\bar{B}$ and θ as above.

The following counterexample shows that $\mathcal{O}_{\mathrm{FI}}$ is *not* compositional. Consider the following program

$$\{p(x) \leftarrow s_1(x), p(x) \leftarrow s_2(x), q(x) \leftarrow s_3(x)$$
$$s_1(a) \leftarrow, s_2(b) \leftarrow, s_3(a) \leftarrow, s_3(b) \leftarrow, r(a) \leftarrow\}.$$

It is easy to see that with respect to this program $\mathcal{O}_{\mathrm{FI}}[\![\leftarrow p(x)]\!] = \mathcal{O}_{\mathrm{FI}}[\![\leftarrow q(x)]\!]$. But, on the other hand, we have $\{x/a\}\delta \in \mathcal{O}_{\mathrm{FI}}[\![\leftarrow r(x), p(x)]\!] \backslash \mathcal{O}_{\mathrm{FI}}[\![\leftarrow r(x), q(x)]\!]$.

4.2. *Success, finite failure and infinite failure sets*

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

$$\mathcal{O}_{\mathrm{SS}} = \{ p(\bar{x})\theta : p(\bar{x}) \in \mathit{EAtom} \wedge \theta \in \mathit{last}(\mathcal{O}_{\mathrm{FI}}[\![\leftarrow p(\bar{x})]\!] \cap \mathit{Subst}^{+})\},$$

$$\mathcal{O}_{\mathrm{FFS}} = \{A : \mathcal{O}_{\mathrm{FI}}[\![\leftarrow A]\!] \subseteq \mathit{Subst}^{*} \cdot \delta\},$$

$$\mathcal{O}_{\mathrm{IFS}} = \{A : \mathcal{O}_{\mathrm{FI}}[\![\leftarrow A]\!] \cap \mathit{Subst}^{*} = \emptyset \wedge \mathcal{O}_{\mathrm{FI}}[\![\leftarrow A]\!] \cap \mathit{Subst}^{\omega} \neq \emptyset\}.$$

In the first set, the function *last* takes from a set of sequences the last elements (not equal to δ). Those elements represent the computed answer substitution for successful refutations. The notion of success set we consider here is introduced in [9, 8] and extends the standard one given in [12, 15]. (See also the section on the declarative semantics for some more discussion.) The second set, $\mathcal{O}_{\mathrm{FFS}}$, contains those atoms that give rise to only failing computations, i.e., sequences of substitutions that end in δ. The last set, $\mathcal{O}_{\mathrm{IFS}}$, 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. [11].) It is defined as follows.

Definition 4.3. We define the parallel composition of two substitutions θ and σ, denoted by $\theta \,\hat{\circ}\, \sigma$, by

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

where $S(\theta) = \{\langle x, t\rangle : x/t \in \theta\}$. Furthermore we define $\theta \,\hat{\circ}\, \delta = \delta \,\hat{\circ}\, \theta = \delta$. (Note that the notion of *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 δ. It is straightforward to show that $\hat{\circ}$ is commutative, associative, and indempotent (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 θ_1 and θ_2*

$$\theta_1 \hat{\circ} \theta_2 = \theta_1 mgu(S(\theta_2)\theta_1)$$

where $S(\theta_2)\theta_1 = \{\langle x\theta_1, t\theta_1\rangle : \langle x, t\rangle \in S(\theta_2)\}$.

For the proof of this lemma and additional discussion of $\hat{\circ}$ we refer to [13]. The definition of $\hat{\circ}$ is illustrated by the following example.

Example 4.5. Let $\theta_1 = \{x/f(y, a), z/g(b)\}$ and $\theta_2 = \{x/f(b, w), z/g(y)\}$. Then

$$\begin{aligned}\theta_1 \hat{\circ} \theta_2 &= mgu\{\langle x, f(y, a)\rangle, \langle z, g(b)\rangle, \langle x, f(b, w)\rangle, \langle z, g(y)\rangle\}\\ &= \{x/f(b, a), z/g(b), y/b, w/a\}.\end{aligned}$$

If we take θ_1 as before and $\theta_2 = \{x/f(a, w), z/g(y)\}$, then we have

$$\begin{aligned}\theta_1 \hat{\circ} \theta_2 &= mgu\{\langle x, f(y, a)\rangle, \langle z, g(b)\rangle, \langle x, f(a, w)\rangle, \langle z, g(y)\rangle\}\\ &= \delta.\end{aligned}$$

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

Definition 4.6. We define

$$(1) \qquad \leftarrow A \xrightarrow{\theta} \leftarrow \bar{B}$$

where $\theta = mgu(A, H)$ and $H \leftarrow \bar{B}$ is an (appropriately renamed) clause of W.

$$(2) \qquad \frac{\leftarrow \bar{A} \xrightarrow{\theta} \leftarrow \bar{A}', \qquad \leftarrow \bar{B} \xrightarrow{\sigma} \leftarrow \bar{B}'}{\leftarrow \bar{A}, \bar{B} \xrightarrow{\theta \hat{\circ} \sigma} \leftarrow \bar{A}', \bar{B}'}.$$

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

Definition 4.7. The operational semantics $\mathcal{O}_{\mathrm{MP}}$ is defined as the fixed point of the contraction $\Phi_{\mathrm{MP}} : (Goal \to P_{\mathrm{ST}}) \to (Goal \to P_{\mathrm{ST}})$, given by

$$\Phi_{\mathrm{MP}}(F)[\![\Box]\!] = \{\varepsilon\}$$

$$\begin{aligned}\Phi_{\mathrm{MP}}(F)[\![\leftarrow \bar{A}]\!] = &\bigcup \{\theta \cdot (\theta \rightsquigarrow_{\mathrm{MP}} F[\![\leftarrow \bar{A}']\!]) : \leftarrow \bar{A} \xrightarrow{\theta} \leftarrow \bar{A}'\}\\ &\bigcup \{\delta : \leftarrow \bar{A} \not\longrightarrow\}\end{aligned}$$

Here $\rightsquigarrow_{\mathrm{MP}} : Subst \times P_{\mathrm{ST}} \to P_{\mathrm{ST}}$ is defined by $\theta \rightsquigarrow_{\mathrm{MP}} X = \{\theta \rightsquigarrow_{\mathrm{MP}} x : x \in X\}$, with

$$\theta \rightsquigarrow_{\mathrm{MP}} \lambda = \lambda,$$

$$\theta \rightsquigarrow_{\mathrm{MP}} \sigma \cdot z = \begin{cases}\delta & \text{if } \theta \hat{\circ} \sigma = \delta,\\ \theta \hat{\circ} \sigma \cdot (\theta \rightsquigarrow_{\mathrm{MP}} z) & \text{otherwise.}\end{cases}$$

The definition of $\mathcal{O}_{\mathrm{MP}}$ is very similar to that of $\mathcal{O}_{\mathrm{FI}}$. Two differences should be noticed here. First, the transition relation that is used is different from the one in the definition of $\mathcal{O}_{\mathrm{FI}}$; secondly, the definition of the function $\theta \rightsquigarrow_{\mathrm{MP}}$ differs from the function $\theta \rightsquigarrow_{\mathrm{FI}}$. It composes θ *in parallel* with the elements of $\mathcal{O}_{\mathrm{MP}}[\![\leftarrow \bar{A}']\!]$, as opposed to $\theta \rightsquigarrow_{\mathrm{FI}}$, 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 $\mathcal{D}_{\mathrm{FI}}$ for the operational interleaving semantics $\mathcal{O}_{\mathrm{FI}}$. We start by introducing the complete metric space P_{FI}, which is defined by

$$P_{\mathrm{FI}} = \mathcal{P}_{\mathrm{nco}}(((\mathit{Subst} \times \mathit{Subst}_\delta)^+)^\infty)$$

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 $\langle v_1, \ldots, v_n, \ldots \rangle$, 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, $\langle(\theta_1, \theta_2)\rangle$ and a vector $\langle v_1, v_2, \ldots \rangle$, and is defined by

$$\langle(\theta_1, \theta_2)\rangle \cdot \langle v_1, v_2, \ldots \rangle = \langle\langle(\theta_1, \theta_2)\rangle, v_1, v_2, \ldots \rangle.$$

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 δ. (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 $\|_{\mathrm{FI}}$. Operationally, a goal is executed by performing from left to right one step of each atom in the goal. The operator $\|_{\mathrm{FI}}$ 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 nth element of a vector represents the nth left to right swap of the goal.

Hence the operator $\|_{\mathrm{FI}}$ combines two vectors by concatenating their elements, i.e., their sequences of substitutions, component-wise.

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

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

where

$$\langle v_1, v_2, \ldots \rangle \|_{\mathrm{FI}} \langle w_1, w_2, \ldots \rangle = \langle v_1 \cdot w_1, v_2 \cdot w_2, \ldots \rangle,$$
$$\langle v_1, \ldots, v_n \rangle \|_{\mathrm{FI}} \langle w_1, w_2, \ldots \rangle = \langle v_1 \cdot w_1, \ldots, v_n \cdot w_n, w_{n+1}, \ldots \rangle.$$

Now we are ready to give the definition of the denotational semantics $\mathscr{D}_{\mathrm{FI}}$.

Definition 5.2. We define $\mathscr{D}_{\mathrm{FI}} : Goal \to P_{\mathrm{FI}}$:

$$\mathscr{D}_{\mathrm{FI}}[\![\Box]\!] = \{\lambda\},$$
$$\mathscr{D}_{\mathrm{FI}}[\![\leftarrow A]\!] = \{\langle(\theta, \theta mgu(A\theta, H))\rangle \cdot \mathscr{D}_{\mathrm{FI}}[\![\leftarrow \bar{B}]\!] : \theta \in Subst, H \leftarrow \bar{B} \in W\}$$
$$\bigcup \{\langle(\theta, \delta)\rangle : \forall H \leftarrow \bar{B} \in W[mgu(A\theta, H) \text{ does not exist}]\},$$
$$\mathscr{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}_1, \bar{A}_2]\!] = \mathscr{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}_1]\!] \|_{\mathrm{FI}} \mathscr{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}_2]\!].$$

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

The following example may help in understanding the definitions of $\mathscr{D}_{\mathrm{FI}}$ and $\|_{\mathrm{FI}}$. Consider a query $\leftarrow A_1, \ldots, A_m$. Assume that for any i the first reduction step of A_i delivers the input-output pair (α_i, β_i) and produces the conjunction B_i, C_i. Assume furthermore that the first reduction steps of B_i and C_i produce the pairs (γ_i, δ_i) and (ε_i, ϕ_i), respectively. Then the following vector will be an element of $\mathscr{D}_{\mathrm{FI}}[\![\leftarrow A_1, \ldots, A_m]\!]$:

$$\langle\langle(\alpha_1, \beta_1), \ldots, (\alpha_m, \beta_m)\rangle,$$
$$\langle(\gamma_1, \delta_1), (\varepsilon_1, \phi_1), \ldots, (\gamma_m, \delta_m), (\varepsilon_m, \phi_m)\rangle, \ldots\rangle.$$

In Section 8, the correctness of $\mathscr{D}_{\mathrm{FI}}$ with respect to $\mathscr{O}_{\mathrm{FI}}$ will be proved.

6. Denotational semantics for maximal parallelism

We next introduce a denotational variant, named $\mathscr{D}_{\mathrm{MP}}$, of the operational model $\mathscr{O}_{\mathrm{MP}}$ 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

$$P_{\mathrm{ST}} = \mathscr{P}_{\mathrm{nco}}(Subst_{\delta}^{\mathrm{st}}).$$

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

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

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

Here $x \|_{\mathrm{MP}} y$ is defined by the following cases.

$$(\sigma_1 \cdot z_1) \|_{\mathrm{MP}} (\sigma_2 \cdot z_2) = \begin{cases} \delta & \text{if } \sigma_1 \hat{\circ} \sigma_2 = \delta, \\ (\sigma_1 \hat{\circ} \sigma_2) \cdot (z_1 \|_{\mathrm{MP}} z_2) & \text{otherwise;} \end{cases}$$

$$\sigma_1 \|_{\mathrm{MP}} (\sigma_2 \cdot z) = (\sigma_2 \cdot z) \|_{\mathrm{MP}} \sigma_1 = \begin{cases} \delta & \text{if } \sigma_1 \hat{\circ} \sigma_2 = \delta, \\ (\sigma_1 \hat{\circ} \sigma_2) \cdot (\sigma_1 \|_{\mathrm{MP}} z) & \text{otherwise.} \end{cases}$$

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

Now we can introduce the semantics $\mathcal{D}_{\mathrm{MP}}$. It turns out to be equal to $\mathcal{O}_{\mathrm{MP}}$; this will be proved in Section 8.

Definition 6.2. Let the function $\mathcal{D}_{\mathrm{MP}}: Goal \rightarrow P_{\mathrm{ST}}$ be the unique fixed point of the contraction $\Psi_{\mathrm{MP}}: (Goal \rightarrow P_{\mathrm{ST}}) \rightarrow (Goal \rightarrow P_{\mathrm{ST}})$, given by

$$\Psi_{\mathrm{MP}}(F)[\![\square]\!] = \{\varepsilon\},$$

$$\Psi_{\mathrm{MP}}(F)[\![\leftarrow A]\!] = \bigcup \{mgu(A, H) \rightsquigarrow_{\mathrm{MP}} F(\leftarrow \bar{B}) : H \leftarrow \bar{B} \in W\}$$

$$\cup\ \{\delta : \forall H \leftarrow \bar{B} \in W\, mgu(A, H) \text{ does not exist}\},$$

$$\Psi_{\mathrm{MP}}(F)[\![\leftarrow \bar{A}_1, \bar{A}_2]\!] = \Psi_{\mathrm{MP}}(F)[\![\leftarrow \bar{A}_1]\!] \|_{\mathrm{MP}} \Psi_{\mathrm{MP}}(F)[\![\leftarrow \bar{A}_2]\!].$$

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

7. Declarative semantics

In this section, we recall the definition of the declarative semantics $\mathcal{D}ec$ introduced in [9]. 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 Emden and Kowalski in [15]. 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

[9] 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 [9] 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 $(P_{\mathrm{Dec}}, \subseteq)$, where $P_{\mathrm{Dec}} = \mathcal{P}(Atom)$.

Proposition 7.2. *$(P_{\mathrm{Dec}}, \subseteq)$ is a complete lattice.*

Definition 7.3. The (extended) immediate consequence operator $T: P_{\mathrm{Dec}} \to P_{\mathrm{Dec}}$, is defined by

$$T(I) = \{H mgu(\bar{B}, \bar{B}') : H \leftarrow \bar{B} \in W, \bar{B}' \in I\}.$$

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

$$T^0(I) = I, \qquad T^{n+1}(I) = T(T^n(I)).$$

The declarative semantics is defined as follows.

Definition 7.5. $\mathcal{D}ec = lfp(T)$.

The next theorem gives the relation between the model-theoretic semantics of W and $\mathcal{D}ec$.

Theorem 7.6. *For every atom A,*

$$W \models A \ (\textit{i.e., } A \textit{ is a logical consequence of } W) \quad \text{iff}$$
$$\exists A' \in \mathcal{D}ec \, \exists \theta \in Subst[A'\theta = A].$$

Finally, the following result expresses the relation between $\mathcal{D}ec$ and the success set.

Theorem 7.7. $\mathcal{D}ec = \mathcal{O}_{\mathrm{SS}}$.

8. The relations between the models

8.1. The relations between the denotational and the operational models

8.1.1. Relating $\mathcal{O}_{\mathrm{FI}}$ and $\mathcal{D}_{\mathrm{FI}}$

We start with the relation between $\mathcal{O}_{\mathrm{FI}}$ and $\mathcal{D}_{\mathrm{FI}}$, the operational and denotational semantics based on interleaving. They will be connected by the following abstraction operator.

Definition 8.1. The operator $\beta_{\mathrm{FI}}: Subst_\delta \to P_{\mathrm{FI}} \to P_{\mathrm{ST}}$ is defined by $\beta_{\mathrm{FI}}(\delta)(X) = \{\lambda\}$, and for $\theta \neq \delta$, by

$$\beta_{\mathrm{FI}}(\theta)(\{\lambda\}) = \{\lambda\}$$

$$\beta_{\mathrm{FI}}(\theta)(X) = \bigcup \{\theta_1 \cdot \beta_{\mathrm{FI}}(\theta_1)(X_{(\theta,\theta_1)}) : X_{(\theta,\theta_1)} \neq \emptyset\}.$$

Here $X_{(\theta,\theta_1)}$ is defined by $X_{(\theta,\theta_1)} = \{\langle v_1, v_2, \ldots\rangle : \langle(\theta, \theta_1)\cdot v_1, v_2, \ldots\rangle \in X\}$.

(The well-definedness of β_{FI} can be established in the by now familiar way: it can be given as the fixed-point of a contraction.) The abstraction operator β_{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 $\mathcal{O}_{\mathrm{FI}}$ and $\mathcal{D}_{\mathrm{FI}}$. (Recall that ε is the empty substitution.)

Theorem 8.2. *For every goal* $\leftarrow \bar{A}$ *we have* $\beta_{\mathrm{FI}}(\varepsilon)\circ\mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}]\!] = \mathcal{O}_{\mathrm{FI}}[\![\leftarrow \bar{A}]\!]$.

Proof. We prove $\beta_{\mathrm{FI}}(\varepsilon)\circ\mathcal{D}_{\mathrm{FI}} = \mathcal{O}_{\mathrm{FI}}$ by showing that $\beta_{\mathrm{FI}}(\varepsilon)\circ\mathcal{D}_{\mathrm{FI}}$ is a fixed-point of the contraction Φ_{FI}. Then the equality follows from Banach's theorem. We omit the deadlock case, which can be taken care of straightforwardly.

The steps marked with (1) and (2) are explained below.

$$\begin{aligned}
&\Phi_{\mathrm{FI}}(\beta_{\mathrm{FI}}(\varepsilon)\circ\mathcal{D}_{\mathrm{FI}})[\![\leftarrow A, \bar{A}]\!]\\
&= \bigcup \{\theta\cdot(\theta \rightsquigarrow_{\mathrm{FI}} (\beta_{\mathrm{FI}}(\varepsilon)\circ\mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}\theta, \bar{B}\theta]\!])) :\\
&\qquad H \leftarrow \bar{B} \in W \text{ and } \theta = mgu(A, H)\}\\
&\overset{(1)}{=} \bigcup \{\theta\cdot(\beta_{\mathrm{FI}}(\theta)\circ\mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}, \bar{B}]\!]) : H \leftarrow \bar{B} \in W \text{ and } \theta = mgu(A, H)\}\\
&= \beta_{\mathrm{FI}}(\varepsilon)(\bigcup \{\langle(\varepsilon, \theta)\rangle\cdot(\mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}]\!] \,\|_{\mathrm{FI}}\, \mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{B}]\!]) :\\
&\qquad\qquad H \leftarrow \bar{B} \in W \text{ and } \theta = mgu(A, H)\})\\
&\overset{(2)}{=} \beta_{\mathrm{FI}}(\varepsilon)(\bigcup\{\langle(\varepsilon, \theta)\rangle\cdot\mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{B}]\!] \,\|_{\mathrm{FI}}\, \mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}]\!] :\\
&\qquad\qquad H \leftarrow \bar{B} \in W \text{ and } \theta = mgu(A, H)\})\\
&= \beta_{\mathrm{FI}}(\varepsilon)(\bigcup\{\langle(\varepsilon, \theta)\rangle\cdot\mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{B}]\!] :\\
&\qquad\qquad H \leftarrow \bar{B} \in W \text{ and } \theta = mgu(A, H)\} \,\|_{\mathrm{FI}}\, \mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}]\!])\\
&= \beta_{\mathrm{FI}}(\varepsilon)(\mathcal{D}_{\mathrm{FI}}[\![\leftarrow A]\!] \,\|_{\mathrm{FI}}\, \mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}]\!])\\
&= \beta_{\mathrm{FI}}(\varepsilon)\circ\mathcal{D}_{\mathrm{FI}}[\![\leftarrow A, \bar{A}]\!].
\end{aligned}$$

Step (1). The identity $\theta \rightsquigarrow_{\mathrm{FI}} \beta_{\mathrm{FI}}(\varepsilon)\circ\mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}\theta, \bar{B}\theta]\!] = \beta_{\mathrm{FI}}(\theta)\circ\mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}, \bar{B}]\!]$ is justified by the following observations. Let $\langle v_1, \ldots\rangle \in \mathcal{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}, \bar{B}]\!]$ be a connected sequence with its first pair of the form (θ, θ'), for some θ'. It follows that $v_1 = \langle(\theta, \theta\theta_1), \ldots, (\theta\theta_1 \cdots \theta_{n-1}, \theta\theta_1 \cdots \theta_n)\rangle$, with $\theta_i = mgu(A_i\theta\theta_1 \cdots \theta_{i-1}, H_i)$, for some $H_i \leftarrow \bar{B}_i$. Here we have $A_1, \ldots A_n = \bar{A}, \bar{B}$. So for $v_1' = \langle(\varepsilon, \theta_1), \ldots, (\theta_1, \cdots \theta_{n-1}, \theta_1 \cdots \theta_n)\rangle$ there exists a sequence $\langle v_1', \ldots\rangle \in \mathscr{D}_{\mathrm{FI}}[\![\leftarrow \bar{A}\theta, \bar{B}\theta]\!]$. Now each pair occurring in $\langle v_2, \ldots\rangle \in \mathscr{D}_{\mathrm{FI}}[\![\leftarrow \bar{B}_1, \ldots, \bar{B}_n]\!]$ is of the form $(\theta\theta', \theta\theta'\theta'')$, where $\theta'' = mgu(B\theta\theta', H)$, for some atoms B and H. But due to the renaming mechanism, which we implicitly assume, we have that θ does not affect the variables of B. So we have that $\theta'' = mgu(B\theta', H)$ implying that we can eliminate θ from the sequence $\langle v_2, \ldots\rangle$. This argument could be formalized by the introduction of an explicit renaming mechanism.

Step (2). We show that $\beta_{\mathrm{FI}}(\varepsilon)(\langle(\varepsilon, \theta)\rangle \cdot (X \parallel_{\mathrm{FI}} Y)) = \beta_{\mathrm{FI}}(\varepsilon)(\langle(\varepsilon, \theta)\rangle \cdot Y \parallel_{\mathrm{FI}} X)$. (For convenience, we write $\langle v_n\rangle_n$ for $\langle v_1, v_2, \ldots\rangle$.)

$$
\begin{aligned}
&\beta_{\mathrm{FI}}(\varepsilon)(\langle(\varepsilon, \theta)\rangle \cdot (X \parallel_{\mathrm{FI}} Y))\\
&\quad = \beta_{\mathrm{FI}}(\varepsilon)(\{\langle(\varepsilon, \theta)\rangle \cdot (\langle w_n\rangle_n \parallel_{\mathrm{FI}} \langle v_n\rangle_n):\\
&\qquad \langle w_n\rangle_n \in X, \langle v_n\rangle_n \in Y\})\\
&\quad = \text{(from Definition 5.1 and Definition 8.1)}\\
&\qquad \beta_{\mathrm{FI}}(\varepsilon)(\{\langle(\varepsilon, \theta) w_1, v_1 w_2, \ldots\rangle : \langle w_n\rangle_n \in X, \langle v_n\rangle_n \in Y\})\\
&\quad = \beta_{\mathrm{FI}}(\varepsilon)(\{\langle(\varepsilon, \theta), v_1, \ldots\rangle \parallel_{\mathrm{FI}} \langle w_1, \ldots\rangle : \langle w_n\rangle_n \in X, \langle v_n\rangle_n \in Y\})\\
&\quad = \beta_{\mathrm{FI}}(\varepsilon)(\langle(\varepsilon, \theta)\rangle \cdot Y \parallel_{\mathrm{FI}} X). \quad \square
\end{aligned}
$$

8.1.2. *Relating $\mathscr{O}_{\mathrm{MP}}$ and $\mathscr{D}_{\mathrm{MP}}$*

Next we prove the identity of the operational model $\mathscr{O}_{\mathrm{MP}}$ and the denotational model $\mathscr{D}_{\mathrm{MP}}$ for maximal parallelism.

Theorem 8.3. $\mathscr{O}_{\mathrm{MP}} = \mathscr{D}_{\mathrm{MP}}$.

Proof. Similarly to the proof of Theorem 8.2, it can be shown that $\mathscr{D}_{\mathrm{MP}}$ is a fixed-point of the contraction Φ_{MP}, from which the theorem follows. $\square$

8.2. *Relating $\mathscr{D}_{\mathrm{FI}}$ and $\mathscr{D}_{\mathrm{MP}}$*

In order to relate $\mathscr{D}_{\mathrm{FI}}$ and $\mathscr{D}_{\mathrm{MP}}$, we introduce an intermediate semantics $\mathscr{I} : Goal \to P_{\mathrm{I}}$, with $P_{\mathrm{I}} = \mathscr{P}_{\mathrm{nco}}((Subst_\delta^+)^\infty)$, as the fixed-point of the contraction $\Psi : (Goal \to P_{\mathrm{I}}) \to (Goal \to P_{\mathrm{I}})$ defined as follows.

Definition 8.4. We define

$$
\begin{aligned}
&\Psi(F)[\![\square]\!] = \{\varepsilon\},\\
&\Psi(F)[\![\leftarrow A]\!] = \bigcup \{\langle mgu(A, H)\rangle \cdot F(\bar{B}) : H \leftarrow \bar{B} \in W\}\\
&\qquad\qquad\quad \bigcup \{\delta : \forall H \leftarrow \bar{B} \in W mgu(A, H) \text{ does not exist}\},\\
&\Psi(F)[\![\leftarrow \bar{A}_1, \bar{A}_2]\!] = \Psi(F)[\![\leftarrow \bar{A}_1]\!] \parallel \Psi(F)[\![\leftarrow \bar{A}_2]\!].
\end{aligned}
$$

Here $\parallel$ is defined in a similar way as $\parallel_{\mathrm{FI}}$.

Now $\mathscr{D}_{\mathrm{FI}}$ and $\mathscr{I}$ are related by the following abstraction operator.

Definition 8.5. We define $\alpha : P_{\mathrm{FI}} \to P_{\mathrm{I}}$ by

$$\alpha(X) = \{\langle\theta_1 \cdots \theta_k, \theta_{k+1} \cdots \theta_l, \cdots\rangle : \langle(\varepsilon, \theta_1) \cdots (\varepsilon, \theta_k), \\ (\varepsilon, \theta_{k+1}) \cdots (\varepsilon, \theta_l), \cdots\rangle \in X\}.$$

(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 ε (the empty substitution) as the first element.

Theorem 8.6. $\mathscr{I} = \alpha \circ \mathscr{D}_{\mathrm{FI}}$.

Proof. It can be shown that $\alpha \circ \mathscr{D}_{\mathrm{FI}}$ is a fixed-point of Ψ. □

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

Definition 8.7. We define $\alpha_{\mathrm{MP}}: P_{\mathrm{I}} \to P_{\mathrm{ST}}$ by

$$\alpha_{\mathrm{MP}}(\langle s_1, s_2, \ldots\rangle) = (\hat{\circ} s_1) \cdot (\hat{\circ}(s_1 \cdot s_2)) \cdots,$$

where $s_i \in \mathit{Subst}_\delta^+$ and $\hat{\circ}\, \theta_1 \cdots \theta_n = \theta_1 \hat{\circ} \cdots \hat{\circ}\, \theta_n$. (If $n = 1$ then $\hat{\circ}\, \theta = \theta$.)

This operator takes for each word $\theta_1 \cdots \theta_n \in \mathit{Subst}_\delta^+$ 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 $\rightsquigarrow_{\mathrm{MP}}$ operator in the definition of $\mathscr{D}_{\mathrm{MP}}$. Now we can establish the following theorem.

Theorem 8.8. $\mathscr{D}_{\mathrm{MP}} = \alpha_{\mathrm{MP}} \circ \mathscr{I}$.

Proof. Again it can easily be shown that $\alpha_{\mathrm{MP}} \circ \mathscr{I}$ is a fixed-point of Ψ_{MP}. □

Combining the two above theorems yields the following corollary.

Corollary 8.9. $\mathscr{D}_{\mathrm{MP}} = \alpha_{\mathrm{MP}} \circ \alpha \circ \mathscr{D}_{\mathrm{FI}}$.

8.3. Relating $\mathscr{D}_{\mathrm{MP}}$ and $\mathscr{D}\mathit{ec}$: an intermediate model $\mathscr{D}_{\mathrm{CS}}$

We introduce an intermediate denotational semantics $\mathscr{D}_{\mathrm{CS}}$ (CS is an abbreviation for computed substitutions), to which both $\mathscr{D}_{\mathrm{MP}}$ and $\mathscr{D}\mathit{ec}$ will be related. It can be seen as a denotational variant of $\mathscr{D}\mathit{ec}$, 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 $\mathscr{D}_{\mathrm{MP}}$ it is a model for maximal parallelism. Formally, $\mathscr{D}_{\mathrm{CS}}$ 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 $P_{\mathrm{CS}} = Goal \to \mathcal{P}(Subst)$.

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

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

$$\left(\bigcup_{F \in \mathcal{F}}\right)[\![\leftarrow \bar{A}]\!] = \bigcup_{F \in \mathcal{F}} F[\![\leftarrow \bar{A}]\!].$$

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

$$X \hat{\circ} Y = \{\theta \hat{\circ} \sigma : \theta \in X, \sigma \in Y, \theta \hat{\circ} \sigma \neq \delta\}.$$

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* $\mathcal{P}(Subst)$ $(\forall k[X_k \subseteq X_{k+1} \wedge Y_k \subseteq Y_{k+1}])$. *Then* $\bigcup_{k \geq 0}(X_k \hat{\circ} Y_k) = (\bigcup_{m \geq 0} X_m) \hat{\circ} (\bigcup_{n \geq 0} Y_n)$.

Next we introduce $\mathcal{D}_{\mathrm{CS}}: Goal \to \mathcal{P}(Subst)$.

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

$$\Psi_{\mathrm{CS}}(F)[\![\square]\!] = \{\varepsilon\},$$

$$\Psi_{\mathrm{CS}}(F)[\![\leftarrow A]\!] = \bigcup \{(mgu(A, H) \hat{\circ} F(\leftarrow \bar{B}))_{|Var(A)} : H \leftarrow \bar{B} \in W\},$$

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

The continuity of Ψ_{CS} is a direct consequence of Lemma 8.11.

8.4. Relating $\mathcal{D}_{\mathrm{MP}}$ and $\mathcal{D}_{\mathrm{CS}}$

The relation between the models $\mathcal{D}_{\mathrm{MP}}$ and $\mathcal{D}_{\mathrm{CS}}$ is described by the abstraction operator $\alpha_{\mathrm{CS}}: P_{\mathrm{ST}} \to \mathcal{P}(Subst)$ defined by

$$\alpha_{\mathrm{CS}}(X) = last(X \cap Subst^{+}).$$

(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. $\mathcal{D}_{CS} = \alpha_{CS} \circ \mathcal{D}_{MP}$.

The theorem is immediate from the following two lemmas, which can be proved by induction on *n*. Let the functions $\perp$ and f_{ε^ω} be defined by $\perp(\leftarrow \bar{A}) = \emptyset$ and $f_{\varepsilon^\omega}(\leftarrow \bar{A}) = \{\varepsilon^\omega\}$, for all $\leftarrow \bar{A}$.

Lemma 8.14. *For all n*: $\Psi^n_{CS}(\perp) = (\alpha_{CS} \circ \Psi^n_{MP})(f_{\varepsilon^\omega})$.

Lemma 8.15. *For all n and* $\leftarrow \bar{B}$:

$$(\alpha_{CS} \circ \Psi^m_{MP})(f_{\varepsilon^\omega})(\leftarrow \bar{B}) \subseteq (\alpha_{CS} \circ \Psi^{n+1}_{MP})(f_{\varepsilon^\omega})(\leftarrow \bar{B}).$$

Proof of Theorem 8.13. For any $\leftarrow \bar{B}$ we have

$$\begin{aligned}(\alpha_{CS} \circ \mathcal{D}_{MP})(\leftarrow \bar{B}) &= \alpha_{CS}(\lim_{n\to\infty} \Psi^n_{MP}(f_{\varepsilon^\omega})(\leftarrow \bar{B}))\\ &= (\text{Lemma 8.15}) \quad \alpha_{CS}(\bigcup \Psi^n_{MP}(f_{\varepsilon^\omega})(\leftarrow \bar{B}))\\ &= \bigcup_n \alpha_{CS}(\Psi^n_{MP}(f_{\varepsilon^\omega})(\leftarrow \bar{B}))\\ &= (\text{Lemma 8.14}) \quad \bigcup_n \Psi^n_{CS}(\perp)(\leftarrow \bar{B})\\ &= \mathcal{D}_{CS}(\leftarrow \bar{B}). \quad \square\end{aligned}$$

8.5. Relating $\mathcal{D}_{CS}$ and $\mathcal{D}ec$

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, ν and μ and by showing that $\mathcal{D}ec = \nu(\mathcal{D}_{CS})$ and $\mathcal{D}_{CS} = \mu(\mathcal{D}ec)$. 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 Ψ_{CS} such that μ and ν make *T* and Ψ_{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 ν allows to *simulate* step by step the fixed-point construction of Ψ_{CS} by *T* and vice-versa, namely: for each $n \geq 0$, $T^n(\emptyset) = \nu(\Psi^n_{CS}(F_0))$ and $\Psi^n_{CS}(F_0) = \mu(T^n(\emptyset))$ (where F_0 is the minimal element of *P*). Finally, by continuity of ν and μ, we can commute also the least upper bounds of these chains, so that $lfp(T) = \bigcup_{n\geq 0} T^n(\emptyset) = \nu(\bigcup_{n\geq 0} \Psi^n_{CS}(F_0)) = \nu(lfp(\Psi_{CS}))$ and $lfp(\Psi_{CS}) = \bigcup_{n\geq 0} \Psi^n_{CS}(F_0) = \mu(\bigcup_{n\geq 0} T^n(\emptyset)) = \mu(lfp(T))$.

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

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

(R1) $F[\![\square]\!] = \{\varepsilon\}$
(R2) $\forall\theta\,[Dom(\theta) \subseteq \bar{x} \Rightarrow (\theta \hat{\circ} F[\![\leftarrow p(\bar{x})]\!])_{|p(\bar{x})\theta} = F[\![\leftarrow p(\bar{x})\theta]\!]]$
(R3) $F[\![\leftarrow \bar{A}_1, \bar{A}_2]\!] = F[\![\leftarrow \bar{A}_1]\!] \hat{\circ} F[\![\leftarrow \bar{A}_2]\!]$
(R4) $\forall \bar{A}\, \forall\theta \in F[\![\leftarrow \bar{A}]\!]\,[Dom(\theta) \subseteq Var(\bar{A})]$

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 $(F_n)_n$ is given by $\bigcup_n F_n$. In other words, $(P, \subseteq)$ is a sub CPO of $(P_{CS}, \subseteq)$.*

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

Definition 8.18.

- The function $\nu : P \to P_{\mathrm{Dec}}$ is defined by

$$\nu(F) = \{p(\bar{x})\theta : \theta \in F[\![\leftarrow p(\bar{x})]\!] \wedge p(\bar{x}) \in EAtom\}.$$

- The function $\mu : P_{\mathrm{Dec}} \to P$ is defined by

$$\mu(I)[\![\square]\!] = \{\varepsilon\},$$

$$\mu(I)[\![\leftarrow A]\!] = \{mgu(A, A')_{|A} : A' \in I \wedge Var(\bar{A}') \cap Var(\bar{A}) = \emptyset\},$$

$$\mu(I)[\![\leftarrow \bar{A}_1, \bar{A}_2]\!] = \mu(I)[\![\leftarrow \bar{A}_1]\!] \hat{\circ} \mu(I)[\![\leftarrow \bar{A}_2]\!].$$

Remark 8.19. The function μ is well defined, i.e., $\forall I \in P_{\mathrm{Dec}}[\mu(I) \in P]$. Indeed, (R1), (R3) and (R4) are trivial, and (R2) is an immediate consequence of the following lemma.

Lemma 8.20. *Let θ be an idempotent substitution, and assume $Dom(\theta) \subseteq \bar{x}$. Let A be an atom such that $Var(A) \cap \{\bar{x}\} = \emptyset$ and $Var(A) \cap Var(\{\bar{x}\theta\}) = \emptyset$. Then*

$$(\theta \hat{\circ} mgu(p(\bar{x}), A))_{|p(\bar{x})\theta} = mgu(p(\bar{x})\theta, A)_{|p(\bar{x})\theta}.$$

Proof. The proof uses some elementary properties of idempotent substitutions (see [13]). □

The following facts can be readily established.

Proposition 8.21. *The functions ν and μ are continuous.*

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

The following result shows that μ and ν commute the functions Ψ_{CS} and T on P.

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

The functions ν and μ allow to simulate, step by step, the fixed-point construction of $\mathcal{D}_{CS}$ in $\mathcal{Dec}$, and vice-versa. There is only one difficulty: the fixed-point construction of Ψ_{CS} starts from the minimal element of P_{CS}, that is the function $F_\perp$ such that for every $\bar{A}$, $F_\perp[\![\leftarrow \bar{A}]\!] = \emptyset$. Unfortunately, $F_\perp$ is not the minimal element of P, in fact $F_\perp \notin P$. The minimal element of P is the function F_0 such that

$$F_0[\![\leftarrow \bar{A}]\!] = \begin{cases} \{\varepsilon\} & \text{iff } \leftarrow \bar{A} = \square, \\ \emptyset & \text{otherwise.} \end{cases}$$

However, the fixed-point of Ψ_{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 $F_0 = \Psi_{CS}(F_\perp)$.

Lemma 8.25.
(1) $\forall n \geq 0\ [T^n(\nu(F_0)) = \nu(\Psi^n_{CS}(F_0))]$.
(2) $\forall n \geq 0\ [\Psi^n_{CS}(\mu(\emptyset)) = \mu(T^n(\emptyset))]$.

Proof. By induction on n.

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

Theorem 8.26.
(1) $\mathcal{Dec} = \nu(\mathcal{D}_{CS})$.
(2) $\mathcal{D}_{CS} = \mu(\mathcal{Dec})$.

Proof.

(1)
$$\begin{aligned}\mathcal{D}ec &= lfp(T)\\ &= \bigcup_{n\geq 0} T^n(\emptyset)\\ &= \bigcup_{n\geq 0} T^n(\nu(F_0)) \quad \text{(by Remark 8.24, part 1)}\\ &= \bigcup_{n\geq 0} \nu(\Psi^n_{\mathrm{CS}}(F_0)) \quad \text{(by Lemma 8.25, part 1)}\\ &= \nu\Big(\bigcup_{n\geq 0} \Psi^n_{\mathrm{CS}}(F_0)\Big) \quad \text{(by continuity of } \nu)\\ &= \nu(lfp(\Psi_{\mathrm{CS}})) \quad \text{(by Remark 8.24, part 2)}\\ &= \nu(\mathcal{D}_{\mathrm{CS}}).\end{aligned}$$

(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.

$$\mathcal{O}_{\mathrm{FI}} = \beta_{\mathrm{FI}} \circ \mathcal{D}_{\mathrm{FI}}.$$

$$\mathcal{I} = \alpha \circ \mathcal{D}_{\mathrm{FI}}.$$

$$\mathcal{O}_{\mathrm{MP}} = \mathcal{D}_{\mathrm{MP}} = \alpha_{\mathrm{MP}} \circ \mathcal{I}.$$

$$\mathcal{D}_{\mathrm{CS}} = \alpha_{\mathrm{CS}} \circ \mathcal{D}_{\mathrm{MP}}.$$

$$\mathcal{D}_{\mathrm{CS}} = \mu(\mathcal{D}ec).$$

$$\mathcal{D}ec = \nu(\mathcal{D}_{\mathrm{CS}}).$$

In Fig. 1, these equalities are graphically represented. Moreover, it contains some arrows between $\mathcal{O}_{\mathrm{FI}}$ and the sets $\mathcal{O}_{\mathrm{SS}}$, $\mathcal{O}_{\mathrm{FFS}}$ and $\mathcal{O}_{\mathrm{IFS}}$, indicating that the definition of these sets is based on that of $\mathcal{O}_{\mathrm{FI}}$.

Combining some of the equalities above, we find

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

a maybe somewhat complicated but precise relationship between the declarative semantics $\mathcal{D}ec$ and the denotational semantics $\mathcal{D}_{\mathrm{FI}}$. 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 [9]. The fact that here the relationship between $\mathcal{D}ec$ and $\mathcal{D}_{\mathrm{FI}}$ and, hence, between $\mathcal{D}ec$ and $\mathcal{O}_{\mathrm{SS}}$ has been decomposed into several steps makes the proof below more transparent.

$$\begin{tikzcd}
\mathcal{O}_{\mathrm{SS}} & \mathcal{O}_{\mathrm{FFS}} & & & \\
\mathcal{O}_{\mathrm{FI}} \arrow[u] \arrow[ur] \arrow[r] & \mathcal{O}_{\mathrm{IFS}} & & & \\
& & \mathcal{O}_{\mathrm{MP}} \arrow[d, equal] & & \\
\mathcal{D}_{\mathrm{FI}} \arrow[uu, "\beta_{\mathrm{FI}}"'] \arrow[r, "\alpha"] & \mathcal{J} \arrow[r, "\alpha_{\mathrm{MP}}"] & \mathcal{D}_{\mathrm{MP}} \arrow[r, "\alpha_{\mathrm{CS}}"] & \mathcal{D}_{\mathrm{CS}} \arrow[r, shift left, "\nu"] & \mathcal{D}ec \arrow[l, shift left, "\mu"]
\end{tikzcd}$$

Fig. 1.

Theorem 9.1 $A \in \mathcal{O}_{\mathrm{SS}} \Leftrightarrow A \in \mathcal{D}ec$.

Proof.

$$\begin{aligned}
A \in \mathcal{O}_{\mathrm{SS}} &\Leftrightarrow (\text{definition } \mathcal{O}_{\mathrm{SS}})\ \exists p(\bar{x}) \exists \theta_1 \cdots \theta_n \in \mathcal{O}_{\mathrm{FI}}[\![\leftarrow p(\bar{x})]\!]\colon A = p(\bar{x})\theta_n \\
&\Leftrightarrow (\mathcal{O}_{\mathrm{FI}} = \beta_{\mathrm{FI}} \circ \mathcal{D}_{\mathrm{FI}}) \exists p(\bar{x}) \exists s_1, \ldots, s_k \in \mathcal{D}_{\mathrm{FI}}[\![\leftarrow p(\bar{x})]\!]\colon \\
&\qquad s_1 \cdots s_k = \langle (\varepsilon, \theta_1), (\theta_1, \theta_2), (\theta_2, \theta_3), \ldots, (\theta_{n-1}, \theta_n) \rangle \wedge A = p(\bar{x})\theta_n \\
&\Leftrightarrow (\text{using } \theta mgu(A\theta, H) = \theta \hat{\circ}\, mgu(A, H), \\
&\qquad \text{a direct consequence of Lemma 4.4}) \\
&\qquad \exists p(\bar{x}) \exists s_1, \ldots, s_k \in \mathcal{D}_{\mathrm{FI}}[\![\leftarrow p(\bar{x})]\!]\colon \\
&\qquad\quad s_1 \cdots s_k = \langle (\varepsilon, \theta_1), (\varepsilon, \theta_2), \ldots, (\varepsilon, \theta_n) \rangle \wedge A = p(\bar{x})(\theta_1 \hat{\circ} \cdots \hat{\circ}\, \theta_n) \\
&\Leftrightarrow (\mathcal{J} = \alpha \circ \mathcal{D}_{\mathrm{FI}}) \\
&\qquad \exists p(\bar{x}) \exists \langle v_1, \ldots, v_k \rangle \in \mathcal{J}[\![\leftarrow p(\bar{x})]\!]\colon \\
&\qquad\quad v_1 \cdots v_k = \theta_1 \cdots \theta_n \wedge A = p(\bar{x})(\theta_1 \hat{\circ} \cdots \hat{\circ}\, \theta_n) \\
&\Leftrightarrow (\mathcal{D}_{\mathrm{MP}} = \alpha_{\mathrm{MP}} \circ \mathcal{J}) \exists p(\bar{x}) \exists \theta_1 \cdots \theta_n \in \mathcal{D}_{\mathrm{MP}}[\![\leftarrow p(\bar{x})]\!]\colon A = p(\bar{x})\theta_n \\
&\Leftrightarrow (\mathcal{D}_{\mathrm{CS}} = \alpha_{\mathrm{CS}} \circ \mathcal{D}_{\mathrm{MP}}) \exists p(\bar{x}) \exists \theta \in \mathcal{D}_{\mathrm{CS}}[\![\leftarrow p(\bar{x})]\!]\colon A = p(\bar{x})\theta \\
&\Leftrightarrow A \in \mathcal{D}ec. \qquad \square
\end{aligned}$$

We deduce from the equalities above a second fact, which says that $\mathcal{O}_{\mathrm{SS}}$, $\mathcal{O}_{\mathrm{FFS}}$ and $\mathcal{O}_{\mathrm{IFS}}$ can be characterized in terms of $\mathcal{O}_{\mathrm{MP}}$ ($=\mathcal{D}_{\mathrm{MP}}$), instead of $\mathcal{O}_{\mathrm{FI}}$. 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 not seem very surprising, it is not completely straightforward, since $\mathcal{O}_{\mathrm{MP}}$ and $\mathcal{O}_{\mathrm{FI}}$ have a different deadlock behavior: the former delivers deadlock for more goals than the latter. (See the counterexample at the end of Section 4.1.)

Theorem 9.2. *We have the following equalities.*

$$\mathcal{O}_{\mathrm{SS}} = \{ p(\bar{x})\theta : p(\bar{x}) \in EAtom \wedge \theta \in last(\mathcal{O}_{\mathrm{MP}}[\![\leftarrow p(\bar{x})]\!] \cap Subst^{+})\}.$$

$$\mathcal{O}_{\mathrm{FFS}} = \{ A : \mathcal{O}_{\mathrm{MP}}[\![\leftarrow A]\!] \subseteq Subst^{*} \cdot \delta\}.$$

$$\mathcal{O}_{\mathrm{IFS}} = \{ A : \mathcal{O}_{\mathrm{MP}}[\![\leftarrow A]\!] \cap Subst^{*} = \emptyset \wedge \mathcal{O}_{\mathrm{MP}}[\![\leftarrow A]\!] \cap Subst^{\omega} \neq \emptyset\}.$$

Proof. Similar to that of the previous theorem. □

Acknowledgment

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

References

[1] K.R. Apt, Introduction to logic programming, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol. B* (Elsevier, Amsterdam, 1990) 493-574.

[2] K.R. Apt and M.H. van Emden, Contributions to the theory of logic programming, *J. ACM* **29**(3) (1982) 841-862.

[3] J.W. de Bakker and J.I. Zucker, Processes and the denotational semantics of concurrency, *Inform. and Control* **54** (1982) 70-120.

[4] F.S. de Boer, J.N. Kok, C. Palamidessi and J.J.M.M. Rutten, Control flow versus logic: a denotational and a declarative model for guarded Horn clauses, in: A. Kreczmar and G. Mirkowska, eds., *Proc. Mathematical Foundations of Computer Science (MFCS '89)*, Lecture Notes in Computer Science, Vol. 379 (Springer, Berlin, 1989) 165-177.

[5] F.S. de Boer, J.N. Kok, C. Palamidessi and J.J.M.M. Rutten, Semantic models for a version of Parlog, in: G. Levi and M. Martelli, eds., *Proc. Internat. Conf. on Logic Programming (ICLP 89)*, pages 621-636. MIT Press, 1989; also *Theoret. Comput. Sci.* **86** (1991) 3-33.

[6] E.P. de Vink, Concurrency semantics applied to logic programming, Technical report, Vrije Universiteit, Amsterdam, 1990.

[7] R. Engelking, *General Topology* (Polish Scientific Publishers, Warsaw, 1977).

[8] M. Falaschi, G. Levi, C. Palamidessi and M. Martelli, A new declarative semantics for logic languages, in: *Proc. Conf. and Symp. on Logic Programming* (MIT Press, Cambridge, MA, 1988) 993-1005.

[9] M. Falaschi, G. Levi, C. Palamidessi and M. Martelli, Declarative modeling of the operational behavior of logic languages, *Theoret. Comput. Sci.* **69**(3) (1989) 289-318.

[10] M. Hennessy and G.D. Plotkin, Full abstraction for a simple parallel programming language, in: J. Bečvar, ed., *Proc. Mathematical Foundations of Computer Science (MFCS '79)*, Lecture Notes in Computer Science, Vol. 74 (Springer, Berlin, 1979) 108-120.

[11] J.-M. Jacquet, Conclog: A methodological approach to concurrent logic programming, Ph.D. thesis, Facultés Universitaires Notre Dame de la Paix, Namur, 1989.

[12] J.W. Lloyd, *Foundations of Logic Programming* (Springer, Berlin, 2nd edn., 1987).

[13] C. Palamidessi, Algebraic properties of idempotent substitutions, in: M.S. Paterson, ed., *Proc. 17th Internat. Colloq. on Automata, Languages and Programming*, Lecture Notes in Computer Science,

Vol. 443 (Springer, Berlin, 1990) 386-399; full version available as Technical Report TR-33/89, Dipartimento di Informatica, Università di Pisa.

[14] E.Y. Shapiro, A subset of Concurrent Prolog and its interpreter, Technical Report TR-003, ICOT, Tokyo, 1983.

[15] M.H. van Emden and R.A. Kowalski, The semantics of predicate logic as a programming language, *J. ACM* **23**(4) (1976) 733-742.