INFORMATION AND COMPUTATION 115, 125–178 (1994)

Fully Abstract Denotational Models for Nonuniform Concurrent Languages

E. HORITA

Centre for Mathematics and Computer Sciences, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands; and NTT Software Laboratories, 3-9-11 Midori-Cho, Musashino-Shi, Tokyo 180, Japan

J. W. DE BAKKER*

Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands; and Departments of Mathematics and Computer Science, Free University of Amsterdam, The Netherlands

AND

J. J. M. M. RUTTEN*

Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

This paper investigates *full abstraction* of denotational model *w.r.t.* operational ones for two concurrent languages. The languages are *nonuniform* in the sense that the meaning of atomic statements generally depends on the current state. The first language, $\mathscr{L}_1$, has *parallel composition* but no communication, whereas the second one, $\mathscr{L}_2$, has CSP-like *communications* in addition. For each of $\mathscr{L}_i$ $(i = 1, 2)$, an operational model $\mathscr{O}_i$ is introduced in terms of a Plotkin-style transition system, while a denotational model $\mathscr{D}_i$ for $\mathscr{L}_i$ is defined compositionally using interpreted operations of the language, with meanings of recursive programs as fixed points in appropriate complete metric spaces. The full abstraction is shown by means of a context with parallel composition:

> Given two statements s_1 and s_2 with different denotational meanings, a suitable statement T is constructed such that the operational meanings of $s_1 \parallel T$ and $s_2 \parallel T$ are distinct.

A combinatorial method for constructing such T is proposed. Thereby the full abstraction of $\mathscr{D}_1$ and $\mathscr{D}_2$ w.r.t. $\mathscr{O}_1$ and $\mathscr{O}_2$, respectively, is established. That is, $\mathscr{D}_i$ is most abstract of those models $\mathscr{C}$ which are compositional and satisfy $\mathscr{O}_i = \alpha \circ \mathscr{C}$ for some abstraction function α $(i = 1, 2)$. © 1994 Academic Press, Inc.

* Partially supported by ESPRIT Basic Research Action (3020) Integration.

0890-5401/94 $6.00

1. Introduction

This paper investigates *full abstraction* of denotational models w.r.t. operational ones for two concurrent languages. The languages are *nonuniform* in the sense that the meaning of atomic statements generally depends on the current state. In particular, they have individual variables which store values, and the elementary actions are (mainly) value assignments to these variables. The first language, $\mathscr{L}_1$, has *parallel composition* but not communication, whereas the second one, $\mathscr{L}_2$, has CSP-like *communications* in addition. Both of the two languages have recursion. For each of $\mathscr{L}_i$ $(i=1,2)$, an operational model $\mathcal{O}_i$ is introduced in terms of a Plotkin-style transition system, while a denotational model $\mathscr{D}_i$ is defined compositionally using interpreted operations of the language and some fixed point method for defining the meanings of recursive programs.

We show that, with the nonuniform languages, one needs to represent, in the meaning of a process, possible interactions between the process and its environment. Merely recording observations of initial and final states or possible computation sequences is not enough to obtain compositionality. One needs sequences in which there are *gaps* between steps to represent possible actions of the environment. This is essential in order to interpret parallel composition compositionally. Furthermore, the model one obtains by adding this information is in fact *fully abstract* w.r.t. the operational semantics, which is established by showing how to construct contexts that distinguish processes with different meanings.

The *full abstraction problem* for programming languages was first raised by Milner in [Mil73]. In general, a model $\mathscr{D}$ for a language $\mathscr{L}$ is called fully abstract w.r.t. another model $\mathcal{O}$, if it makes *just enough* distinctions to be correct (and thus compositional) w.r.t. $\mathcal{O}$. In other words, it is fully abstract w.r.t. $\mathcal{O}$, if

$$\forall s_1, s_2 \in \mathscr{L}[\mathscr{D}[\![s_1]\!] = \mathscr{D}[\![s_2]\!] \\ \Leftrightarrow \forall C[C \text{ is a } context \text{ of } \mathscr{L} \Rightarrow \mathcal{O}[\![C[s_1]]\!] = \mathcal{O}[\![C[s_2]]\!]]],$$

where a *context* is a statement consisting of the language constructs of $\mathscr{L}$ and a *place-holder* (or a *hole*) ξ, and $C[s]$ denotes the result of substituting s for ξ in C.[1] If $\mathscr{D}$ is fully abstract w.r.t. $\mathcal{O}$, then $\mathscr{D}$ is the most abstract of those models $\mathscr{C}$ which are compositional and satisfy $\mathcal{O} = \alpha \circ \mathscr{C}$ for some abstraction function α; i.e., for each of these $\mathscr{C}$'s, there is an *abstraction function* β such that $\beta \circ \mathscr{C} = \mathscr{D}$. The models $\mathscr{D}_i$ $(i=1,2)$ will be *denotational*

[1] For an operational or denotational model $\mathscr{M}$ for a language $\mathscr{L}$ and a statement $s \in \mathscr{L}$, the notation $\mathscr{M}[\![s]\!]$ is used to denote the value of $\mathscr{M}$ at s.

in the sense that apart from being compositional, they treat infinite behavior by means of some fixed point construction.

The mathematical domains we use are *complete metric spaces* [Niv79, BZ82]. In general, the metric approach may have, as a tool in programming language semantics, some advantages over the use of the more traditional complete partial orders: First, many definitions can be given as the (by Banach's theorem) unique fixed points of some higher-order functions. Second, a metric powerdomain can be easily defined (as the collection of closed or compact subsets of a given complete metric space). In comparison, ordered powerdomains are easily defined as well (by means of ideal completion), but often the characterization of their elements is rather technical. For some example of the application of metric spaces to semantics, see for instance [ABKR89, BM88, Bak91].

In Section 2, some mathematical preliminaries on complete metric spaces, especially on spaces consisting (of sets) of streams, are given; the main body of our paper consists of Sections 3 and 4.

In Section 3, the first language, $\mathscr{L}_1$, is introduced; an operational model $\mathscr{O}_1$ is presented in terms of a Plotkin-style transition system; and a denotational model $\mathscr{D}_1$ for $\mathscr{L}_1$ is defined on the basis of a complete metric space consisting of sets of streams of pairs of states with some additional information. First, the correctness of $\mathscr{D}_1$ w.r.t. $\mathscr{O}_1$ is established, as in [Rut89, BR91], by means of the fixed point method introduced in [KR90]. The full abstraction of $\mathscr{D}_1$ is shown by means of a context with parallel composition:

> Given two statements $s_1, s_2 \in \mathscr{L}_1$ with different denotational meanings, a suitable statement T called a *tester* is constructed such that the operational meanings of $s_1 \parallel T$ and $s_2 \parallel T$ are distinct.[2] (1)

A combinatorial method called the *testing method*, which is the key idea of our paper, is proposed for constructing such a tester (Lemma 13). This is in general applicable to denotational models with a domain consisting of sets of streams of pairs of states (possibly with some additional information). Thereby, we can construct testers having the following property:

> Given a process p and a finite sequence $r = (\langle \sigma_1, \sigma'_1 \rangle, ..., \langle \sigma_n, \sigma'_n \rangle)$, we can construct a tester T and an executable sequence $\tilde{r} = (\langle \tilde{\sigma}_1, \tilde{\sigma}'_1 \rangle, ..., \langle \tilde{\sigma}_k, \tilde{\sigma}'_k \rangle)$ with $k \geqslant n$ such that for every process p', the parallel compositions $p' \tilde{\parallel} \mathscr{D}_1[\![T]\!]$ can execute $\tilde{r}$ if there is some sequence q such that $(\langle \sigma_1, \sigma'_1 \rangle, ...,$

[2] The variable T is used to denote a statement when it is considered a tester, while the typical variable for the set of statements is s.

> $\langle\sigma_n, \sigma'_n\rangle)\cdot q \in p'$, and the converse of this holds for $p' = p$. Intuitively, for such T and $\tilde{r}$, the process p is *forced* to execute the steps $\langle\sigma_1, \sigma'_1\rangle, ..., \langle\sigma_n, \sigma'_n\rangle$ (maybe not consecutively but in this order), when $p \,\tilde{\|}\, \mathscr{D}_1[\![T]\!]$ executes the steps $(\langle\tilde{\sigma}_1, \tilde{\sigma}'_1\rangle, ..., \langle\tilde{\sigma}_k, \tilde{\sigma}'_k\rangle)$ consecutively.

By the above property, we can construct such testers T as in (1):

> If s_1 and s_2 are distinct in their denotational meaning, then, putting $p_i = \mathscr{D}_1[\![s_i]\!]$ $(i = 1, 2)$, there exists some sequence r such that $r \cdot q \in p_1$ for some q but $r \cdot q \notin p_2$ for every q (or vice versa). By constructing a tester T and an executable sequence $\tilde{r}$ for r and $p = p_2$ as above, one has $\tilde{r} \in \mathscr{D}_1[\![s_1]\!] \,\tilde{\|}\, \mathscr{D}_1[\![T]\!]$ and $\tilde{r} \notin \mathscr{D}_1[\![s_2]\!] \,\tilde{\|}\, \mathscr{D}_1[\![T]\!]$. Thus one has a difference between the operational meanings of the two statements $s_1 \parallel T$ and $s_2 \parallel T$.

The full abstraction of $\mathscr{D}_1$ is established by means of the testing method as described above.

In Section 4, the second language, $\mathscr{L}_2$, is introduced; an operational model $\mathscr{O}_2$ for $\mathscr{L}_2$ is given as in Section 3. The domain of a denotational model $\mathscr{D}_2$ for $\mathscr{L}_2$ is a kind of *failures model*, which was introduced in [BHR84], and is adapted here to the nonuniform setting. Each element of the domain is a set consisting of elements that are represented as $\langle(\langle\sigma_i, a_i, \sigma'_i\rangle)_i, \langle\sigma'', \Gamma\rangle\rangle$, where σ_i, σ'_i, and σ'' are states, a_i is an action, and Γ is a set of *communication sorts*. These elements are called *failures*; the parts $(\langle\sigma_i, a_i, \sigma'_i\rangle)_i$ and $\langle\sigma'', \Gamma\rangle$ are called a *trace* and a *refusal*, respectively. First, the correctness of $\mathscr{D}_2$ is established as in Section 3. Then, the full abstraction of $\mathscr{D}_2$ is established by a combination of the testing method and the method proposed by Bergstra *et al.* in [BKO88] to establish the full abstraction of a *failures model* for a uniform language without recursion. This method was adapted by Rutten in [Rut89] to be employed for a language with recursion in the framework of complete metric spaces, which suggests how to use it in the present setting. Given two statements s_1 and s_2 of $\mathscr{L}_2$, which are distinct in their denotational meanings, then the denotational meanings are distinct in the trace parts or in the refusal parts. When the distinction is in the trace parts, we can construct a tester by the method described above; otherwise we can construct a tester by the method of [BKO88].

Finally, in Section 5, some remarks on related and future work are given.

For some mathematical proofs, the reader will be referred to [HBR90].

Closely related to this paper is the work of Hennessy and Plotkin [HP79]. The language treated there, which we denote by $\mathscr{L}_{\text{co}}$, is very similar to our first language, $\mathscr{L}_1$, except that it contains "co", a *coroutine* construct, as well as the usual interleaving. In [HP79], a denotational model $\mathscr{V}$ for $\mathscr{L}_{co}$ is constructed and the full abstraction of $\mathscr{V}$ is established. Interestingly, we can construct a fully abstract model $\mathscr{D}_{co}$ for $\mathscr{L}_{co}$ by slightly modifying $\mathscr{D}_1$; thus the two models $\mathscr{V}$, $\mathscr{D}_{co}$ turn out to be isomorphic (see Section 3.6.3 for more comparison with [HP79]).

The work of Roscoe [Ros84] is also related to this paper. The language treated there, a large subset of occam, is similar to our second language $\mathscr{L}_2$ in many respects. However, unlike individual variables in $\mathscr{L}_2$, variables in occam are not shared by two of more parallel processes. Thus, the model proposed in [Ros84] is different from $\mathscr{D}_2$ in its way of involving states into the meaning of a statement (see Section 4.6 for more comparison with [Ros84]).

2. Mathematical Preliminaries

As mathematical domains for our operational and denotational models, we shall use *complete metric spaces* composed of (sets of) *streams*. In this section, we present some standard notions on complete metric spaces and some notions specific to domains of (sets of) streams.

First, we assume the notions of *metric space*, *ultra-metric space* (or *non-Archimedian metric space*), *complete (ultra-)metric space*, *continuous function*, *closed set*, *contraction*, *nonexpansive mapping*, and *isometry* to be known. The fact that *a contraction from a complete metric space to itself has a unique fixed point*, known as Banach's Theorem, is conveniently used (for the notions and fact above, the reader might consult [Dug66] or [Eng77]). We use the following notation:

Notation 1. (1) The usual λ-notation is used for denoting functions; i.e., for a set A, a variable x, and an expression $E(x)$, the expression $(\lambda x \in A : E(x))$ denotes the function which maps $x \in A$ to $E(x)$. For a set X, the cardinality of X is denoted by $\#(X)$, and the set of nonempty subsets of X and the set of finite subsets of X are denoted by $\wp_+(X)$, and $\wp_f(X)$, respectively. For a binary relation R on X, the *reflexive and transitive closure* of R is denoted by R^*. For two sets X and Y, the set of functions from X to Y is denoted by $(X \to Y)$. The set of natural numbers is denoted by ω. Each number $n \in \omega$ is identified with the set $\{i \in \omega : 0 \leqslant i < n\}$ as usual in set theory, and let $\bar{n} = \{i \in \omega : 1 \leqslant i \leqslant n\}$. The *closure* of a subset X of a topological space M is denoted by X^{cls}.

(2) The empty sequence is denoted by ε. For a nonempty finite sequence q, the last element of q is denoted by last(q). For a set A, the set of finite sequences of elements of A is denoted by $A^{<\omega}$, and let $A^+ = A^{<\omega} \setminus \{\varepsilon\}$. The set of finite or infinite (with length ω) of sequences of

elements of A is denoted by $A^{\leqslant\omega}$. For $a \in A$, we sometimes write simply a to denote the sequence (a) consisting only of a; further, we sometimes write simply A to denote $\{(a) : a \in A\}$. For $q_1 \in A^{<\omega}$, $q_2 \in A^{\leqslant\omega}$, the *concatenation* of q_1 and q_2 is denoted by $q_1 \cdot q_2$. Also for $p_1 \subseteq A^{<\omega}$, $p_2 \subseteq A^{\leqslant\omega}$, let $p_1 \cdot p_2 = \{w_1 \cdot w_2 : w_1 \in p_1 \wedge w_2 \in p_2\}$. For $q \in A^{\leqslant\omega}$, the length of q is denoted by $\mathrm{lgt}(q)$. For $n \in \omega$ and $q \in A^{\leqslant\omega}$, the *truncation* of q at level n, denoted by $q^{[n]}$, is the prefix of q with length n if $\mathrm{lgt}(n) \geqslant n$; otherwise it is q. For $p \subseteq A^{\leqslant\omega}$, let $p^{[n]} = \{q^{[n]} : q \in p\}$. An ordered *pair* $\langle a_0, a_1 \rangle$ and a *triple* $\langle a_0, a_1, a_2 \rangle$ $(= \langle a_0, \langle a_1, a_2 \rangle \rangle)$ are distinguished from, but treated as sequences $(a_i)_{i \in n}$ with n being 2 and 3, respectively; for $n = 2$, 3, we sometimes write $\langle a_i \rangle_{i \in n}$ to denote $\langle a_0, ..., a_{n-1} \rangle$. For $n = 2$, 3 and $i \in n$, the ith component of $t = \langle a_i \rangle_{i \in n}$ is denoted by $\pi_i^n(t)$.

An arbitrary set A can be supplied with a metric d_A, called the *discrete metric*, defined by $d_A(x, y) = 0$ if $x = y$, otherwise $d_A(x, y) = 1$. The space $\langle A, d_A \rangle$ is an ultra-metric space. We use the following operation on metric spaces. (In our definition the distance between two elements of a metric space is always bounded by 1.)

DEFINITION 1 (Operations on Metric Spaces). Let $\langle M, d \rangle$, $\langle M_1, d_1 \rangle$, ..., $\langle M_n, d_n \rangle$ be metric spaces. (1) For a real number κ such that $0 < \kappa < 1$, we define $\mathrm{id}_\kappa(\langle M, d \rangle) = \langle M, d' \rangle$, where $d'(x, y) = \kappa \cdot d(x, y)$, for every $x, y \in M$. (2) Let $M_1 \uplus \cdots \uplus M_n$ denote the *disjoint union* of $M_1, ..., M_n$, which can be defined as $\bigcup_{j \in \bar{n}} [\{j\} \times M_j]$. A metric d_{U} on $M_1 \uplus \cdots \uplus M_n$ is defined as follows: For $\langle i, x \rangle$, $\langle j, y \rangle \in M_1 \uplus \cdots \uplus M_n$, $d_{\mathrm{U}}(\langle i, x \rangle, \langle j, y \rangle) = d_i(x, y)$ if $i = j$; otherwise $d_{\mathrm{U}}(\langle i, x \rangle, \langle j, y \rangle) = 1$. (3) A metric d_{P} on the Cartesian product $M_1 \times \cdots \times M_n$ is defined as follows: For $(x_1, ..., x_n)$, $(y_1, ..., y_n) \in M_1 \times \cdots \times M_n$, $d_{\mathrm{P}}((x_1, ..., x_n), (y_1, ..., y_n)) = \max_{j \in \bar{n}}[d_j(x_j, y_j)]$. (4) Let $\wp_{\mathrm{cl}}(M) = \{X \in \wp(M) : X \text{ is closed}\}$. A metric d_{H} on $\wp_{\mathrm{cl}}(M)$, called the *Hausdorff distance*, is defined as follows: For $X, Y \in \wp_{\mathrm{cl}}(M)$, $d_{\mathrm{H}}(X, Y) = \max\{\sup_{x \in X}[\underline{d}(x, Y)], \ \sup_{y \in Y}[\underline{d}(y, X)]\}$, where $\underline{d}(x, Z) = \inf_{z \in Z}[d(x, z)]$ for $Z \subseteq M$, $x \in X$. (We use the convention that $\sup \varnothing = 0$ and $\inf \varnothing = 1$.) The space $\wp_{\mathrm{nc}}(M) = \{X \in \wp(M) : X$ is closed and nonempty$\}$ is supplied with a metric by taking the restriction of d_{H} to it.

Complete metric spaces consisting of streams are introduced as solutions of appropriate domain equations as in [BZ82, AR89]. Namely, for arbitrary two sets A and B, and for an arbitrary real number κ such that $0 < \kappa < 1$, there exists a complete metric space $\langle \mathbf{Q}, d_Q \rangle$, which is unique up to isometry, satisfying the domain equation: $\mathbf{Q} \cong B \uplus (A \times \mathrm{id}_\kappa(\mathbf{Q}))$. (The existence and uniqueness of such Q have been shown in [BZ82] and [AR89], respectively.) Note that id_κ is necessary for the associated functor with this domain equation to be contractive, which condition ensures the uniqueness of the solution (see [AR89]). Henceforth we fix a real number κ such that $0<\kappa<1$. The metric space $\langle \mathbf{Q}, d_Q \rangle$ can be defined in terms of *projection functions* introduced below, where the projection functions are very similar to the truncation functions of streams but slightly different from them, as we will note below.

DEFINITION 2 (Projection Functions). (1) Let $\mathbf{Q}=(A^{<\omega}\cdot B)\uplus A^{\omega}$. We define *projection functions* $\psi_n : \mathbf{Q} \to \mathbf{Q}$ $(n \in \omega)$ inductively as follows: First, an arbitrary element b_0 of B is fixed. Let $q \in \mathbf{Q}$. (i) $\psi_0(q)=b_0$. (ii) $\psi_{n+1}(q)=q$ if $q \in B$, and $\psi_{n+1}(q)=a\cdot\psi_n(q')$ if $q=a\cdot q'$. (2) Let $\mathbf{P}=\wp_{\mathrm{nc}}(\mathbf{Q})$. For $n\in\omega$ and $p\in\mathbf{P}$, let $\tilde{\psi}_n(p)=\{\psi_n(q) : q\in p\}$.

Note the difference between *truncation and projection*: The values of the projection functions are members of $A^{<\omega}\cdot B$ $(\subseteq \mathbf{Q})$, whereas the values of the truncation functions are members of $(A^{<\omega}\cdot B)\cup A^{<\omega}$ not of $\mathbf{Q}$.

As stated earlier, the metric d_Q can also be formulated in terms of projection functions as follows:

LEMMA 1. (1) *For* $q_1, q_2 \in \mathbf{Q}$, $d_Q(q_1, q_2) = \kappa^{\min\{n : \psi_n(q_1) \neq \psi_n(q_2)\} - 1}$ *if* $\exists n[\psi_n(q_1) \neq \psi_n(q_2)]$; *otherwise* $d_Q(q_1, q_2)=0$.

(2) *For* $p_1, p_2 \in \mathbf{P}$, $d_P(p_1, p_2) = \kappa^{\min\{n : \tilde{\psi}_n(p_1) \neq \tilde{\psi}_n(p_2)\} - 1}$ *if* $\exists n[\tilde{\psi}(p_1) \neq \tilde{\psi}_n(p_2)]$; *otherwise* $d_P(p_1, p_2)=0$.

(3) *For every* $n\in\omega$, *there exists* $\varepsilon>0$ *such that* $\forall p_1, p_2 \in \mathbf{P}[d_P(p_1, p_2) \leqslant \varepsilon \Rightarrow \tilde{\psi}_n(p_1)=\tilde{\psi}_n(p_2)]$.

Proof. Omitted (see Appendix 1 of [HBR90]).

The notion of *finitely characterized subset* is introduced for establishing that some subsets of a complete metric space are also complete metric spaces.

DEFINITION 3 (Finitely Characterized Subsets). A subset $\mathbf{P}'$ of $\mathbf{P}$ is *finitely characterized* iff there exists $n\in\omega$ and $\mathbf{P}''\subseteq\mathbf{P}$ such that $\forall p\in\mathbf{P}[p\in\mathbf{P}' \Leftrightarrow \tilde{\psi}(p)\in\mathbf{P}'']$.

A property $\Phi(\cdot)$ defined for elements of $\mathbf{P}$ is called *finitely characterized*, if $\{p\in\mathbf{P} : \Phi(p)\}$ is finitely characterized. The next example presents such a property.

EXAMPLE 1. Fix $n\in\omega$. An element $p\in\mathbf{P}$ is said to be *nonempty at level* n, if $p^{[n]}\cap A^n \neq \varnothing$. Let $\mathbf{P}'=\{p\in\mathbf{P} : p$ is nonempty at level $n\}$. Then it is immediate that $\forall p\in\mathbf{P}[p\in\mathbf{P}' \Leftrightarrow \tilde{\psi}_{n+1}(p)\in\mathbf{P}']$. Thus $\mathbf{P}'$ is finitely characterized, and therefore, the property "being nonempty at level n" is finitely characterized. Note that $\mathbf{P}''$ in Definition 3 is equal to $\mathbf{P}'$ here.

The next lemma states that finitely characterized subsets and intersections of finitely characterized subsets are complete metric spaces with the original metric restricted to them. This lemma will be used in the proof of full abstraction to show that the domains of denotational semantics to be presented below are complete metric spaces.

LEMMA 2. (1) *Every finitely characterized subset* $\mathbf{P}'$ *of* $\mathbf{P}$ *is closed in* $\mathbf{P}$.

(2) *For every family* $\mathscr{P}$ *of finitely characterized subsets of* $\mathbf{P}$, $\bigcap \mathscr{P}$ *is closed in* $\mathbf{P}$.

Proof. Omitted (see the proof of Lemma 2 of [HBR90]).

3. A Nonuniform Language with Parallel Composition

The first language $\mathscr{L}_1$ is a *nonuniform* language with recursion and *parallel composition* but no communication.

First, an operational model $\mathscr{O}_1$ is introduced in terms of a Plotkin-style transition system.

Then a denotational model $\mathscr{D}_1$ is defined compositionally by means of interpreted operations of the language, with meanings of recursive programs as fixed points of the denotational semantic domain, a complete metric space consisting of sets of streams of pairs of states.

The correctness of $\mathscr{D}_1$ w.r.t. $\mathscr{O}_1$ is established, as in [Rut89] and [BR91], by means of the fixed point method introduced in [KR90].

Finally, full abstraction of $\mathscr{D}_1$ is shown by means of a context with parallel composition:

> Given two statements s_1 and s_2 with different denotational meanings, a suitable statement T is constructed such that the operational meanings of $s_1 \parallel T$ and $s_2 \parallel T$ are distinct.

For constructing such T, a combinatorial method called the *testing method* is introduced in Lemma 13 (Testing Lemma). By means of this, the full abstraction of $\mathscr{D}_1$ w.r.t. $\mathscr{O}_1$ is established.

3.1. *The Language* $\mathscr{L}_1$

The language $\mathscr{L}_1$ is the simplest nonuniform concurrent language with recursion: It has parallel composition but no communication, and its elementary actions consist only of value assignments to variables.

Note that sequential composition as in [BKO88] is not included in this language: We use *prefixing* of assignment statements as in [Mil80], where *action prefixing* is used in a uniform setting, for simplicity of models for the language. However, there is no difficulty in constructing a fully abstract denotational model for a language which is like $\mathscr{L}_1$, but which has general sequential composition instead of prefixing.

(From now on we use the phrase "let $(x \in) M$ be $\cdots$" to introduce a set M with variable x ranging over M.)

Notation 2. (1) Let $(v \in)$ V denote some abstract domain of values.

(2) Let $(x \in)$ IVar denote the set of *individual variables*.

(3) Let $(\sigma \in) \Sigma$ denote the domain of *states*: $\Sigma = (\text{IVar} \to \text{V})$.

(4) Let $(e \in)$ VExp denote the set of *value expressions*.

(5) Let $(b \in)$ BExp denote the set of *Boolean expressions*.

We assume a simple syntax (not specified here) for e and b. "Simple" ensures at least that no side effects or nontermination occurs in their evaluation. The evaluations of e and b in state σ are denoted by $[\![e]\!](\sigma)$ and $[\![b]\!](\sigma)$, respectively. The full abstraction of a denotational model is established under this assumption.

Let X range over RVar, the set of recursion variables, and let ξ range over SVar, the set of statement variables. Note that recursion variables are used as names of statements defined by recursion, while statement variables are used as place holders for defining *contexts* of a language.

The language $\mathscr{L}_1$ is introduced as a subset of $\mathscr{L}_1^*$, a language with place holders.

Definition 4 (Language $\mathscr{L}_1$). (1) The set of statements of the nonuniform concurrent language $(S \in) \mathscr{L}_1^*$ is defined by the following BNF-syntax:

$$S ::= \mathbf{0} \mid (x := e); S \mid \text{If}(b, S_1, S_2) \mid S_1 + S_2 \mid S_1 \parallel S_2 \mid X \mid \xi.$$

Here $\mathbf{0}$ denotes *inaction*; $(x := e)$; S denotes the result of prefixing the assignment $(x := e)$ to the statement S; If$(\cdot, \cdot, \cdot)$ is the usual *conditional* construct; $+$ and $\parallel$ denote *alternative choice* and *parallel composition*, respectively.[3]

Let FV(S) denote the set of statement variables contained in S.

(2) Let $(s \in) \mathscr{L}_1$ be the set of statements with not statement variable. That is, $\mathscr{L}_1 = \{S \in \mathscr{L}_1^* : \text{FV}(S) = \varnothing\}$. For $\xi \in \text{SVar}$, let $\mathscr{L}_1^\xi = \{S \in \mathscr{L}_1^* : \text{FV}(S) \subseteq \{\xi\}\}$.

(3) The set of *guarded statements* $(g \in) \mathscr{G}_1$ is defined by the following BNF-syntax:

$$g ::= \mathbf{0} \mid (x := e); s \mid \text{If}(b, g_1, g_2) \mid g_1 + g_2 \mid g_1 \parallel g_2.$$

[3] In this language, the precedence of ';', '+', and '∥' is higher than that of ',' occurring in the construct If$(\cdot, \cdot, \cdot)$.

(4) We assume that each recursion variable X is associated with an element g_X of $\mathscr{G}_1$ by a set of declarations $D = \{\langle X, g_X \rangle\}_{X \in \mathrm{RVar}}$. A *program* consists of a pair $\langle s, D \rangle$.

In the sequel of this section, we fix a declaration set $D = \{\langle X, g_X \rangle\}_{X \in \mathrm{RVar}}$.

For every $b \in \mathrm{BExp}$, we regard the construct "If$(b, \cdot, \cdot)$" as a binary operator on statements. Also, for every $x \in \mathrm{IVar}$ and $e \in \mathrm{VExp}$, we regard the construct "$(x := e); \cdot$" as a unary operator on statements. Thus we get as single-sorted *signature* $\mathscr{S}_1$ with the sort of statements; the languages $\mathscr{L}_1^*$ and $\mathscr{L}_1$ can be formulated as the set of terms and the set of closed terms generated by $\mathscr{S}_1$, respectively.

We introduce the notion of a *context* and some uses of it as follows:

Notation 3. Let $\mathscr{L}^*$ be a language formulated as the set of terms generated by a signature $\mathscr{S}$ and a variable set $\{\xi_i\}$.

(1) For $S \in \mathscr{L}^*$ and a sequence of distinct variables $(\xi_1, ..., \xi_n)$, the pair $\langle S, (\xi_1, ..., \xi_n) \rangle$ is called a context of $\mathscr{L}^*$. We sometimes write $S_{(\xi_1, ..., \xi_n)}$ for $\langle S, (\xi_1, ..., \xi_n) \rangle$. When the notation $S_{(\xi_1, ..., \xi_n)}$ is used, it is always assumed that $\mathrm{FV}(S) \subseteq \{\xi_1, ..., \xi_n\}$.

(2) For a context $S_{(\xi_1, ..., \xi_n)}$ and $S_1, ..., S_n \in \mathscr{L}^*$, the notation $S[(S_1, ..., S_n)/(\xi_1, ..., \xi_n)]$ denotes the result of simultatneously replacing ξ_i in S with S_i, $i \in \bar{n}$. More simply, we sometimes write $S_{(\xi_1, ..., \xi_n)}[S_1, ..., S_n]$ for $S[(S_1, ..., S_n)/(\xi_1, ..., \xi_n)]$.

(3) Let $\mathscr{I}$ be an *interpretation*, i.e., a set of interpreted operations for the signature $\mathscr{S}$ with an underlying domain $\mathbf{P}$ (see [Rut90] for a formal definition of an interpretation for a signature); let $S_{(\xi_1, ..., \xi_n)}$ be a context. For $p_1, ..., p_n \in \mathbf{P}$, let $[\![S]\!]^{\mathscr{I}}\,[(\xi_1, ..., \xi_n)/(p_1, ..., p_n)]$ denote the interpretation of S under $\mathscr{I}$ with the assignment of the value p_i to the variable ξ_i, $i \in \bar{n}$. More simply, we sometimes write $[\![S_{(\xi_1, ..., \xi_n)}]\!]^{\mathscr{I}}\,(p_1, ..., p_n)$ for $[\![S]\!]^{\mathscr{I}}\,[(p_1, ..., p_n)/(\xi_1, ..., \xi_n)]$.

3.2. *Operational Model $\mathscr{O}_1$ for $\mathscr{L}_1$*

The operational model $\mathscr{O}_1$ rests on a transition system $\rightarrow_1$ of the style of [Plo81]. The transition relation $\rightarrow_1 \subseteq (\mathscr{L}_1 \times \Sigma) \times (\mathscr{L}_1 \times \Sigma)$ is defined as follows. For $s_1, s_2 \in \mathscr{L}_1$ and $\sigma_1, \sigma_2 \in \Sigma$, we write $\langle s_1, \sigma_1 \rangle \rightarrow_1 \langle s_2, \sigma_2 \rangle$ for $(\langle s_1, \sigma_1 \rangle, \langle s_2, \sigma_2 \rangle) \in \rightarrow_1$ for easier readability.

DEFINITION 5 (Transition Relation $\rightarrow_1$). The transition relation $\rightarrow_1$ is defined as the smallest relation satisfying the following rules (1) to (6). For

$\sigma \in \Sigma$, $x \in \mathrm{IVar}$, and $v \in \mathbf{V}$, the notation $\sigma[v/x]$ is used to denote a state σ' which is the same as σ except that $\sigma'(x) = v$.

$$(1)\quad \langle (x := e); s, \sigma \rangle \rightarrow_1 \langle s, \sigma[[\![e]\!](\sigma)/x] \rangle.$$

$$(2)\quad \frac{\langle s_1, \sigma \rangle \rightarrow_1 \langle s, \sigma' \rangle}{\langle \mathrm{If}(b, s_1, s_2), \sigma \rangle \rightarrow_1 \langle s, \sigma' \rangle}\ ([\![e]\!](\sigma) = tt)$$

$$(3)\quad \frac{\langle s_2, \sigma \rangle \rightarrow_1 \langle s, \sigma' \rangle}{\langle \mathrm{If}(b, s_1, s_2), \sigma \rangle \rightarrow_1 \langle s, \sigma' \rangle}\ ([\![e]\!](\sigma) = f\!f)$$

$$(4)\quad \frac{\langle s_1, \sigma \rangle \rightarrow_1 \langle s, \sigma' \rangle}{\begin{array}{l}\langle s_1 + s_2, \sigma \rangle \rightarrow_1 \langle s, \sigma' \rangle \\ \langle s_2 + s_1, \sigma \rangle \rightarrow_1 \langle s, \sigma' \rangle\end{array}}$$

$$(5)\quad \frac{\langle s_1, \sigma \rangle \rightarrow_1 \langle s, \sigma' \rangle}{\begin{array}{l}\langle s_1 \parallel s_2, \sigma \rangle \rightarrow_1 \langle s \parallel s_2, \sigma' \rangle \\ \langle s_2 \parallel s_1, \sigma \rangle \rightarrow_1 \langle s_2 \parallel s, \sigma' \rangle\end{array}}$$

$$(6)\quad \frac{\langle g_X, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle}{\langle X, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle}\ (\langle X, g_X \rangle \in D).$$

The last rule, called the *recursion rule*, stipulates that for each declaration $\langle X, g_X \rangle \in D$, transitions of the recursion variable X are derived from those of its body g_X, as usual.

Let us call a statement $s \in \mathscr{L}_1$ *finitely branching* iff for every $\sigma \in \Sigma$, $\{\langle s', \sigma' \rangle \in \mathscr{L}_1 \times \Sigma : \langle s, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle)$ is finite. Then, the transition relation $\rightarrow_1$ is finitely branching in the following sense:

LEMMA 3. *Every $s \in L_1$ is finitely branching.*

Proof. By induction on the structure of s. See the proof of Lemma 3 of [HBR90] for details. ▮

An operational model $\mathcal{O}_1$ is defined by means of $\rightarrow_1$ as the fixed point of a higher-order mapping $\psi_1^{\mathcal{O}}$.

DEFINITION 6 (Operational Model $\mathcal{O}_1$ for $\mathscr{L}_1$).

(1) Let $\mathbf{M}_1^{\mathcal{O}} = (\mathscr{L}_1 \rightarrow (\Sigma \rightarrow \wp_{\mathrm{nc}}(\Sigma^{\leqslant \omega})))$, equipped with a metric d defined as in Section 2. Then, let $\Psi_1^{\mathcal{O}} : \mathbf{M}_1^{\mathcal{O}} \rightarrow \mathbf{M}_1^{\mathcal{O}}$ be defined as follows: For $f \in \mathbf{M}_1^{\mathcal{O}}$, $s \in \mathscr{L}_1$, and $\sigma \in \Sigma$,

$$\Psi_1^{\mathcal{O}}(f)(s)(\sigma) = \begin{cases} \bigcup \{\sigma' \cdot f(s')(\sigma') : \langle s, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle\} \\ \qquad \text{if } \exists \langle s', \sigma' \rangle [\langle s, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle], \\ \{\varepsilon\} \qquad \text{otherwise.} \end{cases}$$

The right-hand side of the above equation is closed by Lemma 3, and therefore, indeed, $\Psi_1^{\mathcal{O}}(f) \in \mathbf{M}_1^{\mathcal{O}}$. Moreover, it is immediate from the above definition that for $f, f' \in \mathbf{M}_1^{\mathcal{O}}$, $d(\Psi_1^{\mathcal{O}}(f), \Psi_1^{\mathcal{O}}(f)) \leqslant \kappa \cdot d(f, f')$, where $\kappa(<1)$ is the fixed positive real number introduced in Section 2. Thus, $\Psi_1^{\mathcal{O}}$ is a contraction from $\mathbf{M}_1^{\mathcal{O}}$ to $\mathbf{M}_1^{\mathcal{O}}$.

(2) Let the operational model $\mathcal{O}_1$ be the unique fixed point of $\Psi_1^{\mathcal{O}}$. By the definition, one has $\mathcal{O}_1 : \mathcal{L}_1 \to (\Sigma \to \wp_{\text{nc}}(\Sigma^{\leqslant \omega}))$, and for each $s \in \mathcal{L}_1$ and $\sigma \in \Sigma$,

$$\mathcal{O}_1[\![s]\!](\sigma) = \begin{cases} \bigcup \{\sigma' \cdot \mathcal{O}_1[\![s']\!](\sigma') : \langle s, \sigma \rangle \to_1 \langle s', \sigma' \rangle\} \\ \qquad \text{if } \exists \langle s', \sigma' \rangle [\langle s, \sigma \rangle \to_1 \langle s', \sigma' \rangle], \\ \{\varepsilon\} \qquad \text{otherwise.} \end{cases}$$

Note that $\mathcal{O}_1$ is not compositional, as the following example shows.

EXAMPLE 2. Let $x \in \text{IVar}$. Then

$$\begin{aligned} \mathcal{O}_1[\![(x := 0); (x := x+1); \mathbf{0}]\!] &= \mathcal{O}_1[\![(x := 0); (x := 1); \mathbf{0}]\!] \\ &= (\lambda\sigma : \{(\sigma[0/x], \sigma[1/x])\}), \end{aligned}$$

but

$$\begin{aligned} &\mathcal{O}_1[\![((x := 0); (x := x+1); 0) \parallel ((x := 2); \mathbf{0})]\!] \\ &\quad \neq \mathcal{O}_1[\![((x := 0); (x := 1); \mathbf{0}) \parallel ((x := 2); \mathbf{0})]\!]. \end{aligned}$$

3.3. *Denotational Model $\mathcal{D}_1$ for $\mathcal{L}_1$*

The denotational model $\mathcal{D}_1$ is defined compositionally by means of interpreted operations of the language.

The denotational semantic domain $\mathbf{P}_1$ is a complete metric space consisting of sets of streams of pairs of states. The meaning of a recursion variable X with the declaration $\langle X, g_X \rangle$ is defined as the fixed point of the contraction which maps each process $p \in \mathbf{P}_1$ to the interpretation of g_X under the interpreted operations with the assignment of p to X. It turns out that the fixed point is the unique solution of the equation $X = g_X$ under the interpretation consisting of the interpreted operations.

The domain $\mathbf{P}_1$ is defined by:

DEFINITION 7 (Denotational Semantic Domain $\mathbf{P}_1$ for $\mathcal{L}_1$). (1) Let $\mathbf{Q}_1$ be the unique solution of $\mathbf{Q}_1 \cong \Sigma \uplus ((\Sigma \times \Sigma) \times \text{id}_\kappa(\mathbf{Q}_1))$. One has $\mathbf{Q}_1 \cong ((\Sigma \times \Sigma)^{<\omega} \cdot \Sigma) \cup (\Sigma \times \Sigma)^{\omega}$.

(2) For $p \in \wp_{nc}(\mathbf{Q}_1)$, and $r \in (\Sigma \times \Sigma)^{<\omega}$, the *remainder* of p with prefix r, denoted by $p[r]$, is defined by $p[r] = \{q \in \mathbf{Q}_1 : r \cdot q \in p\}$.

(3) The *initial state* of a sequence $q \in \mathbf{Q}_1 \cup (\Sigma \times \Sigma)^+$, denoted by $\text{istate}_1(q)$, is defined as follows: Let $\text{istate}_1(q) = \sigma$ if $q = (\langle \sigma, \sigma' \rangle) \cdot q'$, and let $\text{istate}_1(q) = \sigma''$ if $q = (\sigma'')$.

(4) For $p \in \wp_{nc}(\mathbf{Q}_1)$ and $\sigma \in \Sigma$, $p\langle \sigma \rangle$ is the set of those elements of p whose initial state is σ. That is, $p\langle \sigma \rangle = \{q \in p : \text{istate}_1(q) = \sigma\}$.

(5) Let $p \in \wp_{nc}(\mathbf{Q}_1)$, and $n \in \omega$. The process p is *uniformly nonempty at level n* iff

$$\forall r \in (\Sigma \times \Sigma)^n \; [p[r] \neq \emptyset \Rightarrow \forall \sigma \in \Sigma [p[r]\langle \sigma \rangle \neq \emptyset]].$$

Moreover, p is *uniformly nonempty* iff it is uniformly nonempty at every level $n \in \omega$.

(6) The set $\mathbf{P}_1$, the domain of processes for $\mathscr{L}_1$, is given by

$$\mathbf{P}_1 = \{p \in \wp_{nc}(\mathbf{Q}_1) : p \text{ is uniformly nonempty}\}.$$

Remark 1. A subset $\mathbf{P}$ of $\wp_{nc}(\mathbf{Q}_1)$ is said to be *closed under taking remainders* iff $\forall p \in \mathbf{P}$, $\forall r \in (\Sigma \times \Sigma)^{<\omega} \; [p[r] \neq \emptyset \Rightarrow p[r] \in \mathbf{P}]$. Given an arbitrary subset $\mathbf{P}_0$ of $\wp_{nc}(\mathbf{Q}_1)$, it is routine to check that the largest subset $\mathbf{P}'_0$ of $\wp_{nc}(\mathbf{Q}_1)$ which is included in $\mathbf{P}_0$ and closed under taking remainders is given by $\mathbf{P}'_0 = \{p \in \wp_{nc}(\mathbf{Q}_1) : \forall r \in (\Sigma \times \Sigma)^{<\omega} \; [p[r] \neq \emptyset \Rightarrow p[r] \in \mathbf{P}_0]\}$. Thus $\mathbf{P}_1$ is the largest subset of $\wp_{nc}(\mathbf{Q}_1)$ which is included in $\{p \in \wp_{nc}(\mathbf{Q}_1) : p \text{ is uniformly nonempty at level } 0\}$ and closed under taking remainders.

It is needed that each element of $p \in \mathbf{P}_1$ is uniformly nonempty, for defining a parallel composition $\|$ as a binary operation on $\mathbf{P}_1$ in the sequel.

LEMMA 4. *The set* $\mathbf{P}_1$ *is closed in* $\wp_{nc}(\mathbf{Q}_1)$, *and therefore,* $\mathbf{P}_1$ *is a complete metric space with the original metric of* $\wp_{nc}(\mathbf{Q}_1)$ *restricted to it.*

Proof. The closedness can be established using Lemma 2. See the proof of Lemma 4 of [HBR90], for details. ∎

The interpretation $\mathscr{I}_1$ for the signature of $\mathscr{L}_1$ is defined as follows:

DEFINITION 8 (Interpretation $\mathscr{I}_1$ for Signature of $\mathscr{L}_1$). (1) $\tilde{\mathbf{0}}_1 = \{(\sigma) : \sigma \in \Sigma\}$.

(2) For $x \in \text{IVar}$ and $e \in \text{VExp}$, the function $\text{asg}_1(x, e) : \mathbf{P}_1 \to \mathbf{P}_1$, which is the interpretation of the unary operator "$(x := e); \cdot$" on statements, is defined as follows: For every $p \in \mathbf{P}_1$, $\text{asg}_1(x, e)(p) = \{(\langle \sigma, \sigma[[\![e]\!](\sigma)/x] \rangle) \cdot p : \sigma \in \Sigma\}$, where $(\langle \sigma, \sigma[[\![e]\!](\sigma)/x] \rangle) \cdot p$ denotes the concatenation of $(\langle \sigma, \sigma[[\![e]\!](\sigma)/x] \rangle)$ and p.

(3) For $b \in \text{BExp}$, the function $\text{if}(b): \mathbf{P}_1 \times \mathbf{P}_1 \to \mathbf{P}_1$, which is the interpretation of the binary operator "If$(b, \cdot, \cdot)$" on statements, is defined as follows: For every $p_1, p_2 \in \mathbf{P}_1$, $\text{if}(b)(p_1, p_2) = \bigcup_{\sigma \in \Sigma} [\text{if}([\![b]\!](\sigma) = \text{tt}, p_1\langle\sigma\rangle, p_2\langle\sigma\rangle)]$.

(4) For $p \in \mathbf{P}_1$, $p \cap ((\Sigma \times \Sigma) \times \mathbf{Q}_1)$ is called the *action part* of p and denoted by p^+, and the set $p \cap \tilde{\mathbf{0}}_1$ is called the *inaction part* of p. The action part of the alternative composition of two processes is the union of the action parts of those processes, and its inaction part is the intersection of the inaction parts of them. That is, for $p_1, p_2 \in \mathbf{P}_1$, $p_1 \tilde{+} p_2 = p_1^+ \cup p_2^+ \cup \{(\sigma) : (\sigma) \in p_1 \cap p_2\}$.

(5) For $p_1, p_2 \in \mathbf{P}_1$, let $p_1 \# p_2$ be the intersection of the inaction parts of p_1 and p_2. The parallel composition $\tilde{\|}: \mathbf{P}_1 \times \mathbf{P}_1 \to \mathbf{P}_1$ is defined recursively as follows: For every $p_1, p_2 \in \mathbf{P}_1$,

$$\begin{aligned} p_1 \tilde{\|} p_1 &= (p_1 \tilde{\|\!\!\_} p_2) \cup (p_2 \tilde{\|\!\!\_} p_1) \cup (p_1 \# p_2), \\ p_1 \tilde{\|\!\!\_} p_2 &= \bigcup \{\langle\sigma, \sigma'\rangle \cdot (p_1[\langle\sigma, \sigma'\rangle] \tilde{\|} p_2) : p_1[\langle\sigma, \sigma'\rangle] \neq \varnothing\}. \end{aligned} \tag{2}$$

Formally the operation $\tilde{\|}$ is defined as the fixed point of a suitably defined contraction: Let $\mathbf{M}_1^{\|} = (\mathbf{P}_1 \times \mathbf{P}_1)$, $\Omega_1^{\|}: \mathbf{M}_1^{\|} \to \mathbf{M}_1^{\|}$ be defined as follows: For $F \in \mathbf{M}_1^{\|}$, and $p_1, p_2 \in \mathbf{P}_1$, $\Omega_1^{\|}(F)(p_1, p_2) = \Omega_1^{\|\!\!\_}(F)(p_1, p_2) \cup \Omega_1^{\|\!\!\_}(F)(p_2, p_1) \cup (p_1 \# p_2)$, where $\Omega_1^{\|\!\!\_}(F)(p_1, p_2) = \bigcup \{\langle\sigma, \sigma'\rangle \cdot F(p_1[\langle\sigma, \sigma'\rangle], p_2) : p_1[\langle\sigma, \sigma'\rangle] \neq \varnothing\}$. It is shown that $\Omega_1^{\|}(F)(p_1, p_2)$ is nonempty and uniformly nonempty at level 0 as follows: For every $\sigma \in \Sigma$, suppose $\neg\exists\sigma'[\Omega_1^{\|}(F)(p_1, p_2)[\langle\sigma, \sigma'\rangle] \neq \varnothing]$. Then, by the definition of $\Omega_1^{\|}$, one has $\neg\exists\sigma'[p_1[\langle\sigma, \sigma'\rangle] \neq \varnothing]$ and $\neg\exists\sigma'[p_2[\langle\sigma, \sigma'\rangle] \neq \varnothing]$. Thus, by the fact that p_1 and p_2 are uniformly nonempty at level 0, one has $(\sigma) \in (p_1 \# p_2)$. Moreover, $\Omega_1^{\|}(F)(p_1, p_2)$ is uniformly nonempty at level $n \geqslant 1$, since $\Omega_1^{\|\!\!\_}(F)(s_1, s_2)$ and $\Omega_1^{\|\!\!\_}(F)(s_2, s_1)$ are uniformly nonempty at level n by their definitions. Hence $\Omega_1^{\|}(F)(p_1, p_2) \in \mathbf{P}_1$. It is immediate that $\Omega_1^{\|}$ is a contraction. Let $\tilde{\|} = \text{fix}(\Omega_1^{\|})$, and $\tilde{\|\!\!\_} = \Omega_1^{\|\!\!\_}(\tilde{\|})$.

(6) Let $\mathscr{I}_1 = \{\tilde{\mathbf{0}}_1, \{\text{asg}_1(x, e) : \langle x, e\rangle \in \text{IVar} \times \text{VExp}\}, \{\text{if}(b) : b \in \text{BExp}\}, \tilde{+}, \tilde{\|}\}$.

The next lemma follows immediately from Definition 8 (5). We shall use it for establishing the full abstraction of the denotational model $\mathscr{D}_1$ defined below.

LEMMA 5. (1) $\langle\sigma, \sigma'\rangle \cdot q \in p_1 \tilde{\|} p_2 \Leftrightarrow (q \in (p_1[\langle\sigma, \sigma'\rangle] \tilde{\|} p_2)) \vee (q \in (p_1 \tilde{\|} p_2[\langle\sigma, \sigma'\rangle]))$.

(2) $\forall p_1, p_2 \in \mathbf{P}_1[p_1 \tilde{\|} p_2 = p_2 \tilde{\|} p_1]$.

In terms of the interpretation $\mathscr{I}_1$, the denotational model $\mathscr{D}_1$ is defined as follows:

DEFINITION 9 (Denotational Model $\mathscr{D}_1$ for $\mathscr{L}_1$). The model $\mathscr{D}_1: \mathscr{L}_1 \to \mathbf{P}_1$ is defined by induction on the structure of $s \in \mathscr{L}_1$.

(1) First, for each recursion variable X, $\mathscr{D}_1[\![X]\!]$ is defined as the fixed point of a contraction defined in terms of the declarations. Let $D = \{\langle X, g_X \rangle\}_{X \in \mathrm{RVar}}$ be the set of declarations. Let $\mathbf{M}_1^{\mathscr{D}} = (\mathrm{RVar} \to \mathbf{P}_1)$, and let $\Pi_1: \mathbf{M}_1^{\mathscr{D}} \to \mathbf{M}_1^{\mathscr{D}}$ be defined as follows: For $\mathbf{p} \in \mathbf{M}_1^{\mathscr{D}}$, $X \in \mathrm{RVar}$, $\Pi_1(\mathbf{p})(X) = [\![g_X]\!]^{\mathscr{I}_1}[(\mathbf{p}(Y_1^X), ..., \mathbf{p}(Y_{l(X)}^X))/(Y_1^X, ..., Y_{l(X)}^X)]$, where $\{Y_1^X, ..., Y_{l(X)}^X\}$ is the set of recursion variables contained in g_X. (See Notation 3 for the notation $[\![g_X]\!]^{\mathscr{I}_1}(\cdots)$.) The mapping Π_1 is a contraction from $\mathbf{M}_1^{\mathscr{D}}$ to $\mathbf{M}_1^{\mathscr{D}}$. Let $\mathbf{p}_\mathrm{p} = \mathrm{fix}(\Pi_1)$. For $X \in \mathrm{RVar}$, let us define X^{D_1}, the denotational meaning of X by $X^{\mathscr{D}_1} = \mathbf{p}_0(X)$.

(2) Next, for a composite statement $s \in \mathscr{L}_1$, $\mathscr{D}_1[\![s]\!]$ is defined as follows: For each operator F of $\mathscr{L}_1$ with arity r, and $s_1, ..., s_r \in \mathscr{L}_1$, let $\mathscr{D}_1[\![F(s_1, ..., s_r)]\!] = F^{\mathscr{I}_1}(\mathscr{D}_1[\![s_1]\!], ..., \mathscr{D}_1[\![s_r]\!])$, where $F^{\mathscr{I}_1}$ is the interpreted operation in $\mathscr{I}_1$ corresponding to F.

Several properties, including the so-called *image finiteness* for elements of $\mathbf{P}_1$, are introduced. It is shown that the denotational meaning of each statement in $\mathscr{L}_1$ has these properties; this fact is used to establish the full abstraction of $\mathscr{D}_1$.

DEFINITION 10 (Image Finiteness for Elements of $\mathbf{P}_1$). Let $p \in \mathbf{P}_1$ and $n \in \omega$.

(1) The process p is *image finite at level n*, written $\mathrm{IFin}_1^{(n)}(p)$, iff $\forall r \in (\Sigma \times \Sigma)^n$, $\forall \sigma[\{\sigma' \in \Sigma : r \cdot \langle \sigma, \sigma' \rangle \in p^{[n+1]}\}$ is finite]. The process p is *image finite*, written $\mathrm{IFin}_1(p)$, iff $\forall n \in \omega[\mathrm{IFin}_1^{(n)}(p)]$.

(2)(i) We say that *only a finite number of individual variables are relevant to the nonterminating part of p at level n*, written $\mathrm{FIRN}_1^{(n)}(p)$, iff there exists $\mathscr{X} \in \wp_f(\mathrm{IVar})$ such that the following holds:

$$\begin{aligned} &\forall r \in (\Sigma \times \Sigma)^n, \forall \vec{\sigma} \in ((\mathrm{IVar} \backslash \mathscr{X}) \to \mathbf{V})^n\ [r \in p^{[n]} \\ &\quad \Leftrightarrow \forall i \in n[\pi_0^2(r(i)) \upharpoonright (\mathrm{IVar} \backslash \mathscr{X}) = \pi_1^2(r(i)) \upharpoonright (\mathrm{IVar} \backslash \mathscr{X})] \\ &\quad \wedge (\langle (\pi_j^2(r(i)) \upharpoonright \mathscr{X}) \cup \vec{\sigma}(i) \rangle_{j \in 2})_{i \in n} \in p^{[n]}]. \end{aligned} \tag{3}$$

That is, for each $r \in (\Sigma \times \Sigma)^n$, if $r \in p^{[n]}$, then, in every step $r(i) = \langle \pi_0^2(r(i)), \pi_1^2(r(i)) \rangle$ or r $(i \in n)$, the value for $x \in \mathrm{IVar} \backslash \mathscr{X}$ is not changed, i.e., $(*): \pi_0^2(r(i)) \upharpoonright (\mathrm{IVar} \backslash \mathscr{X}) = \pi_1^2(r(i)) \upharpoonright (\mathrm{IVar} \backslash \mathscr{X})$, and one may change the value $\pi_j^2(r(i))(x)$ $(j \in 2)$ arbitrarily, i.e., $(\dagger): (\langle (\pi_j^2(r(i)) \upharpoonright \mathscr{X}) \cup \vec{\sigma}(i) \rangle_{j \in 2})_{i \in n} \in p^{[n]}$ for arbitrary $\vec{\sigma} \in ((\mathrm{IVar} \backslash \mathscr{X}) \to \mathbf{V})^n$. Conversely, for arbitrary $\vec{\sigma} \in ((\mathrm{IVar} \backslash \mathscr{X}) \to \mathbf{V})^n$, if one has $(*)$ and $(\dagger)$, then $r \in p^{[n]}$. (See Remark 3 below for a motivation of this definition.)

(ii) Similarly, we say that *only a finite number of individual variables are relevant to the terminating part of p at level n*, written $\mathrm{FIRT}_1^{(n)}(p)$, iff there exists $\mathscr{X} \in \wp_{\mathrm{f}}(\mathrm{IVar})$ such that

$$\forall q \in (\Sigma \times \Sigma)^n \cdot \Sigma, \forall \vec{\sigma} \in ((\mathrm{IVar} \backslash \mathscr{X}) \to \mathbf{V})^{n+1} [q \in p$$
$$\Leftrightarrow \forall i \in n[\pi_0^2(q(i)) \restriction (\mathrm{IVar} \backslash \mathscr{X}) = \pi_1^2(q(i)) \restriction (\mathrm{IVar} \backslash \mathscr{X})]$$
$$\wedge (\langle (\pi_j^2(q(i)) \restriction \mathscr{X}) \cup \vec{\sigma}(i) \rangle_{j \in 2})_{i \in n} \cdot ((q(n) \restriction \mathscr{X}) \cup \vec{\sigma}(n)) \in p]. \quad (4)$$

(iii) We say that *only a finite number of individual variable are relevant to p*, written $\mathrm{FIR}_1(p)$, iff $\forall n \in \omega[\mathrm{FIRN}_1^{(n)}(p) \wedge \mathrm{FIRT}_1^{(n)}(p)]$.

(3) $\mathbf{P}_1^* = \{p \in \mathbf{P}_1 : \mathrm{IFin}_1(p) \wedge \mathrm{FIR}_1(p)\}$.

Remark 2. It is immediate that $\{p \in \mathbf{P}_1 : \mathrm{IFin}_1(p)\}$ is the largest subset of $\mathbf{P}_1$ which is included in $\{p \in \mathbf{P}_1 : \mathrm{IFIN}_1^{(0)}(p)\}$ and closed under taking remainders,

Remark 3. (1) Note that for some set D of declarations and some statement s, we cannot take *one* $\mathscr{X} \in \wp_{\mathrm{f}}(\mathrm{IVar})$ such that (3) holds for *every* $n \in \omega$ and $p = \mathscr{D}[\![s]\!]$. For example, suppose $\mathrm{IVar} = \{x_n : n \in \omega\}$ and $\mathrm{RVar} = \{X_n : n \in \omega\}$, and let $D = \{\langle X_n, (x_n := 1); X_{n+1} \rangle : n \in \omega\}$, $p = \mathscr{D}_1[\![X_0]\!]$. Then, the greater $n \in \omega$ is given, the greater $\mathscr{X} \in \wp_{\mathrm{f}}(\mathrm{IVar})$ should be taken so that one has (3).

(2) It is easy to check that for $\mathscr{X}_1, X_2 \in \wp_{\mathrm{f}}(\mathrm{IVar})$ with $\mathscr{X}_1 \subseteq \mathscr{X}_2$, the property (3) (resp. (4)) for $\mathscr{X} = \mathscr{X}_1$ implies (3) (resp.(4)) for $\mathscr{X} = \mathscr{X}_2$.

It turns out that the denotational meaning of each statement is a member of $\mathbf{P}_1^*$, which is used for establishing the full abstraction of $\mathscr{D}_1$.

LEMMA 6. (1) *The set* $\mathbf{P}_1^*$ *is closed in* $\mathbf{P}_1$.

(2) $\forall p \in \mathbf{P}_1^*, \forall r \in (\Sigma \times \Sigma)^{<\omega} [p[r] \neq \varnothing \Rightarrow p[r] \in \mathbf{P}_1^*]$. *That is,* $\mathbf{P}_1^*$ *is closed under taking remainders.*

(3) *The set* $\mathbf{P}_1^*$ *is closed under all interpreted operations of* $\mathscr{L}_1$.

(4) $\mathscr{D}_1[\mathscr{L}_1] \subseteq \mathbf{P}_1^*$.

(5) $\forall p \in \mathscr{D}_1[\mathscr{L}_1], \forall r \in (\Sigma \times \Sigma)^{<\omega} [p[r] \neq \varnothing \Rightarrow p[r] \in \mathbf{P}_1^*]$.

Proof. Similar to the proof of Lemma 4. See the proof of Lemma 6 of [HBR90], for details. ∎

3.4. *Correctness of $\mathscr{D}_1$ with Respect to $\mathscr{O}_1$*

The correctness of the denotational model is shown as in [Rut89]: For the denotational model $\mathscr{D}_1$, an alternative formulation, called an *intermediate model*, is given, in terms of the same transition system which was sed for the definition of $\mathcal{O}_1$. Let $\tilde{\mathcal{O}}_1$ be the intermediate model. Then the orrectness is proved by showing that, for an appropriate abstraction funcon α_1, both $\alpha_1 \circ \tilde{\mathcal{O}}_1$ and $\mathcal{O}_1$ are a fixed point of the same contraction, which y Banach's Theorem has a unique fixed point.

.4.1. *Intermediate Model for $\mathcal{L}_1$ and Semantic Equivalence*

First, the intermediate model $\tilde{\mathcal{O}}_1$, which is an alternative formulation of $\mathcal{O}_1$, is defined in terms of the transition relation $\rightarrow_1$.

DEFINITION 11 (Intermediate Model $\tilde{\mathcal{O}}_1$ for $\mathcal{L}_1$).

(1) Let $\mathbf{M}_1 = (\mathcal{L}_1 \rightarrow \mathbf{P}_1)$, and let $\Psi_1 : \mathbf{M}_1 \rightarrow \mathbf{M}_1$ be defined as follows: or $F \in \mathbf{M}_1$, $s \in \mathcal{L}_1$,

$$\Psi_1(F)(s) = \bigcup \{(\langle \sigma, \sigma' \rangle) \cdot F(s') : \sigma \in \Sigma \wedge \langle s, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle\}$$
$$\cup \{(\sigma) : \sigma \in \Sigma \wedge \neg \exists \langle s', \sigma' \rangle [\langle s, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle]\}.$$

he right-hand side of the above equation is closed by Lemma 3; Ψ_1 is a ontraction from $\mathbf{M}_1$ to $\mathbf{M}_1$.

(2) Let $\tilde{\mathcal{O}}_1 = \text{fix}(\Psi_1)$. By the definition, one has, for $s \in \mathcal{L}_1$, that

$$\tilde{\mathcal{O}}_1[\![s]\!] = \bigcup \{(\langle \sigma, \sigma' \rangle) \cdot \tilde{\mathcal{O}}_1[\![s']\!] : \sigma \in \Sigma \wedge \langle s, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle\}$$
$$\cup \{(\sigma) : \sigma \in \Sigma \wedge \neg \exists \langle s', \sigma \rangle [\langle s, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle]\}.$$

It turns out that $\tilde{\mathcal{O}}_1$ is identical to $\mathcal{D}_1$.

LEMMA 7 (Semantic Equivalence for $\mathcal{L}_1$). (1) *Let F be an operator of $\mathcal{L}_1$ with arity r, and let $s_1, ..., s_r \in \mathcal{L}_1$. Then one has*

$$\tilde{\mathcal{O}}_1[\![F(s_1, ..., s_r)]\!] = F^{\mathscr{I}_1}(\tilde{\mathcal{O}}_1[\![s_1]\!], ..., \tilde{\mathcal{O}}_1[\![s_r]\!]).$$

(2) *For $s \in \mathcal{L}_1$, one has $\tilde{\mathcal{O}}_1[\![s]\!] = \mathcal{D}_1[\![s]\!]$.*

As a preliminary to the proof of Lemma 7, we give the next lemma stating that the operation $\tilde{\|}$ is *distributive* w.r.t. set-theoretical union.

LEMMA 8 (Distributivity of $\tilde{\|}$ in $\mathbf{P}_1$). *For $k, l \geqslant 1$, and $p_1, ..., p_k$, $p'_1, ..., p'_l \in \mathbf{P}_1$,*

$$\bigcup_{i \in \bar{k}} [p_i] \,\tilde{\|}\, \bigcup_{j \in \bar{l}} [p'_j] = \bigcup_{\langle i, j \rangle \in \bar{k} \times \bar{l}} [p_i \,\tilde{\|}\, p'_j].$$

Proof. Omitted (see Appendix 2 of [HBR90]). ∎

Proof of Lemma 7. (1) Here we prove the claim for the operator $\|$. For the other operators this is proved (more straightforwardly) in a similar fashion. Let $\mathbf{H}_1 = (\mathscr{L}_1 \times \mathscr{L}_1 \to \mathbf{P}_1)$, and let $F, G \in \mathbf{H}_1$ be defined as follows: For $s_1, s_2 \in \mathscr{L}_1$, $F(s_1, s_2) = \tilde{\mathscr{O}}_1[\![s_1 \parallel s_2]\!]$, $G(s_1, s_2) = \tilde{\mathscr{O}}_1[\![s_1]\!] \mathbin{\tilde{\|}} \tilde{\mathscr{O}}_1[\![s_2]\!]$. Moreover, let $\mathscr{F}_1^{\|} : \mathbf{H}_1 \to \mathbf{H}_1$ be defined as follows: For $f \in \mathbf{H}_1$ and $s_1, s_2 \in \mathscr{L}_1$,

$$\mathscr{F}_1^{\|}(f)(s_1, s_2) = \mathscr{F}_1^{\llcorner\!\llcorner}(f)(s_1, s_2) \cup \mathscr{F}_1^{\llcorner\!\llcorner}(f)(s_2, s_1) \cup \mathscr{F}_1^{\#}(f)(s_1, s_2),$$

where $\mathscr{F}_1^{\llcorner\!\llcorner}(f)(s_1, s_2) = \bigcup \{(\langle \sigma, \sigma' \rangle) \cdot f(s_1', s_2) : \langle s_1, \sigma \rangle \to_1 \langle s_1', \sigma' \rangle\}$, and

$$\mathscr{F}_1^{\#}(s_1, s_2) = \{(\sigma) : \neg \exists \langle s_1', \sigma' \rangle [\langle s_1, \sigma \rangle \to_1 \langle s_1', \sigma' \rangle] \wedge \neg \exists \langle s_2', \sigma \rangle [\langle s_2, \sigma \rangle \to_1 \langle s_2', \sigma' \rangle]\}.$$

Then $\mathscr{F}_1^{\|}$ is a contraction. Let $s_1, s_2 \in \mathscr{L}_1$. By the definition of $\tilde{\mathscr{O}}_1$ and $\to_1$, and Lemma 3, one has $F(s_1, s_2) = \mathscr{F}_1^{\|}(F)(s_1, s_2)$. That is, $F = \text{fix}(\mathscr{F}_1^{\|})$.

Next, let us show that $G = \text{fix}(\mathscr{F}_1^{\|})$. By the definition of $\tilde{\|}$, one has

$$G(s_1, s_2) = (\tilde{\mathscr{O}}_1[\![s_1]\!] \mathbin{\tilde{\llcorner\!\llcorner}} \tilde{\mathscr{O}}_1[\![s_2]\!]) \cup (\tilde{\mathscr{O}}_1[\![s_2]\!] \mathbin{\tilde{\llcorner\!\llcorner}} \tilde{\mathscr{O}}_1[\![s_1]\!]) \cup (\tilde{\mathscr{O}}_1[\![s_1]\!] \mathbin{\tilde{\#}} \tilde{\mathscr{O}}_1[\![s_2]\!]).$$

Thus, for showing $G = \text{fix}(\mathscr{F}_1^{\|})$, it suffices to show $(*)$: $(\tilde{\mathscr{O}}_1[\![s_1]\!] \mathbin{\tilde{\llcorner\!\llcorner}} \tilde{\mathscr{O}}_1[\![s]\!]) = \mathscr{F}_1^{\llcorner\!\llcorner}(G)(s_1, s_2)$, and $(\dagger)$: $(\tilde{\mathscr{O}}_1[\![s_1]\!] \mathbin{\tilde{\#}} \tilde{\mathscr{O}}_1[\![s_2]\!]) = \mathscr{F}_1^{\#}(G)(s_1, s_2)$. The fact $(*)$ is shown as follows:

$$\begin{aligned}
&\tilde{\mathscr{O}}_1[\![s_1]\!] \mathbin{\tilde{\llcorner\!\llcorner}} \tilde{\mathscr{O}}_1[\![s_2]\!] \\
&\quad = \bigcup \{(\langle \sigma, \sigma' \rangle) \cdot (\tilde{\mathscr{O}}_1[\![s_1]\!][\langle \sigma, \sigma' \rangle] \mathbin{\tilde{\|}} \tilde{\mathscr{O}}_1[\![s_2]\!]) : \tilde{\mathscr{O}}_1[\![s_1]\!][\langle \sigma, \sigma' \rangle] \neq \varnothing\} \\
&\quad = \bigcup \Big\{(\langle \sigma, \sigma' \rangle) \cdot \Big(\bigcup \{\tilde{\mathscr{O}}_1[\![s_1']\!] : \langle s_1, \sigma \rangle \to_1 \langle s_1', \sigma' \rangle\} \mathbin{\tilde{\|}} \tilde{\mathscr{O}}_1[\![s_2]\!]\Big) : \\
&\qquad\qquad \exists s_1' [\langle s_1, \sigma \rangle \to_1 \langle s_1', \sigma' \rangle]\Big\} \\
&\quad = \bigcup \Big\{(\langle \sigma, \sigma' \rangle) \cdot \Big(\bigcup \{\tilde{\mathscr{O}}_1[\![s_1']\!] \mathbin{\tilde{\|}} \tilde{\mathscr{O}}_1[\![s_2]\!] : \langle s_1, \sigma \rangle \to_1 \langle s_1', \sigma' \rangle\}\Big) : \\
&\qquad\qquad \exists s_1' [\langle s_1, \sigma \rangle \to_1 \langle s_1', \sigma' \rangle]\Big\} \qquad \text{(by Lemma 8)} \\
&\quad = \bigcup \{(\langle \sigma, \sigma' \rangle) \cdot (\tilde{\mathscr{O}}_1[\![s_1']\!] \mathbin{\tilde{\|}} \tilde{\mathscr{O}}_1[\![s_2]\!]) : \langle s_1, \sigma \rangle \to_1 \langle s_1', \sigma' \rangle\} \\
&\quad = \mathscr{F}_1^{\llcorner\!\llcorner}(G)(s_1, s_2).
\end{aligned}$$

The other fact (†) follows immediately from the definition of $\#$. Thus one has $G(s_1, s_2) = \mathscr{F}_1^{\|}(G)(s_1, s_2)$, i.e., $G = \mathrm{fix}(\mathscr{F}_1^{\|})$. Thus, by Banach's Theorem, one has $F = G$, i.e.,

$$\forall s_1, s_2 \in \mathscr{L}_1[\tilde{\mathcal{O}}_1[\![s_1 \parallel s_2]\!] = \tilde{\mathcal{O}}_1[\![s_1]\!] \mathbin{\tilde{\|}} \tilde{\mathcal{O}}_1[\![s_2]\!]].$$

(2) First, let us show, for $X \in \mathrm{RVar}$, that (‡): $\tilde{\mathcal{O}}_1[\![X]\!] = \mathscr{D}_1[\![X]\!]$. Let $\langle X, g_X \rangle \in D$. Then,

$$\begin{aligned}\tilde{\mathcal{O}}_1[\![X]\!] &= \tilde{\mathcal{O}}_1[\![g_X]\!] \qquad \text{(by the definition of } \tilde{\mathcal{O}}_1) \\ &= [\![g_X]\!]^{\mathscr{I}_1}[(\tilde{\mathcal{O}}_1[\![Y_1^X]\!], ..., \tilde{\mathcal{O}}_1[\![Y_{l(X)}^X]\!])/(Y_1^X, ..., Y_{l(X)}^X)] \qquad \text{(by (1))}, \qquad (5)\end{aligned}$$

where $\{Y_1^X, ..., Y_{l(X)}^X\}$ is the set of recursion variables contained in g_X. Hence $(\lambda X \in \mathrm{RVar} : \tilde{\mathcal{O}}_1[\![X]\!])$ is the fixed point of Π_1 defined in Definition 9. Therefore by the definition of $\mathscr{D}_1[\![X]\!]$, one has (‡). It follows from this and (1), by induction on the structure of $s \in \mathscr{L}_1$, that $\forall s \in \mathscr{L}_1[\tilde{\mathcal{O}}_1[\![s]\!] = \mathscr{D}_1[\![s]\!]]$. ∎

3.4.2. *Correctness of $\mathscr{D}_1$ with Respect to $\mathcal{O}_1$*

An *abstraction function* $\alpha_1 : \mathbf{P}_1 \to (\Sigma \to \wp_{\mathrm{nc}}(\Sigma^{\leqslant \omega}))$ is defined as follows. First, it is defined as the fixed point of a higher-order contraction. Next, it is shown that for a process p, $\alpha(p)$ is characterized as the set of *histories* of *executable* elements of p, where the notions of history and executability to be formally defined below.

DEFINITION 12 (Abstraction Function α_1 for $\mathscr{L}_1$). (1) Let $\mathbf{M}_1^\alpha = (\mathbf{P}_1^* \to (\Sigma \to \wp_{\mathrm{nc}}(\Sigma^{\leqslant \omega})))$, and let $\Delta_1 : \mathbf{M}_1 \to \mathbf{M}_1^\alpha$ be defined as follows: For $F \in \mathbf{M}_1^\alpha$, $p \in \mathbf{P}_1^*$, and $\sigma \in \Sigma$,

$$\begin{aligned}\Delta_1(F)(p)(\sigma) = \bigcup \{(\sigma') \cdot F(p[\langle \sigma, \sigma' \rangle])(\sigma') : p[\langle \sigma, \sigma' \rangle] \neq \varnothing\} \\ \cup \mathrm{if}((\sigma) \in p, \{\varepsilon\}, \varnothing).\end{aligned}$$

Note that the right-hand side of the the above equation is nonempty, since p is uniformly nonempty at level 0. Thus the mapping Δ_1 is a contraction from $\mathbf{M}_1^\alpha$ to $\mathbf{M}_1^\alpha$.

(2) Let $\alpha_1 = \mathrm{fix}(\Delta_1)$. By this definition, it holds for $p \in \mathbf{P}_1^*$ and $\sigma \in \Sigma$, that

$$\begin{aligned}\alpha_1(p)(\sigma) = \bigcup \{(\sigma') \cdot \alpha_1(p[\langle \sigma, \sigma' \rangle])(\sigma') : p[\langle \sigma, \sigma' \rangle] \neq \varnothing\} \\ \cup \mathrm{if}((\sigma) \in p, \{\varepsilon\}, \varnothing).\end{aligned}$$

The abstraction function is to be characterized in another way. First, we need some preliminary definitions.

Intuitively, a sequence $(\langle\sigma_i, \sigma_i'\rangle)_i$ in a process represents a possibility of executing the step $\langle\sigma_i, \sigma_i'\rangle$ if the process is in the state σ_i. After this execution, the process is in the state σ_i'. Thus a sequence $(\langle\sigma_i, \sigma_i'\rangle)_i$ such that the second component of each element $\langle\sigma_i, \sigma_i'\rangle$ is the same as the first component of the next element $\langle\sigma_{i+1}, \sigma_{i+1}'\rangle$ represents a possibility of executing the steps $\langle\sigma_0, \sigma_0'\rangle, \langle\sigma_1, \sigma_1'\rangle, \ldots$, and therefore is called *executable*. In other words, a sequence is executable when it has no gaps.

DEFINITION 13 (Histories of Elements of $\mathbf{Q}_1$). Let $q \in \mathbf{Q}_1 \cup (\Sigma\times\Sigma)^{<\omega}$.

(1) The sequence q is *executable*, written $\mathrm{Exec}_1(q)$, iff

$$\begin{aligned} &\exists\nu \in \omega \cup \{\omega\}, \\ &\exists(\langle\sigma_i, \sigma_i'\rangle)_{i\in\nu}\,[q = (\langle\sigma_i, \sigma_i'\rangle)_{i\in\nu} \wedge \forall i \in \nu[i+1 \in \nu \Rightarrow \sigma_i' = \sigma_{i+1}]] \\ &\vee\ \exists k \in \omega, \exists(\langle\sigma_i, \sigma_i'\rangle)_{i\in k}, \\ &\exists\sigma_k[q = (\langle\sigma_i, \sigma_i'\rangle)_{i\in k}\cdot(\sigma_k) \wedge \forall i \in k[\sigma_i' = \sigma_{i+1}]]. \end{aligned}$$

Let $\mathbf{E}_1 = \{q \in \mathbf{Q}_1 \cup (\Sigma\times\Sigma)^{<\omega} : \mathrm{Exec}_1(q)\}$. For $\sigma \in \Sigma$, let $\mathbf{E}_1\langle\sigma\rangle = \{q \in \mathbf{E}_1\backslash\{\varepsilon\} : \mathrm{istate}_1(q) = \sigma\}$.

(2) Let q be executable. The *history* of q, denoted by $\mathrm{hist}_1(q)$, is defined by

$$\mathrm{hist}_1(q) = \begin{cases} (\sigma_i')_{i\in\nu} & \text{if } q = (\langle\sigma_i, \sigma_i'\rangle)_{i\in\nu}, \\ (\sigma_i')_{i\in k} & \text{if } q = (\langle\sigma_i, \sigma_i'\rangle)_{i\in k}\cdot(\sigma_k). \end{cases}$$

Now we can give another formulation of α_1 as follows:

LEMMA 9 (Another Formulation of Abstraction Function α_1). (1) *For* $p \in \mathbf{P}_1^*$, $\sigma \in \Sigma$, *one has* $\alpha_1(p)(\sigma) = \{\mathrm{hist}_1(q) : q \in p \cap \mathbf{E}_1\langle\sigma\rangle\}$.

(2) $\forall k \geqslant 1, \forall p_1, \ldots, p_k \in \mathbf{P}_1^*, \forall\sigma[\alpha_1(\bigcup_{i\in\bar{k}}[p_i])(\sigma) = \bigcup_{i\in\bar{k}}[\alpha_1(p_i)(\sigma)]]$.

Proof. Omitted (see Appendix 3 of [HBR90]). ∎

By means of this lemma, one has the correctness of $\mathscr{D}_1$.

LEMMA 10 (Correctness of $\mathscr{D}_1$). (1) $\alpha_1 \circ \tilde{\mathcal{O}}_1 = \mathcal{O}_1$.

(2) $\alpha_1 \circ \mathscr{D}_1 = \mathcal{O}_1$.

Proof. (1) By showing that $\alpha_1 \circ \tilde{\mathcal{O}}_1$ is the fixed point of $\Psi_1^{\mathcal{O}}$ defined in Definition 6.

(2) Immediate from (1) and Lemma 7 (2). ∎

3.5. *Full Abstraction of $\mathscr{D}_1$ with Respect to $\mathcal{O}_1$*

The full abstraction of $\mathscr{D}_1$ is shown by means of a context with parallel composition:

> Given two statements $s_1, s_2 \in \mathscr{L}_1$ with different denotational meanings, a suitable statement T called a *tester* is constructed such that the operational meanings of $s_1 \parallel T$ and $s_2 \parallel T$ are distinct. (6)

A combinatorial method for constructing such a tester is proposed in Lemma 13 (Testing Lemma). Using this method, we can construct testers having the following property:

> Given a process and a finite sequence $r = (\langle\sigma_1, \sigma'_1\rangle, ..., \langle\sigma_n, \sigma'_n\rangle)$, we can construct a tester T and an executable sequence $\tilde{r} = (\langle\tilde{\sigma}_1, \tilde{\sigma}'_1\rangle, ..., \langle\tilde{\sigma}_k, \tilde{\sigma}'_k\rangle)$ with $k \geqslant n$ such that for every process p', the parallel composition $p' \tilde{\parallel} \mathscr{D}_1[\![T]\!]$ can execute $\tilde{r}$ if there is some sequence q such that $(\langle\sigma_1, \sigma'_1\rangle, ..., \langle\sigma_n, \sigma'_n\rangle) \cdot q \in p'$, i.e., $p'[\langle\sigma_1, \sigma'_1\rangle, ..., \langle\sigma_n, \sigma'_n\rangle] \neq \varnothing$, and the converse of this holds for $p' = p$. Intuitively, for such T and $\tilde{r}$, the process p is *forced* to execute the steps $\langle\sigma_1, \sigma'_1\rangle, ..., \langle\sigma_n, \sigma'_n\rangle$ (perhaps not consecutively but in this order) when $p \tilde{\parallel} \mathscr{D}_1[\![T]\!]$ executes the steps $(\langle\tilde{\sigma}_1, \tilde{\sigma}'_1\rangle, ..., \langle\tilde{\sigma}_k, \tilde{\sigma}'_k\rangle)$ consecutively.

By the above property, we can construct such testers T as in (6) as follows:

> If s_1 and s_2 are distinct in their denotational meanings, then, putting $p_i = \mathscr{D}_1[\![s_i]\!]$ $(i = 1, 2)$, there exists some sequence r such that $p_1[r] \neq \varnothing$ but $p_2[r] = \varnothing$ (or vice versa). By constructing a tester T and an executable sequence $\tilde{r}$ for r and $p = p_2$ as above, one has $\tilde{r} \in \mathscr{D}_1[\![s_1]\!] \tilde{\parallel} \mathscr{D}_1[\![T]\!]$ and $\tilde{r} \notin \mathscr{D}_1[\![s_2]\!] \tilde{\parallel} \mathscr{D}_1[\![T]\!]$. Thus one has a difference between the operational meanings of the two statements $s_1 \parallel T$ and $s_2 \parallel T$.

First, the notion of *full abstraction* is defined:

DEFINITION 14 (Full Abstraction). Let $\mathscr{L}$ be a language and $\mathscr{O}$ an operational model for $\mathscr{L}$. A denotational model $\mathscr{D}$ is said to be fully abstract w.r.t. the operational model $\mathscr{O}$ iff for every $s_1, s_2 \in \mathscr{L}_1$, one has $\forall \xi \in \mathrm{SVar}, \forall S \in \mathscr{L}_1^{\xi}[\mathscr{O}[\![S_{(\xi)}[s_1]]\!] = \mathscr{O}[\![S_{(\xi)}[s_2]]\!]] \Leftrightarrow \mathscr{D}[\![s_1]\!] = \mathscr{D}[\![s_2]\!]$.

For a language $\mathscr{L}$ which can be formulated as the set of terms generated by a single-sorted signature, and an operational model $\mathscr{O}$ for it, a fully abstract compositional model for $\mathscr{L}$ w.r.t. $\mathscr{O}$ is unique in the following sense and exists if $\mathscr{L}$ has no recursion, as was shown in [BKO88].

LEMMA 11 (Uniqueness and Existence of Fully Abstract Compositional Model). *If two compositinal models $\mathscr{D}$ and $\mathscr{D}'$ are fully abstract w.r.t. $\mathscr{O}$,* *then there is an isomorphism from* $\mathscr{D}[\mathscr{L}]$ *to* $\mathscr{D}'[\mathscr{L}]$; *i.e., there is a bijection* $\varphi : \mathscr{D}[\mathscr{L}] \to \mathscr{D}'[\mathscr{L}]$; *i.e., there is a bijection* $\varphi : \mathscr{D}[\mathscr{L}] \to \mathscr{D}'[\mathscr{L}]$ *such that for every operator* F *in* $\mathscr{L}$ *with arity* r, *and* $p_1, ..., p_r \in \mathscr{D}[\mathscr{L}]$, *one has* $\varphi(F^D(p_1, ..., p_r)) = F^{\mathscr{D}'}(\varphi(p_1), ..., \varphi(p_r))$. *In other words, the fully abstract compositional model is unique except for isomorphism.*

Moreover, there exists a fully abstract compositional model, if $\mathscr{L}$ *has no recursion.*

Proof. See Proposition 7.1.1 of [BKO88]. ∎

Let us proceed to establish the full abstraction of $\mathscr{D}_1$ w.r.t. $\mathscr{O}_1$, stated by the following theorem, under the assumption that **V** is *infinite*. The reader might expect that the same result can be obtained without this assumption, but it is necessary. In fact, if **V** is *finite*, then $\mathscr{D}_1$ is *not* fully abstract w.r.t. $\mathscr{O}_1$ (see Example 3 in Section 3.6.1).

THEOREM 1 (Full Abstraction of $\mathscr{D}_1$). *Let* **V** *be infinite. Then, for every* $s_1, s_2 \in \mathscr{L}_1$, *one has*

$$\mathscr{D}_1[\![s_1]\!] \neq \mathscr{D}_1[\![s_2]\!] \Rightarrow \exists T \in \mathscr{L}_1[\alpha_1(\mathscr{D}_1[\![s_1]\!] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!]) \neq \alpha_1(\mathscr{D}_1[\![s_2]\!] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])].$$

To establish Theorem 1, we present the following lemma, from which Theorem 1 follows easily. (In the remainder of this paper, we fix an element $\bar{v}$ of **V**, and for $\mathscr{X} \in \wp_f(\text{IVar})$ we set $\Sigma_{\mathscr{X}} = \{\sigma \in \Sigma : \forall x \in (\text{IVar} \backslash \mathscr{X})\ [\sigma(x) = \bar{v}]\}$.)

LEMMA 12 (Uniform Distinction Lemma for $\mathscr{L}_1$). *Let* $\mathscr{X} \in (\wp_f(\text{IVar}) \backslash \{\varnothing\})$.

(1) *For every* $r \in (\Sigma_{\mathscr{X}} \times \Sigma_{\mathscr{X}})^{<\omega}$,

$$\forall p_1, p_2 \in \mathbf{P}_1^*[p_1[r] \neq \varnothing \wedge p_2[r] = \varnothing \Rightarrow \forall \sigma_0 \in \Sigma_{\mathscr{X}}, \exists T \in \mathscr{L}_1[\alpha_1(p_1 \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])(\sigma_0) \backslash \alpha_1(p_2 \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])(\sigma_0) \neq \varnothing]]. \tag{7}$$

(2) *For every* $q \in (\Sigma_{\mathscr{X}} \times \Sigma_{\mathscr{X}})^{<\omega} \cdot \Sigma_{\mathscr{X}}$,

$$\forall p_1, p_2 \in \mathbf{P}_1^*[q \in p_1 \backslash p_2 \Rightarrow \forall \sigma_0 \in \Sigma_{\mathscr{X}}, \exists T \in \mathscr{L}_1[\alpha_1(p_1 \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])(\sigma_0) \backslash \alpha_1(p_2 \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])(\sigma_0) \neq \varnothing]]. \tag{8}$$

Proof of Theorem 1. Let $p_1 = \mathscr{D}_1[\![s_1]\!]$, $p_2 = \mathscr{D}_1[\![s_2]\!]$, and suppose $p_1 \neq p_2$. We can assume, without loss of generality, that there exists q such that $q \in p_1$ and $q \notin p_2$. The proof is given by distinguishing two cases according to whether q is infinite or finite.

Case 1. Suppose q is infinite. First, let us show by contradiction that there is an $n \in \omega$ such that $q^{[n]} \notin (p_2)^{[n]}$. Assume, to the contrary, that $\forall n \in \omega[p_2[q^{[n]}] \neq \emptyset]$. Then, by the closedness of p_2, one has $q \in p_2$, which contradicts the fact $q \notin p_2$. Hence, there is $n \in \omega$ such that $p_2[q^{[n]}] = \emptyset$. From the fact that $\mathrm{FIR}_1(p_i)$ $(i = 1, 2)$ and from Remark 3(2), it follows that there is an $\mathscr{X} \in (\wp_f(\mathrm{IVar}) \backslash \{\emptyset\})$ such that (3) holds for $p = p_i$ $(i = 1, 2)$. Fix such an $\mathscr{X}$, and let $\bar{\sigma} = (\lambda x \in (\mathrm{IVar} \backslash \mathscr{X}) : \bar{v})$ and $r = (\langle (\pi_j^2(q(i)) \upharpoonright \mathscr{X}) \cup \bar{\sigma} \rangle_{j \in 2})_{i \in n}$. Then $r \in (\Sigma_{\mathscr{X}} \times \Sigma_{\mathscr{X}})^n$. It follows from the fact that $q^{[n]} \in ((p_1)^{[n]} \backslash (p_2)^{[n]})$ and (3), for $p = p_i$ $(i = 1, 2)$, that $r \in ((p_1)^{[n]} \backslash (p_2)^{[n]})$. Thus applying Lemma 12(1), one has $\exists T \in \mathscr{L}_1[\alpha_1(\mathscr{D}_1[\![s_1]\!] \,\tilde{\|}\, \mathscr{D}_1[\![T]\!]) \backslash \alpha_1(\mathscr{D}_1[\![s]\!] \,\tilde{\|}\, \mathscr{D}_1[\![T]\!]) \neq \emptyset]$.

Case 2. Suppose q is finite. Then one obtains the same result in a similar fashion to that for Case 1, but using Lemma 12(2) instead of Lemma 12(1) used in Case 1. ∎

3.5.1. *Proof of Lemma* 12

Testers for proving Lemma 12(1) (resp. Lemma 12(2)) are constructed by induction on the length $r \in (\Sigma_{\mathscr{X}} \times \Sigma_{\mathscr{X}})^{<\omega}$ (resp. $q \in (\Sigma_{\mathscr{X}} \times \Sigma_{\mathscr{X}})^{<\omega} \cdot \Sigma_{\mathscr{X}}$). The following lemma is used to construct testers for r (or q) with length $n+1$ by means of testers for r (or q) with length n. The assumption that $\mathbf{V}$ is infinite will be essentially used in the proof of Lemma 13.

LEMMA 13 (Testing Lemma for $\mathscr{L}_1$). *Let* $\mathscr{X} \in (\wp_f(\mathrm{IVar}) \backslash \{\emptyset\})$, $p \in \mathbf{P}_1^*$, *and* $\sigma', \sigma'', \sigma_0 \in \Sigma_{\mathscr{X}}$. *Then there are two finite sequences* $r_1, r_2 \in (\Sigma_{\mathscr{X}} \times \Sigma_{\mathscr{X}})^{<\omega}$ *such that the following hold*:

(1) $r_1 \cdot \langle \sigma', \sigma'' \rangle \cdot r_2 \in \mathbf{E}_1 \langle \sigma_0 \rangle$.

(2) *For every tester* $T' \in \mathscr{L}_1$, *there exists another tester* $T \in \mathscr{L}_1$ *such that the following hold*:

(i) $\mathscr{D}_1[\![T]\!][r_1 \cdot r_2] = \mathscr{D}_1[\![T']\!]$,

(ii) *The process* p *is forced to execute the step* $\langle \sigma', \sigma'' \rangle$ *and forbidden to execute any other steps when the parallel composition* $p \,\tilde{\|}\, \mathscr{D}_1[\![T]\!]$ *executes the sequence*: $r_1 \cdot \langle \sigma', \sigma'' \rangle \cdot r_2$. *That is, the following holds for every* $q' \in \mathbf{Q}_1$:

$$r_1 \cdot \langle \sigma', \sigma'' \rangle \cdot r_2 \cdot q' \in p \,\tilde{\|}\, \mathscr{D}_1[\![T]\!] \Rightarrow p[\langle \sigma', \sigma'' \rangle] \neq \emptyset \wedge q' \in p[\langle \sigma', \sigma'' \rangle] \,\tilde{\|}\, \mathscr{D}_1[\![T']\!]. \tag{9}$$

The proof of this lemma will be given later. First, we will prove the following corollary, and thereby, Lemma 12.

COROLLARY 1. *Let* $\mathscr{X} \in (\wp_f(\text{IVar}) \backslash \{\emptyset\})$, $p \in \mathbf{P}_1^*$, $\langle\sigma', \sigma''\rangle \in \Sigma_{\mathscr{X}} \times \Sigma_{\mathscr{X}}$, *and* $\sigma_0 \in \Sigma_{\mathscr{X}}$. *Then there are two finite sequences* $\rho_1, \rho_2 \in (\Sigma_{\mathscr{X}})^{<\omega}$ *such that for every tester* $T' \in \mathscr{L}_1$ *there exists another tester* $T \in \mathscr{L}_1$ *such that, putting* $\sigma_1 = \text{last}(\rho_1 \cdot \sigma'' \cdot \rho_2)$, *the following hold*:

(1) *For every* $p' \in \mathbf{P}_1^*$, *one has*

$$\forall \rho' \in \Sigma^{\leqslant\omega}[p'[\langle\sigma', \sigma''\rangle] \neq \emptyset \wedge \rho' \in \alpha_1(p'[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T']\!])(\sigma_1)$$
$$\Rightarrow \rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p' \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])(\sigma_0)]. \tag{10}$$

(2) *For* $p' = p$, *the converse of* (10) *holds. That is,*

$$\forall \rho' \in \Sigma^{\leqslant\omega}[\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])(\sigma_0)$$
$$\Rightarrow p[\langle\sigma', \sigma''\rangle] \neq \theta \wedge \rho' \in \alpha_1(p[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T']\!])(\sigma_1)]. \tag{11}$$

Proof. Take r_1, r_2 as in Lemma 13, and put $\rho_1 = \text{hist}_1(r_1)$, $\rho_2 = \text{hist}_1(r_2)$, and let $\sigma_1 = \text{last}(\rho_1 \cdot \sigma'' \cdot \rho_2)$. Also, for T', take T as in Lemma 13.

Part (1). Let $p' \in \mathbf{P}_1^*$, and $\rho' \in \Sigma^{\leqslant\omega}$. Suppose $p'[\langle\sigma', \sigma''\rangle] \neq \emptyset$ and $\rho' \in \alpha_1(p'[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T']\!])(\sigma_1)$. Then, by Lemma 9(1), there exists $q' \in (p'[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T']\!])$ such that $q' \in \mathbf{E}_1\langle\sigma_1\rangle \wedge \text{hist}_1(q') = \rho'$. Fix such q'. By Lemma 13(1), one has $r_1 \cdot \langle\sigma', \sigma''\rangle \cdot r_2 \cdot q' \in \mathbf{E}_1\langle\sigma_0\rangle$. By Lemma 13(2)(i), $q' \in (p'[\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!][r_1 \cdot r_2])$. Thus, applying the $\Leftarrow$-part of Lemma 5(1) successively, one has $r_2 \cdot q' \in (p'[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!][r_1])$, $\langle\sigma', \sigma''\rangle \cdot r_2 \cdot q' \in (p' \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!][r_1])$, and $r_1 \cdot \langle\sigma', \sigma''\rangle \cdot r_2 \cdot q' \in (p' \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])$. Hence, $\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' = \text{hist}_1(r_1 \cdot \langle\sigma', \sigma''\rangle \cdot r_2 \cdot q') \in \alpha_1(p' \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])(\sigma_0)$.

Part (2). Let $\rho' \in \Sigma^{\leqslant\omega}$, and suppose $\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p \mathbin{\tilde{\|}} \mathscr{D}_1[\![T]\!])(\sigma_0)$. Then, by Lemma 9(1), there exists q' such that $(*)$: $q' \in \mathbf{E}_1\langle\sigma_1\rangle \wedge \text{hist}_1(q') = \rho'$. Fix such q'. By (9), one has $p[\langle\sigma', \sigma''\rangle] \neq \emptyset$ and $q' \in p[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T']\!]$. Thus, by $(*)$, one has $\rho' = \text{hist}_1(q') \in \alpha_1(p[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T']\!])(\sigma_1)$. ∎

Proof of Lemma 12. Let $\mathscr{X} \in (\wp_f(\text{IVar}) \backslash \{\emptyset\})$.

Part (1). We will prove that (7) holds for every $r \in (\Sigma_{\mathscr{X}} \times \Sigma_{\mathscr{X}})^{<\omega}$ by induction on the length of r.

Induction Base. Let $\text{lgt}(r) = 0$, i.e., let $r = \varepsilon$, and let $p_1, p_2 \in \mathbf{P}_1^*$. Then one has (7) vacuously, since $\forall p \in \mathbf{P}_1^*[p[\varepsilon] = p \neq \emptyset]$, and therefore it does not hold that $p_1[r] \neq \emptyset \wedge p_2[r] = \emptyset$.

Induction Step. Let $k \in \omega$, and assume that the claim holds for every r such that $\text{lgt}(r) \leqslant k$. Fix an arbitrary sequence r of length $k+1$, say

$r=\langle\sigma',\sigma''\rangle\cdot\tilde{r}$. Let $p_1,p_2\in\mathbf{P}_1^*$ such that (*): (i) $p_1[r]\neq\emptyset$, (ii) $p_2[r]=\emptyset$. Finally let $\sigma_0\in\Sigma_{\mathscr{X}}$. We distinguish two cases according to whether $p_2[\langle\sigma',\sigma''\rangle]=\emptyset$ or not.

Case 1. Suppose $p_2[\langle\sigma',\sigma''\rangle]=\emptyset$. Then, applying Corollary 1 with $p=p_2$ and $T'\equiv\mathbf{0}$, there are ρ_1, ρ_2, T such that:

$$\begin{cases}\text{(i)} & \forall p\in\mathbf{P}_1^*,\forall\rho'\in\Sigma^{\leqslant\omega}[p[\langle\sigma',\sigma''\rangle]\\ & \quad\neq\emptyset\wedge\rho'\in\alpha_1(p[\langle\sigma',\sigma''\rangle]\,\tilde{\|}\,\tilde{\mathbf{0}}_1)(\sigma_1)\\ & \quad\Rightarrow\rho_1\cdot\sigma''\cdot\rho_2\cdot\rho'\in\alpha_1(p\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)],\\ \text{(ii)} & \forall\rho'\in\Sigma^{\leqslant\omega}[\rho_1\cdot\sigma''\cdot\rho_2\cdot\rho'\in\alpha_1(p_2\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)\\ & \quad\Rightarrow p_2[\langle\sigma',\sigma''\rangle]\neq\emptyset\wedge\rho'\in\alpha_1(p_2[\langle\sigma',\sigma''\rangle]\,\tilde{\|}\,\tilde{\mathbf{0}}_1)(\sigma_1)],\end{cases}\tag{12}$$

where $\sigma_1=\text{last}(\rho_1\cdot\sigma''\cdot\rho_2)$. By (*)(i), there exists $\rho'\in\alpha_1(\rho\in\alpha_1(p_1[r]\,\tilde{\|}\,\tilde{\mathbf{0}}_1)$ (σ_1). Let us fix such a ρ'. By (12)(i) for $p=p_1$, one has $\rho_1\cdot\sigma''\cdot\rho_2\cdot\rho'\in$ $\alpha_1(p_1\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)$. Next, assume (for the sake of contradiction) that $\rho_1\cdot\sigma''\cdot\rho_2\cdot\rho'\in\alpha_1(p_2\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)$. Then, by (12)(ii), one has $p_2[\langle\sigma',\sigma''\rangle]$ $\neq\emptyset$, which contradicts the fact that $p_2[\langle\sigma',\sigma''\rangle]=\emptyset$. Hence, $\rho_1\cdot\sigma''\cdot\rho_2\cdot$ $\rho'\notin\alpha_1(p_2\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)$.

Case 2. Suppose $p_2[\langle\sigma',\sigma''\rangle]\neq\emptyset$, and let us denote $p_1[\langle\sigma',\sigma''\rangle]$ and $p_2[\langle\sigma',\sigma''\rangle]$ by p_1' and p_2', respectively. Then, one has, by (*), that (†): $p_1'[\tilde{r}]\neq\emptyset\wedge p_2'[\tilde{r}]=\emptyset$. Applying Corollary 1 with $p=p_2$, there are ρ_1, ρ_2 such that for every $T'\in\mathscr{L}_1$ there exists T satisfying

$$\begin{cases}\text{(i)} & \forall p\in\mathbf{P}_1^*,\forall\rho'\in\Sigma^{\leqslant\omega}[p[\langle\sigma',\sigma''\rangle]\\ & \quad\neq\emptyset\wedge\rho'\in\alpha_1(p[\langle\sigma',\sigma''\rangle]\,\tilde{\|}\,\mathscr{D}_1[\![T']\!])(\sigma_1)\\ & \quad\Rightarrow\rho_1\cdot\sigma''\cdot\rho_2\cdot\rho'\in\alpha_1(p\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)],\\ \text{(ii)} & \forall\rho'\in\Sigma^{\leqslant\omega}[\rho_1\cdot\sigma''\cdot\rho_2\cdot\rho'\in\alpha_1(p_2\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)\\ & \quad\Rightarrow p_2[\langle\sigma',\sigma''\rangle]\neq\emptyset\wedge\rho'\in\alpha_1(p_2[\langle\sigma',\sigma''\rangle]\,\tilde{\|}\,\mathscr{D}_1[\![T']\!])(\sigma_1)],\end{cases}\tag{13}$$

where $\sigma_1=\text{last}(\rho_1\cdot\sigma''\cdot\rho_2)$. By the induction hypothesis and (†), there are T_0 and ρ' such that

$$\rho'\in\alpha_1(p_1'\,\tilde{\|}\,\mathscr{D}_1[\![T_0]\!])(\sigma_1)\backslash\alpha_1(p_2'\,\tilde{\|}\,\mathscr{D}_1[\![T_0]\!])(\sigma_1).\tag{14}$$

Let $\rho=\rho_1\cdot\sigma''\cdot\rho_2\cdot\rho'$, and take T such that (13) holds for $T'=T_0$. By (13)(i) for $p=p_1$ and (14), one has $\rho\in\alpha_1(p_1\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)$. Next, assume (to obtain a contraction) that $\rho_1\cdot\sigma''\cdot\rho_2\cdot\rho'\in\alpha_1(p_2\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)$. Then, it follows from (13)(ii) that $\rho'\in\alpha_1(p_2'\,\tilde{\|}\,\mathscr{D}_1[\![T_0]\!])(\sigma_1)$, which contradicts (14). Thus, $\rho_1\cdot\sigma''\cdot\rho_2\cdot\rho'\notin\alpha_1(p_2\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)$. Summing up, in this case too there is a ρ such that $\rho\in\alpha_1(p_1\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)\backslash\alpha_1(p_2\,\tilde{\|}\,\mathscr{D}_1[\![T]\!])(\sigma_0)$.

Part 2. In order to establish part (2), we will prove that (8) holds for every $q \in (\Sigma_{\mathscr{X}} \times \Sigma_{\mathscr{X}})^{<\omega} \cdot \Sigma_{\mathscr{X}}$, by induction on the length of q.

Induction Base. Let $\mathrm{lgt}(q) = 1$, say $q = (\sigma')$. Let $p_1, p_2 \in \mathbf{P}_1^*$ such that $q \in p_1 \setminus p_2$, and let $\sigma_0 \in \Sigma_{\mathscr{X}}$. Since $\mathscr{X}$ is finite and nonempty, we can put $\mathscr{X} = \{x_1, ..., x_r\}$. Then, let us set $T \equiv (x_1 := \sigma'(x_1)); ...; (x_r := \sigma'(x_r)); \mathbf{0}$, and $t = \mathscr{D}_1[\![T]\!]$. By the definition of $\tilde{\|}$, one has $(\langle \sigma'_0, \sigma'_1 \rangle, ..., \langle \sigma'_{r-1}, \sigma'_r \rangle, \sigma') \in p_1 \tilde{\|}\, t$, i.e., $(\sigma'_1, ..., \sigma'_r) \in \alpha_1(p_1 \tilde{\|}\, t)(\sigma_0)$, where $\sigma'_i = \sigma_0[(\sigma'(x_1), ..., \sigma'(x_i))/(x_1, ..., x_i)]$ $(i \in r+1)$. Let us prove, by contradiction, that $(\langle \sigma'_0, \sigma'_1 \rangle, ..., \langle \sigma'_{r-1}, \sigma'_r \rangle, \sigma') \notin p_2 \tilde{\|}\, t$. Indeed, if $(\langle \sigma'_0, \sigma'_1 \rangle, ..., \langle \sigma'_{r-1}, \sigma'_r \rangle, \sigma') \in p_2 \tilde{\|}\, t$, then the first r-steps $\langle \sigma'_0, \sigma'_1 \rangle, ..., \langle \sigma'_{r-1}, \sigma'_r \rangle$ must stem from the right-hand side t. Thus, it must hold that $(\sigma') \in p_2 \tilde{\|}\, t[(\langle \sigma'_0, \sigma'_1 \rangle, ..., \langle \sigma'_{r-1}, \sigma'_r \rangle)] = p_2 \tilde{\|}\, \tilde{\mathbf{0}}_1$. However, this is impossible since $(\sigma') \notin p_2$. Summing up, one has $(\langle \sigma'_0, \sigma'_1 \rangle, ..., \langle \sigma'_{r-1}, \sigma'_r \rangle, \sigma') \notin p_2 \tilde{\|}\, t$, i.e., $(\sigma'_1, ..., \sigma'_r) \notin \alpha_1(p_2 \tilde{\|}\, t)(\sigma_0)$.

Induction Step: Similar to the induction step of part (1). ∎

Finally, let us prove Lemma 13. A crucial ingredient of the proof is the fact that the value of a variable can be changed from any value to any other value in *one atomic step*, by means of an assignment statement.

Proof of Lemma 13. The proof is formulated by supposing that $\mathscr{X}$ is reduced to one variable, $\mathscr{X} = \{x\}$, which simplifies the proof, allowing us to identify a state $\sigma \in \Sigma_{\mathscr{X}}$ with its value $\sigma(x) \in \mathbf{V}$. However, the lemma still holds when $\mathscr{X}$ is composed of more than one variable, as established in Appendix 4 of [HBR90]. For $v \in \mathbf{V}$, let $\bar{\sigma}(v) = (\lambda y \in \mathrm{IVar} : \mathrm{if}(y = x, v, \bar{v}))$.

Trying to construct a desired tester T, we first observe that the composition $p \tilde{\|}\, \mathscr{D}_1[\![T]\!]$ must be in the state σ' when p executes the step $\langle \sigma', \sigma'' \rangle$. Therefore, if $\sigma_0(x) \neq \sigma'(x)$, then $\mathscr{D}_1[\![T]\!]$ must execute the step $\langle \sigma, \sigma' \rangle$ for some σ, and therefore, T must have an assignment "$x := \sigma'(x)$" in it. Moreover, we need a trick for forbidding p to execute the step $\langle \sigma, \sigma' \rangle$ instead of $\mathscr{D}_1[\![T]\!]$ and forbidding $\mathscr{D}_1[\![T]\!]$ to execute the step $\langle \sigma', \sigma'' \rangle$ instead of p. The proof of Lemma 13 is given by distinguishing two cases according to whether $\sigma_0(x) = \sigma'(x)$.

Case 1. When $\sigma_0(x) = \sigma'(x)$, we can easily construct two sequences r_1, r_2 satisfying (1) and (2) of Lemma 13 as follows: Let $r_1 = \varepsilon$, $r_2 = \langle \sigma'', \bar{\sigma}(v_1) \rangle$, where v_1 is chosen such that

$$\text{(i)} \quad v_1 \neq \sigma''(x), \qquad \text{(ii)} \quad v_1 \notin \{v \in \mathbf{V} : \langle \sigma', \sigma'' \rangle \cdot \langle \sigma'', \bar{\sigma}(v) \rangle \in p^{[2]}\}. \qquad (15)$$

Note that the right-hand side of (15)(ii) is finite since p is image finite by Definition 10, and therefore, there is a v_1 satisfying (15). It is immediate that Lemma 13(1) holds. Let us show Lemma 13(2). For every $T' \in \mathscr{L}_1$, let $T \equiv (x := v_1); T'$. It is immediate that (2)(i) holds. Let us show (2)(ii), i.e., that (9) holds for every $q' \in \mathbf{Q}_1$.

Suppose $\langle\sigma', \sigma''\rangle \cdot \langle\sigma'', \bar{\sigma}(v_1)\rangle \cdot q' \in p \tilde{\|} \mathscr{D}_1[\![T]\!]$. Let us show that the first two steps, $\langle\sigma', \sigma''\rangle$ and $\langle\sigma'', \bar{\sigma}(v_1)\rangle$, must stem from p and $\mathscr{D}_1[\![T]\!]$, respectively. The first step cannot stem from $\mathscr{D}_1[\![T]\!]$ by (15)(i). Also, the second step cannot stem from p by (15)(ii). Thus one has the desired result.

Case 2. When $\sigma_0(x) \neq \sigma'(x)$, we can construct two sequences r_1, r_2 satisfying (1) and (2) of Lemma 13 as follows. Let $r_1 = \langle\sigma_0, \sigma'\rangle$ and $r_2 = \langle\sigma'', \bar{\sigma}(v_1)\rangle$, where v_1 is chosen such that

$$\begin{cases} \text{(i)} & v_1 \notin \{v \in \mathbf{V} : \langle\sigma_0, \sigma''\rangle \cdot \langle\sigma', \sigma''\rangle \cdot \langle\sigma'', \bar{\sigma}(v)\rangle \in p^{[3]}\}, \\ \text{(ii)} & v_1 \neq \sigma'(x), \\ \text{(iii)} & v_1 \neq \sigma''(x), \\ \text{(iv)} & v_1 \notin \{v \in \mathbf{V} : \langle\sigma', \sigma''\rangle \cdot \langle\sigma'', \bar{\sigma}(v)\rangle \in p^{[2]}\}. \end{cases} \tag{16}$$

Note that the right-hand sides of (16)(i) and (iv) are finite, since p is image finite by Definition 10, and therefore, there is v_1 satisfying (16). It is immediate that (1) holds. Let us show (2), i.e., that for every $T' \in \mathscr{L}_1$, there exists $T \in \mathscr{L}_1$ satisfying (2)(i), (2)(ii). A tester T with these properties can be constructed in the following format: For $\bar{v}_0$, $\bar{v}'$, $\bar{v}_1$, $\bar{v}_2 \in \mathbf{V}$, and $s \in \mathscr{L}_1$, let

$$F(\bar{v}_0, \bar{v}', \bar{v}_1, \bar{v}_2, s) \equiv \text{If}(x = \bar{v}_0, (x := \bar{v}'); (x := \bar{v}_1); s, (x := \bar{v}_2); \mathbf{0}). \tag{17}$$

We set $T \equiv F(\sigma_0(x), \sigma'(x), v_1, v_2, T')$, where v_2 is chosen such that (∗): (i) $v_2 \neq \sigma''(x)$, (ii) $v_2 \neq v_1$. In this case also, it is immediate that (2)(i) holds. Let us show (2)(ii), i.e., that (9) holds for every $q' \in \mathbf{Q}_1$. First, put $t' = \mathscr{D}_1[\![T']\!]$, $t = \mathscr{D}_1[\![T]\!]$.

Suppose $\langle\sigma_0, \sigma'\rangle \cdot \langle\sigma', \sigma''\rangle \cdot \langle\sigma'', \bar{\sigma}(v_1)\rangle \cdot q' \in p \tilde{\|} t$. Let us show that the first three steps, $\langle\sigma_0, \sigma'\rangle$, $\langle\sigma', \sigma''\rangle$, $\langle\sigma'', \bar{\sigma}(v_1)\rangle$, must stem from t, p, t, respectively.

First, let us show by contradiction that the first step $\langle\sigma_0, \sigma'\rangle$ cannot step from p. Assume that the first step stems from p, i.e., that $\langle\sigma', \sigma''\rangle \cdot \langle\sigma'', \bar{\sigma}(v_1)\rangle \cdot q' \in p[\langle\sigma_0, \sigma'\rangle] \tilde{\|} t$. Then the second step $\langle\sigma', \sigma''\rangle$ must stem from either of $p[\langle p_0, \sigma'\rangle]$ or t; Let us show that it can stem from neither of them. Suppose that the second step stems from t, i.e., $\langle\sigma'', \bar{\sigma}(v_1)\rangle \cdot q' \in p[\langle\sigma_0, \sigma'\rangle] \tilde{\|} t[\langle\sigma', \sigma''\rangle]$. Then $t[\langle\sigma', \sigma''\rangle] \neq \varnothing$, and therefore, under the assumption that $\sigma_0(x) \neq \sigma'(x)$, the assignment "$x := v_2$" must be executed in the second step, which yields $\sigma''(x) = v_2$. However, this contradicts (∗)(i). Thus $\langle\sigma'', \bar{\sigma}(v_1)\rangle \cdot q' \in p[\langle\sigma_0, \sigma'\rangle \cdot \langle\sigma', \sigma''\rangle] \tilde{\|} t$. The third step $\langle\sigma'', \bar{\sigma}(v_1)\rangle$ cannot step from $p[\langle\sigma_0, \sigma'\rangle \cdot \langle\sigma', \sigma''\rangle]$, since, by (16)(i), $p[\langle\sigma_0, \sigma'\rangle \cdot \langle\sigma', \sigma''\rangle \cdot \langle\sigma'', \bar{\sigma}(v_1)\rangle] = \varnothing$. Thus the third step must stem from t, which implies $v_1 = \sigma'(x)$ or $v_1 = v_2$. However, both are impossible by (16)(ii) and (∗)(ii), respectively. Summing up, the first step cannot stem from p, and therefore, it must stem from t. Thus one has $\langle\sigma', \sigma''\rangle \cdot \langle\sigma'', \bar{\sigma}(v_1)\rangle \cdot q' \in p \mathbin{\tilde{\|}} t[\langle\sigma_0, \sigma'\rangle]$.

Next let us show the second step $\langle\sigma', \sigma''\rangle$ cannot stem from $t[\langle\sigma_0, \sigma'\rangle]$. If it stems from $t[\langle\sigma_0, \sigma'\rangle]$, then $t[\langle\sigma_0, \sigma'\rangle \cdot \langle\sigma', \sigma''\rangle] \neq \emptyset$, which implies, by the form of T, that $\sigma''(x) = v_1$. This contradicts (16)(iii). Thus the second step must stem from p, and therefore, $\langle\sigma'', \bar{\sigma}(v_1)\rangle \cdot q' \in p[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} t[\langle\sigma_0, \sigma'\rangle]$.

Finally, the third step $\langle\sigma'', \bar{\sigma}(v_1)\rangle$ cannot stem from $p[\langle\sigma', \sigma''\rangle]$, since $p[\langle\sigma', \sigma''\rangle \cdot \langle\sigma'', \bar{\sigma}(v_1)\rangle] = \emptyset$, by (16)(iv). Thus the third step must stem from $t[\langle\sigma_0, \sigma'\rangle]$, and therefore, $q' \in p[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} t[\langle\sigma_0, \sigma'\rangle \cdot \langle\sigma'', \bar{\sigma}(v_1)\rangle]$, that is, $q' \in p[\langle\sigma', \sigma''\rangle] \mathbin{\tilde{\|}} \mathscr{D}_1[\![T']\!]$. ∎

Remark 4. Note that if $\sigma_0(x) \neq \sigma'(x)$ and $\sigma'(x) \neq \sigma''(x)$, then a simpler tester $T \equiv (x := \sigma'(x))$; $(x := v_1)$; T', with v_1 satisfying (16), is sufficient to establish the above lemma. However, if $\sigma_0(x) \neq \sigma'(x)$ and $\sigma'(x) = \sigma''(x)$, then we need a tester defined in the format (17) to exclude the possibility that the first three steps of the parallel composition may stem from p, t, and t, respectively.

3.6. *Comparison of $\mathscr{D}_1$ and Other Models*

3.6.1. *Comparison with a More Abstract Model than $\mathscr{D}_1$ for $\mathscr{L}_1$ with* **V** *Finite*

As stated earlier, the assumption that **V** is infinite is necessary for the full abstraction of $\mathscr{D}_1$. In fact, if **V** is *finite*, then we can construct another compositional model $\hat{\mathscr{D}}_1$ which is correct w.r.t. $\mathscr{O}_1$ and *more* abstract than $\mathscr{D}_1$. Thus $\mathscr{D}_1$ cannot be fully abstract w.r.t. $\mathscr{O}_1$. The model $\hat{\mathscr{D}}_1$ is constructed from $\mathscr{D}_1$ by abstracting from certain redundant information present in $\mathscr{D}_1$, as follows:

DEFINITION 15. Let $p \in \wp(\mathbf{Q}_1)$. (1) Let $q \in p$, and $\langle n, i\rangle \in \omega \times 2$. Let us say *$q$ is pruned away from p at place $\langle n, i\rangle$* iff q is infinite and $q \notin q^{[n+i]} \cdot (\mathbf{E}_1\langle\pi_i^2(q(n))\rangle \cap (\Sigma \times \Sigma)^\omega) \subseteq p$.

(2) A *pruning function* $\Lambda: \wp(\mathbf{Q}_1) \to \wp(\mathbf{Q}_1)$ is defined as follows: $\Lambda(p) = \{q \in p : \neg\exists\langle n, i\rangle \in \omega \times 2[q$ is pruned away from p at place $\langle n, i\rangle]\}$.

(3) For $s \in \mathscr{L}_1$, let $\hat{\mathscr{D}}_1[\![s]\!] = \Lambda(\mathscr{D}_1[\![s]\!])$.

Since *executable passes* in $\hat{\mathscr{D}}_1[\![s]\!]$ are the same as those in $\mathscr{D}_1[\![s]\!]$ $(s \in \mathscr{L}_1)$ by the definition of Λ, one has the *correctness* of $\hat{\mathscr{D}}_1$ w.r.t. $\mathscr{O}_1$:

LEMMA 14. $\alpha_1 \circ \hat{\mathscr{D}}_1 = \alpha_1 \circ \mathscr{D}_1 = \mathscr{O}_1$.

Moreover, we can show that $\hat{\mathscr{D}}_1$ is compositional w.r.t. all the operators of $\mathscr{L}_1$. For this purpose, we define another set of semantic operations from that defined in Definition 8. For each syntactical operator F with arity r of $\mathscr{L}_1$, a semantic operation $\tilde{F}$ with domain $(\mathbf{P}_1)$ has been defined in Definition 8; we can extend the domain of $\tilde{F}$ from $(\mathbf{P}_1)^r$ to $(\wp(\mathbf{Q}_1))^r$ straightforwardly except for $F \equiv \|$. As to $\|$, we can extend the domain of $\tilde{\|}$ to $(\wp(\mathbf{Q}_1))^2$ by means of a merge operation on elements of $\mathbf{Q}_1$; this operation can be defined as in [Hor91], where merge operation on infinite sequences (taking communication into account) is defined.

DEFINITION 16. (1) Let $r \in \omega$. For a meaning function $\mathscr{D}$ with $\mathrm{dom}(\mathscr{D}) = \mathscr{L}_1$, and $\vec{s} \in (\mathscr{L}_1)^r$, let $\mathscr{D}[\![\vec{s}]\!] = (\mathscr{D}[\![s(i)]\!])_{i \in r}$. Also, for a function f with $\mathrm{dom}(f) = \wp(\mathbf{Q}_1)$, and $\vec{p} \in (\wp(\mathbf{Q}_1))^r$, let $f(\vec{p}) = (f(\vec{p}(i)))_{i \in r}$.

(2) Let $\mathscr{S}_1$ be the set of syntactical operators of $\mathscr{L}_1$, and for $r \in \omega$, let $\mathscr{S}_1(r) = \{F \in \mathscr{S}_1 : \text{the arity of } F \text{ is } r\}$. Let $F \in \mathscr{S}_1(r)$ and $\tilde{F}$ be the semantic operation corresponding to F in the interpretation structure $\mathscr{I}_1$. From $\tilde{F}$, let us define another semantic operation $\hat{F}$ as follows: For every $\vec{p} \in (\wp(\mathbf{Q}_1))^r$, let $\hat{F}(\vec{p}) = \Lambda(\tilde{F}(\vec{p}))$.

From the semantic operations $\hat{F}$, one obtains the compositionality of $\hat{D}_1$ w.r.t. all the operators of $\mathscr{L}_1$:

LEMMA 15. *For every $r \in \omega$ and $F \in \mathscr{S}_1(r)$, one has $\forall \vec{s} \in (\mathscr{L}_1)^r\ [\hat{\mathscr{D}}_1[\![F(\vec{s})]\!] = \hat{F}(\hat{\mathscr{D}}_1[\![\vec{s}]\!])]$.*

Proof. Let $r \in \omega$ and $F \in \mathscr{S}_1(r)$. It can be shown that $(*)$: $\forall \vec{p} \in (\wp(\mathbf{Q}_1))^r\ [\Lambda(\tilde{F}(\vec{p})) = \Lambda(\tilde{F}(\Lambda(\vec{p})))]$. From this one obtains the desired result as follows: Let $\vec{s} \in (\mathscr{L}_1)^r$, and $\vec{p} = \mathscr{D}_1[\![\vec{s}]\!]$. Then

$$\begin{aligned}
\hat{\mathscr{D}}_1[\![F(\vec{s})]\!] &= \Lambda(\mathscr{D}_1[\![F(\vec{s})]\!]) && \text{(by the definition of } \hat{D}_1\text{)}\\
&= \Lambda(\tilde{F}(\vec{p})) && \text{(by the compositionality of } \mathscr{D}_1\text{)}\\
&= \Lambda(\tilde{F}(\Lambda(\vec{p}))) && \text{(by } (*)\text{)}\\
&= \hat{F}(\hat{\mathscr{D}}_1[\![\vec{s}]\!]) && \text{(by the definition of } \hat{D}_1 \text{ and } \hat{F}\text{).} \quad \blacksquare
\end{aligned}$$

When $\mathbf{V}$ is finite, the model $\hat{\mathscr{D}}_1$ is strictly more abstract than $\mathscr{D}_1$, as can seen from the following example. Thus $\mathscr{D}_1$ is not fully abstract in this case.

EXAMPLE 3. Assume that $\mathbf{V} = \{0, 1\}$. Moreover, let us assume, for simplicity, that $\mathrm{IVar} = \{x\}$. Then Σ is identified with $\mathbf{V}$. Let $g \equiv ((x := 0); X_0) + ((x := 1); X_0)$, and suppose $\langle X_0, g \rangle \in D$. Then, setting $s_1 \equiv X_0 + \mathrm{If}(x = 0, (x := 0); \mathrm{If}(x = 0, X_0, \mathbf{0}), X_0)$, and $s_2 \equiv \mathrm{If}(x = 0, ((x := 0); \mathrm{If}(x = 0, X_0, \mathbf{0})) + ((x := 1); X_0), X_0)$, one has $\mathscr{D}_1[\![s_1]\!][\langle 0, 0 \rangle \cdot \langle 1, 1 \rangle] \neq \varnothing$, but $\mathscr{D}_1[\![s_2]\!][\langle 0, 0 \rangle \cdot \langle 1, 1 \rangle] = \varnothing$. Thus, $(*)\ \mathscr{D}_1[\![s_1]\!] \neq \mathscr{D}_1[\![s_2]\!]$. However, by the definition of $\hat{\mathscr{D}}_1$ and Λ, one has $(\dagger)\ \hat{\mathscr{D}}_1[\![s_1]\!] = \Lambda(\mathscr{D}_1[\![s_1]\!]) = \Lambda(\mathscr{D}_1[\![s_2]\!]) = \hat{\mathscr{D}}_1[\![s_2]\!] = \{q \in \mathscr{D}_1[\![s_2]\!] : q$ is finite $\vee q$ is infinite and executable}, since if q ($\in \mathbf{Q}_1$) is infinite and executable, then $q \in \mathscr{D}_1[\![s_i]\!]$ ($i=1, 2$). Thus, for every context $S_{(\xi)} \in \mathscr{L}_1^*$, one has $\mathcal{O}_1[\![S_{(\xi)}[s_1]]\!] = \alpha_1(\hat{\mathscr{D}}_1[\![S_{(\xi)}[s_1]]\!]) = \alpha_1(\hat{\mathscr{D}}_1[\![S_{(\xi)}[s_2]]\!]) = \mathcal{O}_1[\![S_{(\xi)}[s_2]]\!]$. From this and (*), it follows that $\mathscr{D}_1$ is not fully abstract w.r.t. $\mathcal{O}_1$.

Note that, when $\mathbf{V}$ is infinite, we cannot construct a statement yielding all infinite paths, such as X_0 in the above lemma; thus (†) in the above example does not hold when $\mathbf{V}$ is infinite. Moreover, for every $s \in \mathscr{L}_1$, it is shown that

$$\mathscr{D}_1[\![s]\!] = \Lambda(\mathscr{D}_1[\![s]\!]) = \hat{\mathscr{D}}_1[\![s]\!], \tag{18}$$

as follows: First, for every $q \in \mathscr{D}_1[\![s]\!]$, $\langle n, i \rangle \in \omega \times 2$, it does not hold that $q^{[n+i]} \cdot (\mathbf{E}_1 \langle \pi_i^2(q(n)) \rangle \cap (\Sigma \times \Sigma)^{\omega}) \subseteq p$, since $\mathscr{D}_1[\![s]\!]$ is image finite by Lemma 6(4). Hence, $q \in \mathscr{D}_1[\![s]\!]$ is not pruned away from $\mathscr{D}_1[\![s]\!]$ at place $\langle n; i \rangle$. Thus, one has (18).

3.6.2. *Comparison with a Less Abstract Model than $\mathscr{D}_1$ for $\mathscr{L}_1$*

In [BR91], another denotational model $\mathscr{D}'_1$ for a language, which is like $\mathscr{L}_1$ but has general sequential composition instead of prefixing, was proposed. The model $\mathscr{D}'_1$ was presented on the basis of the domain: $\mathbf{P}'_1 = \wp_{\mathrm{nc}}(\mathbf{Q}'_1)$, where $\mathbf{Q}'_1 \cong \{\varepsilon\} \cup (\Sigma \to (\Sigma \to \mathbf{Q}'_1))$. The outline of $\mathscr{D}'_1$ is as follows (the interpretation of the parallel composition is omitted, since this is not necessary for the present purpose):

(i) $\mathscr{D}'_1[\![x := e); s]\!] = \{(\lambda\sigma : \langle \sigma[[\![e]\!](\sigma)/x], q \rangle) : q \in \mathscr{D}'_1[\![s]\!]\}$.

(ii) The operation $\tilde{+}' : \mathbf{P}'_1 \times \mathbf{P}'_1 \to \mathbf{P}'_1$ is defined by $\{\varepsilon\} + p = p + \{\varepsilon\} = p$ and, for p_1, $p_2 \neq \{\varepsilon\}$, $p_1 + p_2$ is the set-theoretic union of p_1 and p_2.

(iii) $\mathscr{D}'_1[\![\mathrm{If}(b, s_1, s_2])] = \{(\lambda\sigma : \mathrm{if}([\![b]\!](\sigma) = \mathrm{tt}, q_1(\sigma), q_2(\sigma))); \; q_1 \in \mathscr{D}'_1[\![s_1]\!] \wedge q \in \mathscr{D}'_1[\![s_2]\!]\}$.

It turns out that $\mathscr{D}'_1$ is not fully abstract w.r.t. $\mathscr{D}_1$ as the next example shows. Thus, $\mathscr{D}'_1$ is less abstract than $\mathscr{D}_1$.

EXAMPLE 4. Let us assume, for simplicity, that $\mathrm{IVar} = \{x\}$. Then, Σ is identified with $\mathbf{V}$. Let $q_1 \equiv ((x := 0); \mathbf{0}) + ((x := 1); \mathbf{0})$, and

$$s_2 \equiv \mathrm{If}(x = 0, (x := 0); \mathbf{0}, (x := 1); \mathbf{0}) + \mathrm{If}(x = 0, (x := 1); \mathbf{0}, (x := 0); \mathbf{0}).$$

Then (*) $\mathscr{D}_1[\![s_1]\!] = \mathscr{D}_1[\![s_2]\!] = \{(\langle v, v' \rangle) : v \in \mathbf{V} \wedge (v' = 0 \vee v' = 1)\} \cdot \tilde{\mathbf{0}}_1$. On the other hand, $\mathscr{D}'_1[\![s_1]\!] = \{q_1, q_2\}$, where $q_1 = (\lambda v \in \mathbf{V} : \langle 0, \varepsilon \rangle)$, $q_2 = (\lambda v \in \mathbf{V} : \langle 1, \varepsilon \rangle)$. Also, $\mathscr{D}'_1[\![s_2]\!] = \{q'_1, q'_2\}$, where $q'_1 = (\lambda v \in \mathbf{V} : \mathrm{if}(v = 0, \langle 0, \varepsilon \rangle, \langle 1, \varepsilon \rangle))$, $q'_2 = (\lambda v \in \mathbf{V} : \mathrm{if}(v = 0, \langle 1, \varepsilon \rangle, \langle 0, \varepsilon \rangle))$. Hence (†) $\mathscr{D}'_1[\![s_1]\!] \neq \mathscr{D}'_1[\![s_2]\!]$. If $\mathscr{D}'_1$ is also fully abstract, then one has $\forall s_1, s_2 \in \mathscr{L}_1[\mathscr{D}_1[\![s_1]\!] =$

$\mathscr{D}_1[\![s_2]\!] \Leftrightarrow \mathscr{D}'_1[\![s_1]\!] = \mathscr{D}'_1[\![s_2]\!]]$, which contradicts (*) and (†). Hence $\mathscr{D}'_1$ cannot be fully abstract.

3.6.3. *Comparison with Hennessy and Plotkin's Resumptions Model*

The language treated in [HP79], which we denote by $\mathscr{L}_{co}$, is very similar to $\mathscr{L}_1$, except that it contains "co", a *coroutine* construct, as well as the usual interleaving. On the basis of a set $(a \in)$ Act of *primitive actions*, $(s \in)$ $\mathscr{L}_{co}$ is given by $s ::= a \mid (s_1, s_2) \mid \mathrm{If}(b, s_1, s_2) \mid \mathrm{While}(b, s) \mid (s_1 + s_2) \mid (s_1 \parallel s_2) \mid (s_1 \mathrm{\ co\ } s_2)$. A transition relation $\rightarrow \subseteq \mathscr{L}_{co} \times \mathrm{Str}$ with $\mathrm{Str} = \Sigma \cup (\mathscr{L}_{co} \times \Sigma)$ is defined, as $\rightarrow_1$, with the help of a given interpretation $\mathscr{A} : \mathrm{Act} \rightarrow (\Sigma \rightarrow \Sigma)$ (see Section 2 of [HP79]). The expression $\langle s, \sigma \rangle \rightarrow \sigma'$ means that the configuration $\langle s, \sigma \rangle$ terminates with state σ'. The operational semantics $\mathscr{B}$ treated in [HP79] is defined as follows: For every statement s and state σ, $\mathscr{B}[\![s]\!](\sigma) = \{\sigma' : \langle s, \sigma \rangle \rightarrow^* \sigma'\} \cup \mathrm{if}(\exists (\langle s_n, \sigma_n \rangle)_{n \in \omega} [\langle s_0, \sigma_0 \rangle = \langle s, \sigma \rangle \wedge \forall n \in \omega [\langle s_n, \sigma_n \rangle \rightarrow \langle s_{n+1}, \sigma_{n+1} \rangle]], \{\perp\}, \emptyset)$. Obviously $\mathscr{B}$ is more abstract than another operational semantics $\mathscr{O}_{co}: \mathscr{L}_{co} \rightarrow (\Sigma \rightarrow \wp_{nc}(\Sigma^{\leqslant \omega} \backslash \{\varepsilon\}))$ which is constructed by slightly modifying $\mathscr{O}_1$ in the obvious way. Then, a denotational model $\mathscr{V}$ for $\mathscr{L}_{co}$ is defined on the bases of a domain **R** which is the solution of a domain equation in the category of *non-deterministic domains*. Furthermore, the full abstraction of $\mathscr{V}$ w.r.t. $\mathscr{B}$ is established under the following three assumptions (see the paragraph immediately preceding Lemma 5.6 of [HP79]):

> (i) The set Σ of states is infinite. (ii) For each $\sigma \in \Sigma$, there exists a statement $K(\sigma) \in \mathrm{Act}$ such that $\forall \sigma' [\mathscr{A}[\![K(\sigma)]\!](\sigma') = \sigma]$. (iii) For each $\sigma \in \Sigma$, there exists an expression $\mathrm{is}(\sigma) \in \mathrm{BExp}$ such that $\forall \sigma' [[\![\mathrm{is}(\sigma)]\!](\sigma') = \mathrm{tt} \Leftrightarrow \sigma' = \sigma]$. (19)

We can construct a denotational model $\mathscr{D}_{co}$ for $\mathscr{L}_{co}$ by slighly modifying $\mathscr{D}_1$. First, the underlying domain $\mathbf{P}_{co}$ is defined by slightly modifying $\mathbf{P}_1$ as follows: $\mathbf{P}_{co} = \wp_{nc}(\mathbf{Q}_{co})$, where $\mathbf{Q}_{co}$ is the solution of domain equation: $\mathbf{Q}_{co} \cong (\Sigma \times \{\langle \surd, \sigma \rangle : \sigma \in \Sigma\}) \uplus (\Sigma \times \Sigma) \times \mathrm{id}_\kappa(\mathbf{Q}_{co})$ with '$\surd$' being some symbol standing for *termination*. Writing $\surd(\sigma)$ for $\langle \surd, \sigma \rangle$ for the sake of readability, one has $\mathbf{Q}_{co} \cong (\Sigma \times \Sigma)^{<\omega} \cdot \{(\langle \sigma, \surd(\sigma') \rangle) : \sigma, \sigma' \in \Sigma\} \uplus (\Sigma \times \Sigma)^\omega$, as with $\mathbf{Q}_1$. Then, the model $\mathscr{D}_{co} \rightarrow (\Sigma \rightarrow \mathbf{P}_{co})$ is defined by $\mathscr{D}_{co}[\![s]\!](\sigma) = \mathscr{D}^{t}_{co}[\![s]\!](\sigma) \cup \mathscr{D}^{n}_{co}[\![s]\!](\sigma)$, where $\mathscr{D}^{t}_{co}[\![s]\!](\sigma)$ and $\mathscr{D}^{n}_{co}[\![s]\!](\sigma)$ are the *terminating* and *nonterminating* parts of $\mathscr{D}_{co}[\![s]\!](\sigma)$; these parts are defined as follows: First, $\mathscr{D}^{t}_{co}[\![s]\!](\sigma) = \{(\langle \sigma_i, \sigma''_i \rangle)_{i \in n} \cdot (\langle \sigma_n, \surd(\sigma'_n) \rangle) : n \in \omega \wedge \sigma_0 = \sigma \wedge \exists (s_i)_{i \in (n+1)} [s_0 \equiv s \wedge \forall i \in n [\langle s_i, \sigma_i \rangle \rightarrow \langle s_{i+1}, \sigma'_i \rangle] \wedge \langle s_n, \sigma_n \rangle \rightarrow s'_n]\}$. Next, $\mathscr{D}^{n}_{co}[\![s]\!](\sigma) = \{(\langle \sigma_i, \sigma'_i \rangle)_{i \in \omega} : \sigma_0 = \sigma \wedge \exists (s_i)_{i \in \omega} [s_0 \equiv s \wedge \forall i \in \omega [\langle s_i, \sigma_i \rangle \rightarrow \langle s_{i+1}, \sigma'_i \rangle]]\}$. The model $\mathscr{D}_{co}$ can also be formulated by means of appropriate semantic operations and Banach's Theorem, as $\mathscr{D}_1$.

Interestingly, the full abstraction of $\mathscr{D}_{co}$ can also be established under the assumptions (19). Thus, the two models $\mathscr{V}$ and $\mathscr{D}_{co}$ are isomorphic in the sense of Lemma 11, while the two models are constructed rather differently. The proof of its full abstraction is outlined below.

Proof of Full Abstraction of $\mathscr{D}_{co}$. Let $s_1, s_2 \in \mathscr{L}_{co}$ such that $\mathscr{D}_{co}[\![s_1]\!] \neq \mathscr{D}_{co}[\![s_2]\!]$. Then, either $\mathscr{D}^{n}_{co}[\![s_1]\!] \neq \mathscr{D}^{n}_{co}[\![s_2]\!]$, or $\mathscr{D}^{t}_{co}[\![s_1]\!] \neq \mathscr{D}^{t}_{co}[\![s_2]\!]$. Let us set $p_i = \mathscr{D}_{co}[\![s_i]\!]$ $(i = 1, 2)$.

Case 1. Suppose $\mathscr{D}^{n}_{co}[\![s_1]\!] \neq \mathscr{D}^{n}_{co}[\![s_2]\!]$. Then, we can assume, without loss of generality, that there exists q such that $q \in \mathscr{D}^{n}_{co}[\![s_1]\!] \backslash \mathscr{D}^{n}_{co}[\![s_2]\!]$. Thus, by the closedness of p_2, there exists $(\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)}$ such that $(*)$ $(\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)} \in p_1^{[m+1]} \backslash p_2^{[m+1]}$ for some $m \in \omega$. As in [HP79], we can construct an appropriate tester T_0 for distinguishing s_1 and s_2 as follows: First, let $T_m \equiv \mathrm{If}(\mathrm{is}(\sigma'_m), K(\bar{\sigma}), K(\bar{\sigma}'))$, where $\bar{\sigma}$, $\bar{\sigma}'$ will be chosen below. Then T_i $(i \in m,)$ are defined by $T_i \equiv \mathrm{If}(\mathrm{is}(\sigma'_i), (K(\sigma_{i+1}); T_{i+1}), K(\bar{\sigma}'))$. We choose $\bar{\sigma}$ and $\bar{\sigma}'$ so that $(\dagger)$ $\bar{\sigma} \notin \bigcup_{k \in (m+1)} (\{\sigma : (\langle \sigma_i, \sigma'_i \rangle)_{i \in k} \cdot (\langle \sigma_k, \sigma \rangle) \in p_2^{[k+1]}\})$, $(\ddagger)$ $\bar{\sigma}' \neq \bar{\sigma}$. Note that the right-hand side of $(\dagger)$ is finite since the transition relation $\rightarrow$ is finitely branching, and thus, by the assumption (19)(i), we can choose such states. Then, obviously one has $(\langle \sigma_0, \sigma'_0 \rangle,$ $\langle \sigma'_0, \sigma_1 \rangle, ..., \langle \sigma_m, \sigma'_m \rangle, \langle \sigma_{m'}, \surd(\bar{\sigma}) \rangle) \in \mathscr{D}_{co}[\![s_1 \text{ co } T_0]\!]$, and therefore , $(**)$ $\bar{\sigma} \in \mathscr{B}[\![s_1 \text{ co } T_0]\!](\sigma_0)$. On the other hand, by the conditions $(\dagger)$ and $(\ddagger)$, one can show that $\bar{\sigma} \in \mathscr{B}[\![s_2 \text{ co } T_0]\!](\sigma_0) \Rightarrow (\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)} \in p_2^{[m+1]}$. Thus, since $(\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)} \notin p_2^{[m+1]}$, one has $(\dagger\dagger)$ $\bar{\sigma} \notin \mathscr{B}[\![s_2 \text{ co } T_0]\!](\sigma_0)$. By $(**)$ and $(\dagger\dagger)$, one has $\mathscr{B}[\![s_1 \text{ co } T_0]\!] \neq \mathscr{B}[\![s_2 \text{ co } T_0]\!]$.

Case 2. Suppose $\mathscr{D}^{t}_{co}[\![s_1]\!] \neq \mathscr{D}^{t}_{co}[\![s_2]\!]$. Then, we can assume, without loss of generality, that there exists $(\langle \sigma_i, \sigma'_i \rangle)_{i \in m} \cdot (\langle \sigma_m, \surd(\sigma'_m) \rangle) \in p_1 \backslash p_2$ $(m \in \omega)$. Let us choose $\bar{\sigma}$ so that $(\ddagger\ddagger)$ $\bar{\sigma} \notin \{\sigma'_i : i \in (m+1)\} \cup$ $\{\sigma : (\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)} \cdot (\langle \sigma'_m, \sigma \rangle) \in p_2^{[m+2]}\}$, and let $T \equiv (K(\bar{\sigma}))$; T' with T' being an arbitrary statement. Then obviously one has $(\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)} \cdot$ $(\langle \sigma'_m, \bar{\sigma} \rangle) \in (p_1; T)^{[m+2]}$. On the other hand, by the condition $(\ddagger\ddagger)$ it is impossible that $(\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)} \cdot (\langle \sigma'_m, \bar{\sigma} \rangle) \in (p_2; T)^{[m+2]}$. Hence, one has $(\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)} \cdot (\langle \sigma'_m, \bar{\sigma} \rangle) \in (p_1; T)^{[m+2]} \backslash (p_2; T)^{[m+2]}$. Thus, one obtains the same proposition as $(*)$ in Case 1, replacing $(\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)}$ by $(\langle \sigma_i, \sigma'_i \rangle)_{i \in (m+1)} \cdot (\langle \sigma'_m, \bar{\sigma} \rangle)$, and p_i by $(p_i; T)$ for $i = 1, 2$. Hence, one can construct T_0 such that $\mathscr{B}[\![(s_1; T) \text{ co } T_0]\!] \neq \mathscr{B}[\![(s_2; T) \text{ co } T_0]\!]$, as in Case 1. ▮

The full abstraction result for $\mathscr{V}$ and $\mathscr{D}_{co}$ essentially depends on the "co" construct; without this, the two models would not be fully abstract w.r.t. $\mathscr{B}$, which is also conjectured by Hennessy and Plotkin for $\mathscr{V}$ (see [HP79, Sect. 6]).

4. A Nonuniform Language with Communication

The second language $\mathscr{L}_2$ is a nonuniform language which has CSP-like *communications* in addition to all constructs of the first language. An operational model $\mathcal{O}_2$ for $\mathscr{L}_2$ is given as in Section 3.

The domain of a denotational model $\mathscr{D}_2$ for $\mathscr{L}_2$ is a kind of *failures model*, introduced in [BHR84], adapted to the nonuniform setting. Each element of the domain is a set consisting of such elements as $\langle(\langle\sigma_i, a_i, \sigma'_i\rangle)_i, \langle\sigma'', \Gamma\rangle\rangle$, where σ_i, σ'_i, and σ'' are states, a_i is an action, and Γ is a set of *communication sorts*. These elements are called *failures*; the parts $(\langle\sigma_i, a_i, \sigma'_i\rangle)_i$ and $\langle\sigma'', \Gamma\rangle$ are called a *trace* and a *refusal*, respectively.

First, the correctness of $\mathscr{D}_2$ is established as in Section 3. Then, the full abstraction of $\mathscr{D}_2$ is established by a combination of the testing method introduced in Section 3 and the method proposed by Bergstra *et al.* in [BKO88] to establish the full abstraction of a failures model for a uniform language without recursion. This method was adapted by Rutten in [Rut89] to employ it for a language with recursion in the framework of complete metric spaces, which suggests how to use it in the present setting.

The full abstraction of the denotational model for $\mathscr{L}_2$ is established as follows: Given two statements s_1 and s_2 of $\mathscr{L}_2$ which are distinct in their denotational meanings, the denotational meanings are distinct in the trace parts or in the refusal parts. When the distinction is in the trace parts, we can construct a tester by the testing method; otherwise we can construct a tester by the method of Bergstra *et al.*

4.1. *The Language* $\mathscr{L}_2$

In addition to all constructs of $\mathscr{L}_1$, the language $\mathscr{L}_2$ has CSP-like communications; i.e., it has *inputs* "$(c?x)$" and *outputs* "$(c!e)$" for all channels c, individual variables x, and value expressions e.

Definition 17 (Language $\mathscr{L}_2$). The set of statements of the nonuniform concurrent language $(S \in)\ \mathscr{L}_2^*$ is defined by the following BNF-syntax:

$$S ::= \mathbf{0} \mid (x := e); S \mid (c!e); S \mid (c?x); S \mid \text{If}(b, S_1, S_2) \mid S_1 + S_2 \mid S_1 \parallel S_2 \mid X \mid \xi.$$

Here X ranges over RVar, the set of recursion variables; ξ ranges over SVar, the set of place holders used for defining contexts as in Definition 4. In addition, c ranges over Chan, the set of *communication channels.* Let $(s \in)\ \mathscr{L}_2 = \{S \in \mathscr{L}_2^* : \text{FV}(S) = \varnothing\}$; for $\xi \in \text{SVar}$, let $\mathscr{L}_2^{\xi} = \{S \in \mathscr{L}_2^* : \text{FV}(S) \subseteq \{\xi\}\}$.

Then the set of *guarded statements* $(g \in) \mathscr{G}_2$ is defined by the following BNF-syntax:

$$g ::= \mathbf{0} \mid (x := e); s \mid (c!\,e); s \mid (c?\,x); s \mid \mathrm{If}(b, g_1, g_2) \mid g_1 + g_2 \mid g_1 \parallel g_2.$$

We assume that each recursion variable X is associated with an element g_X of $\mathscr{G}_2$ by a set of declarations $D = \{\langle X, g_X\rangle\}_{X \in \mathrm{RVar}}$.

In the sequel of this section, we fix a declaration set $D = \{\langle X, g_X\rangle\}_{X \in \mathrm{RVar}}$. As for $\mathscr{L}_1$, $\mathscr{L}_2^*$ and $\mathscr{L}_2$ can be formulated as the set of terms and the set of closed terms generated by a signature $\mathscr{S}_2$, respectively.

4.2. *Operational Model $\mathcal{O}_2$ for $\mathscr{L}_2$*

An operational model $\mathcal{O}_2$ for $\mathscr{L}_2$ is defined in terms of a transition relation $\rightarrow_2$. The following definition is given as a preliminary to the definition of $\rightarrow_2$.

DEFINITION 18 (Actions). (1) The set of *communication sorts*, $(\gamma \in) \mathbf{C}$, is given by $\mathbf{C} = \{c! : c \in \mathrm{Chan}\} \cup \{c? : c \in \mathrm{Chan}\}$.

(2) The set of *actions*, $(a \in) \mathbf{A}$, is given by $\mathbf{A} = (\mathbf{C} \times \mathbf{V}) \cup \{\tau\}$.

(3) The set of *action sorts*, $(A \in)$ ASort, is given by $\mathrm{Asort} = \mathbf{C} \cup \{\tau\}$.

(4) A function $\mathrm{sort}: \mathbf{A} \rightarrow \mathrm{ASort}$ is defined as follows: For $a \in \mathbf{A}$, $\mathrm{sort}(a) = \gamma$ if $a = \langle \gamma, v\rangle \in \mathbf{C} \times \mathbf{V}$; otherwise $\mathrm{sort}(a) = \tau$. ∎

The transition relation $\rightarrow_2 \subseteq (\mathscr{L}_2 \times \Sigma) \times \mathbf{A} \times (\mathscr{L}_2 \times \Sigma)$ is defined as follows. For $s_1, s_2 \in \mathscr{L}_2$, $\sigma_1, \sigma_2 \in \Sigma$, and $a \in \mathbf{A}$, we write $\langle s_1, \sigma_1\rangle \xrightarrow{a}_2 \langle s_2, \sigma_2\rangle$ for $\langle\langle s_1, \sigma_1\rangle, a, \langle s_2, \sigma_2\rangle\rangle \in \rightarrow_2$. For $c!$, $c? \in \mathbf{C}$ and $v \in \mathbf{V}$, we sometimes write $c!\,v$ and $c?\,v$ for $\langle c!, v\rangle$ and $\langle c?, v\rangle$, respectively.

DEFINITION 19 (Transition Relation $\rightarrow_2$). The transition relation $\rightarrow_2$ is defined as the smallest relation satisfying the following rules (1) to (9):

(1) $\langle (x := e); s, \sigma\rangle \xrightarrow{\tau}_2 \langle s, \sigma[[\![e]\!](\sigma)/x]\rangle$

(2) $\langle (c!\,e); s, \sigma\rangle \xrightarrow{\langle c!, [\![e]\!](\sigma)\rangle}_2 \langle s, \sigma\rangle$

(3) $\langle (c?\,x); s, \sigma\rangle \xrightarrow{c?v}_2 \langle s, \sigma[v/x]\rangle \; (v \in \mathbf{V})$

(4) $$\frac{\langle s_1, \sigma\rangle \xrightarrow{a}_2 \langle s, \sigma'\rangle}{\langle \mathrm{If}(b, s_1, s_2), \sigma\rangle \xrightarrow{a}_2 \langle s, \sigma'\rangle} \; ([\![b]\!](\sigma) = \mathrm{tt})$$

(5) $$\frac{\langle s_2, \sigma\rangle \xrightarrow{a}_2 \langle s, \sigma'\rangle}{\langle \mathrm{If}(b, s_1, s_2), \sigma\rangle \xrightarrow{a}_2 \langle s, \sigma'\rangle} \quad ([\![b]\!](\sigma) = \mathrm{ff})$$

(6) $$\frac{\langle s_1, \sigma\rangle \xrightarrow{a}_2 \langle s, \sigma'\rangle}{\begin{array}{l}\langle s_1 + s_2, \sigma\rangle \xrightarrow{a}_2 \langle s, \sigma'\rangle\\ \langle s_2 + s_1, \sigma\rangle \xrightarrow{a}_2 \langle s, \sigma'\rangle\end{array}} \quad (a \in \mathbf{A})$$

(7) $$\frac{\langle s_1, \sigma\rangle \xrightarrow{a}_2 \langle s, \sigma'\rangle}{\begin{array}{l}\langle s_2 \parallel s_1, \sigma\rangle \xrightarrow{a}_2 \langle s \parallel s_2, \sigma'\rangle\\ \langle s_2 \parallel s_1, \sigma\rangle \xrightarrow{a}_2 \langle s_2 \parallel s, \sigma'\rangle\end{array}} \quad (a \in \mathbf{A})$$

(8) $$\frac{\langle s_1, \sigma\rangle \xrightarrow{c!v}_2 \langle s_1', \sigma\rangle,\ \langle s_2, \sigma\rangle \xrightarrow{c?v}_2 \langle s_2', \sigma'\rangle}{\begin{array}{l}\langle s_1 \parallel s_2, \sigma\rangle \xrightarrow{\tau}_2 \langle s_1' \parallel s_2', \sigma'\rangle\\ \langle s_2 \parallel s_1, \sigma\rangle \xrightarrow{\tau}_2 \langle s_2' \parallel s_1', \sigma'\rangle\end{array}} \quad (c \in \mathrm{Chan}, v \in \mathbf{V})$$

(9) $$\frac{\langle g_X, \sigma\rangle \xrightarrow{a}_2 \langle s', \sigma'\rangle}{\langle X, \sigma\rangle \xrightarrow{a}_2 \langle s', \sigma'\rangle} \quad (\langle X, g_X\rangle \in D)$$

For $\langle s, \sigma\rangle \in \mathscr{L}_2 \times \Sigma$, let $\mathrm{act}(s, \sigma) = \{a \in \mathbf{A} : \exists \langle s', \sigma'\rangle \in \mathscr{L}_2 \times \Sigma[\langle s, \sigma\rangle \xrightarrow{a}_2 \langle s', \sigma'\rangle]\}$. Moreover, let $\mathrm{sact}(s, \sigma) = \mathrm{sort}[\mathrm{act}(s, \sigma)]$.

The transition relation is *image finite* in the sense of part (1) of the following lemma:

LEMMA 16. *For every* $s \in \mathscr{L}_2$, $\sigma \in \Sigma$, *the following hold*:

(1) *For every* $a \in \mathbf{A}$, *the set* $\{\langle s', \sigma'\rangle \in \mathscr{L}_2 \times \Sigma : \langle s, \sigma\rangle \xrightarrow{a}_2 \langle s', \sigma'\rangle\}$ *is finite.*

(2) $\mathrm{asort}(s, \sigma)$ *is finite.*

(3) *For every* $c \in \mathrm{Chan}$, *the set* $\{v \in \mathbf{V} : \langle c!, v\rangle \in \mathrm{act}(s, \sigma)\}$ *is finite.*

Proof. These are shown in a fashion similar to the proof of Lemma 3. ∎

In terms of the transition relation $\rightarrow_2$, the operational model $\mathcal{O}_2$ is defined as follows:

DEFINITION 20 (Operational Model $\mathcal{O}_2$ for $\mathscr{L}_2$). (1) Let $\mathbf{M}_2^{\mathcal{O}} = (\mathscr{L}_2 \rightarrow (\Sigma \rightarrow \wp_{\mathrm{nc}}((\mathbf{A} \times \Sigma)^{\leqslant \omega})))$, and let $\Psi_2^{\mathcal{O}} : \mathbf{M}_2^{\mathcal{O}} \rightarrow \mathbf{M}_2^{\mathcal{O}}$ be defined as follows: For $f \in \mathbf{M}_2^{\mathcal{O}}$, $s \in \mathscr{L}_2$, and $\sigma \in \Sigma$,

$$\Psi_2^{\mathcal{O}}(f)(s)(\sigma) = \bigcup \{\langle a, \sigma'\rangle \cdot f(s')(\sigma') : \langle s, \sigma\rangle \xrightarrow{a}_2 \langle s', \sigma'\rangle\}$$
$$\cup\, \mathrm{If}(\tau \notin \mathrm{act}(s, \sigma), \{\varepsilon\}, \varnothing).$$

It follows that $\Psi_2^{\mathcal{O}}$ is a contraction from $\mathbf{M}_2^{\mathcal{O}}$ to $\mathbf{M}_2^{\mathcal{O}}$, as in Definition 6.

(2) Let the operational model $\mathcal{O}_2$ be the unique fixed point of $\Psi_2^{\mathcal{O}}$. By the definition, one has $\mathcal{O}_2: \mathcal{L}_2 \to (\Sigma \to \wp_{\mathrm{nc}}((\mathbf{A} \times \Sigma)^{\leqslant \omega}))$, and for each $s \in \mathcal{L}_2$ and $\sigma \in \Sigma$,

$$\mathcal{O}_2[\![s]\!](\sigma) = \bigcup \{\langle a, \sigma' \rangle \cdot \mathcal{O}_2[\![s']\!](\sigma') : \langle s, \sigma \rangle \xrightarrow{a}_2 \langle s', \sigma' \rangle\}$$
$$\cup \, \mathrm{If}(\tau \notin \mathrm{act}(s, \sigma), \{\varepsilon\}, \varnothing).$$

4.3. *Denotational Model $\mathcal{D}_2$ for $\mathcal{L}_2$*

The domain of a denotational semantic domain $\mathbf{P}_2$ for $\mathcal{L}_2$ is a kind of *failures model*, which was introduced in BHR84], adapted to the non-uniform setting. Each element of the domain is a set consisting of such elements as $\langle (\langle \sigma_i, a_i, \sigma_i' \rangle)_i, \langle \sigma'', \Gamma \rangle \rangle$, where σ_i, σ_i', and σ'' are states, a_i is an action, and Γ is a set of *communication sorts*. These elements are called *failures*. Formally $\mathbf{P}_2$ is defined by:

DEFINITION 21 (Denotational Semantic Domain $\mathbf{P}_2$ for $\mathcal{L}_2$). (1) Let $\mathbf{Q}_2$ be the unique solution of $\mathbf{Q}_2 \cong (\Sigma \times \wp(\mathbf{C})) \uplus ((\Sigma \times \mathbf{A} \times \Sigma) \times \mathrm{id}_\kappa(\mathbf{Q}_2))$. One has $\mathbf{Q}_2 \cong ((\Sigma \times \mathbf{A} \times \Sigma)^{<\omega} \cdot (\Sigma \times \wp(\mathbf{C}))) \cup (\Sigma \times \mathbf{A} \times \Sigma)^{\omega}$.

(2) For $p \in \wp_{\mathrm{nc}}(\mathbf{Q}_2)$ and $r \in (\Sigma \times \mathbf{A} \times \Sigma)^{<\omega}$, the *remainder* of p with prefix r, denoted by $p[r]$, is defined by $p[r] = \{q' \in \mathbf{Q}_2 : r \cdot q' \in p\}$.

(3) For $q \in \mathbf{Q}_2 \cup (\Sigma \times \mathbf{A} \times \Sigma)^+$, let $\mathrm{istate}_2(q) = \sigma$ if $q = (\langle \sigma, a, \sigma' \rangle) \cdot q'$, and let $\mathrm{istate}_2(q) = \sigma''$ if $\exists \Gamma [q = (\langle \sigma'', \Gamma \rangle)]$.

(4) For $p \in \wp_{\mathrm{nc}}(\mathbf{Q}_2)$ and $\sigma \in \Sigma$, let $p\langle \sigma \rangle = \{q \in p : \mathrm{istate}_2(q) = \sigma\}$.

(5) The process $p \in \wp_{\mathrm{nc}}(\mathbf{Q}_2)$ is *uniformly nonempty at level* iff

$$\forall r \in (\Sigma \times \mathbf{A} \times \Sigma)^n \, [p[r] \neq \varnothing \Rightarrow \forall \sigma \in \Sigma [p[r]\langle \sigma \rangle \neq \varnothing]].$$

Moreover, p is *uniformly nonempty* iff p is uniformly nonempty at every level $n \in \omega$.

(6) Let $\mathbf{P}_2$, the domain of processes for $\mathcal{L}_2$, be given by

$$\mathbf{P}_2 = \{p \in \wp(\mathbf{Q}_2) : p \text{ is uniformly nonempty}\}.$$

(7) For $\gamma \in \mathbf{C}$, let $\bar{\gamma} = c?$ if $\gamma = c!$; otherwise $\gamma = c?$ and $\bar{\gamma} = c!$. Moreover, for $\Gamma \in \wp(\mathbf{C})$, let $\bar{\Gamma} = \{\bar{\gamma} : \gamma \in \Gamma\}$.

We have the following lemma for $\mathbf{P}_2$, which is similar to Lemma 4 for $\mathbf{P}_1$.

LEMMA 17. *The set $\mathbf{P}_2$ is closed in $\wp_{\mathrm{nc}}(\mathbf{Q}_2)$, and therefore, $\mathbf{P}_2$ is a complete metric space with the original metric of $\wp_{\mathrm{nc}}(\mathbf{Q}_2)$.*

Proof. This is proved in a similar fashion to the proof of Lemma 4. ▮

The interpretation $\mathcal{I}_2$ for the signature of $\mathcal{L}_2$ is defined as follows:

DEFINITION 22 (Interpretation $\mathcal{I}_2$ for Signature of $\mathcal{L}_2$). (1) $\tilde{\mathbf{0}}_2 = \{(\langle\sigma, \Gamma\rangle) : \langle\sigma, \Gamma\rangle \in \Sigma \times \wp(\mathbf{C})\}$.

(2) For $x \in \text{IVar}$ and $e \in \text{VExp}$, $\text{asg}_2(x, e) : \mathbf{P}_2 \to \mathbf{P}_2$ is defined as follows: For $p \in \mathbf{P}_2$,

$$\text{asg}_2(x, e)(p) = \{(\langle\sigma, \tau, \sigma[\llbracket e\rrbracket(\sigma)/x]\rangle) \cdot p : \sigma \in \Sigma\}.$$

(3) For $c \in \text{Chan}$ and $e \in \text{VExp}$, $\text{out}(c, e) : \mathbf{P}_2 \to \mathbf{P}_2$ is defined as follows: For $p \in \mathbf{P}_2$,

$$\text{out}(c, e)(p) = \{(\langle\sigma, \langle c!, \llbracket e\rrbracket(\sigma)\rangle, \sigma\rangle) \cdot p : \sigma \in \Sigma\} \cup \{(\langle\sigma, \Gamma\rangle) : \sigma \in \Sigma \wedge \Gamma \subseteq \mathbf{C} \backslash \{c!\}\}.$$

(4) For $c \in \text{Chan}$ and $x \in \text{IVar}$, $\text{inp}(c, x) : \mathbf{P}_2 \to \mathbf{P}_2$ is defined as follows: For $p \in \mathbf{P}_2$,

$$\text{inp}(c, x)(p) = \{(\langle\sigma, c?v, \sigma[v/x]\rangle) \cdot p : \sigma \in \Sigma \wedge v \in \mathbf{V}\} \cup \{(\langle\sigma, \Gamma\rangle) : \sigma \in \Sigma \wedge \Gamma \subseteq \mathbf{C} \backslash \{c?\}\}.$$

(5) For $b \in \text{BExp}$, $\text{if}(b) : \mathbf{P}_2 \times \mathbf{P}_2 \to \mathbf{P}_2$ is defined as follows: For $p_1, p_2 \in \mathbf{P}_2$,

$$\text{if}(b)(p_1, p_2) = \bigcup_{\sigma \in \Sigma} [\text{if}(\llbracket b\rrbracket(\sigma) = \text{tt}, p_1\langle\sigma\rangle, p_2\langle\sigma\rangle)].$$

(6) For $p \in \mathbf{P}_2$, $p \cap ((\Sigma \times \mathbf{A} \times \Sigma) \times \mathbf{Q}_2)$ is called the *action part* of p and denoted by p^+.

For $p_1, p_2 \in \mathbf{P}_2$, $p_1 \tilde{+} p_2$ is defined as in Definition 8 by

$$p_1 \tilde{+} p_2 = p_1^+ \cup p_2^+ \cup \{(\langle\sigma, \Gamma\rangle) \in \Sigma \times \wp(\mathbf{C}) : (\langle\sigma, \Gamma\rangle) \in p_1 \cap p_2\}.$$

A process $p \in \mathbf{P}_2$ is said to be *downward closed* at level 0 if

$$\forall\sigma, \forall\Gamma[(\langle\sigma, \Gamma\rangle) \in p \Rightarrow \forall\Gamma'[\Gamma' \subseteq \Gamma \Rightarrow (\langle\sigma, \Gamma'\rangle) \in p]].$$

It follows immediately from the definition of $\tilde{+}$ that if p_1 and p_2 are downward closed, then

$$p_1 \tilde{+} p_2 = p_1^+ \cup p_2^+ \cup \{(\langle\sigma, \Gamma\rangle) \in \Sigma \times \wp(\mathbf{C}) : \exists(\langle\sigma, \Gamma_1\rangle) \in p_1; \exists(\langle\sigma, \Gamma_2\rangle) \in p_2[\Gamma \subseteq \Gamma_1 \cap \Gamma_2]\}.$$

(7) We have the unique operation $\tilde{\|}: \mathbf{P}_2 \times \mathbf{P}_2 \to \mathbf{P}_2$ satisfying the following equation; the existence and uniqueness of such an operation are obtained as in Definition 8(5). For $p_1, p_2 \in \mathbf{P}_2$,

$$p_1 \tilde{\|} p_2 = (p_1 \mathbin{\|\!\!\_} p_2) \cup (p_2 \mathbin{\|\!\!\_} p_1) \cup (p_1 \rhd p_2) \cup (p_2 \rhd p_1) \cup (p_1 \# p_2),$$

where

$$p_1 \mathbin{\|\!\!\_} p_2 = \bigcup \{\langle \sigma, a, \sigma' \rangle \cdot (p_1[\langle \sigma, a, \sigma' \rangle] \tilde{\|} p_2) : p_1[\langle \sigma, a, \sigma' \rangle] \neq \varnothing\},$$

$$p_1 \rhd p_2 = \left(\bigcup \{(\langle \sigma, \tau, \sigma' \rangle) \cdot (p_1[\langle \sigma, c!\, v, \sigma \rangle] \tilde{\|} p_2[\langle \sigma, c?\, v, \sigma' \rangle]) : \right.$$
$$\left. p_1[\langle \sigma, c!\, v, \sigma \rangle] \neq \varnothing \wedge p_2[\langle \sigma, c?\, v, \sigma' \rangle] \neq \varnothing\} \right)^{\mathrm{cls}},$$

$$p_1 \# p_2 = \{(\langle \sigma, \Gamma \rangle) : \exists (\langle \sigma, \Gamma_1 \rangle) \in p_1,$$
$$\exists (\langle \sigma, \Gamma_2 \rangle) \in p_2 [(\mathbf{C} \backslash \Gamma_1) \cap \overline{(\mathbf{C} \backslash \Gamma_2)} = \varnothing \wedge \Gamma \subseteq \Gamma_1 \cap \Gamma_2]\}.$$

Note that taking closure in the right-hand side of (20) is necessary, as Example 5 shows below.

(8) $$\mathscr{I}_2 = \{\tilde{\mathbf{0}}_2, \{\mathrm{asg}_2(x, e) : \langle x, e \rangle \in \mathrm{IVar} \times \mathrm{VExp}\},$$
$$\{\mathrm{if}(b) : b \in \mathrm{BExp}\}, \tilde{+}, \tilde{\|},$$
$$\{\mathrm{out}(c, e) : c \in \mathrm{Chan} \wedge e \in \mathrm{VExp}\},$$
$$\{\mathrm{inp}(c, x) : c \in \mathrm{Chan} \wedge x \in \mathrm{IVar}\}\}.$$

EXAMPLE 5. Let us assume, for simplicity, that $\mathrm{IVar} = \{x\}$ and $\mathbf{V} = \{v\}$. Then the set of states consists only of one state denoted by v. Moreover assume that $\mathrm{Chan} = \{c_i : i \in \omega\}$ and $c_i \neq c_j$ for $i \neq j$. Let p_1 and p_2 be defined by $p_1 = \{q_n : n \in \omega\}$, $p_2 = \{(\langle v, c_n?\, v, v \rangle, \langle v, \varnothing \rangle) : n \in \omega\}$, where $q_n = \langle v, c_n!\, v, v \rangle \cdot \underbrace{\langle v, c_0!\, v, v \rangle \cdots \langle v, c_0!\, v, v \rangle}_{n} \cdot \langle v, \varnothing \rangle$. Then p_1 and p_2 belong to $\mathbf{P}_2$, and moreover they are image finite, which notion is to be defined in Definition 24. Nevertheless, it is shown that the right-hand side of (20) without taking closure is not closed as follows. This set is $\{q'_n : n \in \omega\}$, where $q'_n = \langle v, \tau, v \rangle \cdot \underbrace{\langle v, c_0!\, v, v \rangle \cdots \langle v, c_0!\, v, v \rangle}_{n} \cdot \langle v, \varnothing \rangle$. This is not closed, since the infinite sequence $(\langle v, \tau, v \rangle, \langle v, c_0!\, v, v \rangle, \langle v, c_0!\, v, v \rangle, \ldots)$ is a member of its closure but is not a member of it.

The next lemma follows immediately from Definition 22(7).

LEMMA 18. $\forall p_1, p_2 \in \mathbf{P}_2[p_1 \tilde{\|} p_2 = p_2 \tilde{\|} p_1]$.

In terms of the interpretation $\mathscr{I}_2$, the denotational model $\mathscr{D}_2$ is defined by induction on the structure of $s \in \mathscr{L}_2$, as in Definition 9.

DEFINITION 23 (Denotational Model $\mathscr{D}_2$ for $\mathscr{L}_2$). First, a contraction Π_2 from $\mathbf{M}_2^{\mathscr{D}} = (\mathrm{RVar} \to \mathbf{P}_2)$ to itself is defined as in Definition 9(1), using $\mathscr{I}_2$ instead of $\mathscr{I}_1$. Let $\mathbf{p}_0 = \mathrm{fix}(\Pi_2)$, and for $X \in \mathrm{RVar}$, let us define $X^{\mathscr{D}_2}$, the denotational meaning for X, by: $X^{\mathscr{D}_2} = \mathbf{p}_0(X)$. Next, for each operator F of $\mathscr{L}_2$ with arity r, and $s_1, ..., s_r \in \mathscr{L}_2$, let $\mathscr{D}_2[\![F(s_1, ..., s_2)]\!] = F^{\mathscr{I}_2}(\mathscr{D}_2[\![s_1]\!], ..., \mathscr{D}_2[\![s_r]\!])$, where $F^{\mathscr{I}_2}$ is the interpreted operation corresponding to F.

Several properties including the so-called *image finiteness* for elements of $\mathbf{P}_2$ are introduced. It will be shown that the denotational meaning of each statement in $\mathscr{L}_2$ has these properties; this fact is used for establishing the full abstraction of $\mathscr{D}_2$.

DEFINITION 24 (Image Finiteness for Elements of $\mathbf{P}_2$). Let $p \in \mathbf{P}_2$ and $n \in \omega$.

(1) The process p is *image finite at level n*, written $\mathrm{IFin}_2^{(n)}(p)$, iff

$$\forall r \in (\Sigma \times \mathbf{A} \times \Sigma)^{<\omega}\ [p[r] \neq \varnothing \Rightarrow \forall \sigma \in \Sigma, \forall a \in \mathbf{A}[\{\sigma' \in \Sigma : p[r][\langle \sigma, a, \sigma' \rangle] \neq \varnothing\} \text{ is finite}]].$$

The process p is *image finite*, written $\mathrm{IFin}_2(p)$, iff $\forall n \in \omega[\mathrm{IFin}_2^{(n)}(p)]$.

(2) The process p is *finite w.r.t. action sorts at level n*, written $\mathrm{ASFin}^{(n)}(p)$, iff

$$\forall r \in (\Sigma \times \mathbf{A} \times \Sigma)^{<\omega}\ [p[r] \neq \varnothing \Rightarrow \forall \sigma \in \Sigma[\mathrm{sact}(p[r], \sigma) \text{ is finite}]].$$

The process p is *finite w.r.t. action sorts*, written $\mathrm{ASFin}(p)$, iff $\forall n \in \omega[\mathrm{ASFin}^{(n)}(p)]$.

(3) The process p is *finite w.r.t. output values at level n*, written $\mathrm{OVFin}^{(n)}(p)$, iff

$$\forall r \in (\Sigma \times \mathbf{A} \times \Sigma)^{<\omega}\ [p[r] \neq \varnothing \Rightarrow \forall \sigma \in \Sigma, \forall c \in \mathrm{Chan}[\{v \in \mathbf{V} : \exists \sigma'[p[r][\langle \sigma, c!\,v, \sigma' \rangle] \neq \varnothing]\} \text{ is finite}]].$$

The process p is *finite w.r.t. output values*, written $\mathrm{OVFin}(p)$, iff $\forall n \in \omega[\mathrm{OVFin}^{(n)}(p)]$.

(4) The process p satisfies the *disjointness inaction condition at level* n, written $\mathrm{DIC}^{(n)}(p)$, iff

$$\forall r \in (\Sigma \times \mathbf{A} \times \Sigma)^n \; [p[r] \neq \emptyset$$
$$\Rightarrow \forall \sigma \in \Sigma, \exists \mathscr{R} \subseteq \wp(\mathrm{sact}(p[r], \sigma) \cap \mathbf{C})[\forall \Gamma \in \wp(\mathbf{C})[(\langle \sigma, \Gamma \rangle) \in p[r]$$
$$\Leftrightarrow \exists R \in \mathscr{R}[\Gamma \cap R = \emptyset]]]].$$

The process p satisfies the *disjointness inaction condition*, written $\mathrm{DIC}(p)$, iff $\forall n \in \omega[\mathrm{DIC}^{(n)}(p)]$. (See Example 5, for a motivation of this definition.)

(5) Properties $\mathrm{FIRN}_2^{(n)}(p)$, $\mathrm{FIRT}_2^{(n)}(p)$, and $\mathrm{FIR}_2(p)$ are defined as $\mathrm{FIRN}_1^{(n)}(p)$, $\mathrm{FIRT}_1^{(n)}(p)$, and $\mathrm{FIR}_1(p)$ in Definition 10(2). Formally, these are defined as follows:

(i) First, $\mathrm{FIRN}_2^{(n)}(p)$ iff there exists $\mathscr{X} \in \wp_f(\mathrm{IVar})$ such that the following holds: $\forall r \in (\Sigma \times \mathbf{A} \times \Sigma)^n$, $\forall \vec{\sigma} \in ((\mathrm{IVar} \backslash \mathscr{X}) \to \mathbf{V})^n \; [r \in p^{[n]} \Leftrightarrow \forall i \in n[\pi_0^3(r(i)) \upharpoonright (\mathrm{IVar} \backslash \mathscr{X}) = \pi_2^3(r(i)) \upharpoonright (\mathrm{IVar} \backslash \mathscr{X})] \wedge (\langle (\pi_0^3(r(i)) \upharpoonright \mathscr{X}) \cup \vec{\sigma}(i), \pi_1^3(r(i)), \pi_2^3(r(i)) \upharpoonright \mathscr{X}) \cup \vec{\sigma}(i) \rangle)_{i \in n} \in p^{[n]}]$. That is, for each $r \in (\Sigma \times \mathbf{A} \times \Sigma)^n$, if $r \in p^{[n]}$, then, in every step $r(i) = \langle \pi_0^3(r(i)), \pi_1^3(r(i)), \pi_1^3(r(i)) \rangle$ of r $(i \in n)$, the value for $x \in \mathrm{IVar} \backslash \mathscr{X}$ is not changed, i.e., $(*)$ $\pi_0^3(r(i)) \upharpoonright (\mathrm{IVar}) \backslash \mathscr{X}) = \pi_2^3(r(i)) \upharpoonright (\mathrm{IVar} \backslash \mathscr{X})$, and one may change the value $\pi_j^3(r(i))(x)$ $(j = 0, 2)$ arbitrarily, i.e., $(\dagger)$ $(\langle (\pi_0^3(r(i)) \upharpoonright \mathscr{X}) \cup \vec{\sigma}(i), \pi_1^3(r(1)), (\pi_0^3(r(i)) \upharpoonright \mathscr{X}) \cup \vec{\sigma}(i) \rangle)_{i \in n} \in p^{[n]}$ for arbitrary $\vec{\sigma} \in (\mathrm{IVar} \backslash \mathscr{X}) \to \mathbf{V})^n$. Conversely, for arbitrary $\vec{\sigma} \in (\mathrm{IVar} \backslash \mathscr{X}) \to \mathbf{V})^n$, if one has $(*)$ and $(\dagger)$, then $r \in p^{[n]}$.

(ii) Similarly, $\mathrm{FIRT}_2^{(n)}(p) \Leftrightarrow \exists \mathscr{X} \in \wp_f(\mathrm{IVar})$, $\forall q \in (\Sigma \times \mathbf{A} \times \Sigma)^n \cdot (\Sigma \times \wp(\mathbf{C}))$, $\forall \vec{\sigma} \in (\mathrm{IVar} \backslash \mathscr{X}) \to \mathbf{V})^{n+1} \; [q \in p \Leftrightarrow \forall i \in n[\pi_0^3(q(i)) \upharpoonright (\mathrm{IVar} \backslash \mathscr{X}) = \pi_2^3(q(i)) \upharpoonright (\mathrm{IVar} \backslash \mathscr{X})] \wedge (\langle (\pi_0^3(q(i)) \upharpoonright \mathscr{X}) \cup \vec{\sigma}(i), \pi_1^3(q(i)), (\pi_2^3(q(i)) \upharpoonright \mathscr{X}) \cup \vec{\sigma}(i) \rangle)_{i \in n} \cdot (\langle (\pi_0^2(q(n)) \upharpoonright \mathscr{X}) \cup \vec{\sigma}(n), \pi_1^2(q(n)) \rangle) \in p]$.

(iii) $\mathrm{FIR}_2(p) \Leftrightarrow \forall n \in \omega[\mathrm{FIRN}_2^{(n)}(p) \wedge \mathrm{FIRT}_2^{(n)}(p)]$.

(6) $\mathbf{P}_2^* = \{p \in \mathbf{P}_2 : \mathrm{IFin}_2(p) \wedge \mathrm{ASFin}(p) \wedge \mathrm{OVFin}(p) \wedge \mathrm{DIC}(p) \wedge \mathrm{FIR}_2(p)\}$.

Remark 5. Though the condition $\mathrm{DIC}^{(0)}(\cdot)$ might seem too complicated, it is characterized in terms of a simpler condition $D(\cdot)$ defined as follows: For $p \in \mathbf{P}_2$, $(*)$ $D(p) \Leftrightarrow \forall \sigma[\exists \Gamma[\langle \sigma, \Gamma \rangle \in p] \Rightarrow \exists R \subseteq \mathrm{sact}(s, \sigma) \cap \mathbf{C}, \forall \Gamma[\langle \sigma, \Gamma \rangle \in p \Leftrightarrow \Gamma \cap R = \emptyset]]$. Let $\mathbf{P}'$ be the smallest subset of $\mathbf{P}_2$ which includes $\{p \in \mathbf{P}_2 : D(p)\}$ and is closed under set-theoretical union; i.e., let $\mathbf{P}' = \{\bigcup \mathbf{P}'' : \mathbf{P}'' \subseteq \mathbf{P}_2 \wedge \bigcup \mathbf{P}'' \in \mathbf{P}_2 \wedge \forall p' \in \mathbf{P}''[D(p')]\}$. Then one has $\mathbf{P}' = \{p \in \mathbf{P}_2 : \mathrm{DIC}^{(0)}(p)\}$. The part $\mathbf{P}' \supseteq \{p \in \mathbf{P}_2 : \mathrm{DIC}^{(0)}(p)\}$ is shown as follows (the other part is shown more straightforwardly). Let $p \in \mathbf{P}_2$ with $\mathrm{DIC}^{(0)}(p)$, and $\Sigma' = \{\sigma : \exists \Gamma[\langle \sigma, \Gamma \rangle \in p]\}$. Then for each $\sigma \in \Sigma'$, there exists $\mathscr{R}_\sigma$ such that $\forall \Gamma[\langle \sigma, \Gamma \rangle \in p \Leftrightarrow \exists R \in \mathscr{R}_\sigma[\Gamma \cap R = \emptyset]]$. Fix such $\mathscr{R}_\sigma$, and for each $\vec{R} \in \prod_{\sigma \in \Sigma'}(\mathscr{R}_\sigma)$, put $p(\vec{R}) = \{q \in p : \mathrm{lgt}(q) \geqslant 2\} \cup \{\langle \sigma, \Gamma \rangle : \sigma \in \Sigma' \wedge \Gamma \cap \vec{R}(\sigma) = \emptyset\}$. Then, one has $D(p(\vec{R}))$ and $p = \bigcup \{p(\vec{R}) : \vec{R} \in \prod_{\sigma \in \Sigma'}(\mathscr{R}_\sigma)\}$, and therefore, $p \in \mathbf{P}'$.

Also, as is obvious from Remark 1, the set $\{p \in \mathbf{P}_2 : \mathrm{DIC}(p)\}$ is defined as the largest subset of $\mathbf{P}_2$ which is included in $\{p \in \mathbf{P}_2 : \mathrm{DIC}^{(0)}(p)\}$ and *closed under taking remainders*, where closedness under taking remainders for subsets of $\mathbf{P}_2$ is defined as in Remark 1. It is easy to check that the downward closedness of $p \in \mathbf{P}_2$ follows from that fact that $\mathrm{DIC}(p)$.

It turns out that the denotational meaning of each statement is a member of $\mathbf{P}_1^*$, which is used for establishing the full abstraction of $\mathscr{D}_2$.

LEMMA 19 (1) *The set* $\mathbf{P}_2^*$ *is closed in* $\mathbf{P}_2$.

(2) $\forall p \in \mathbf{P}_2^*, \forall r \in (\Sigma \times \mathbf{A} \times \Sigma)^{<\omega}\, [p[r] \neq \varnothing \Rightarrow p[r] \in \mathbf{P}_2^*]$.

(3) *The set* $\mathbf{P}_2^*$ *is closed under all interpreted operations of* $\mathscr{L}_2$.

(4) $\mathscr{D}_2[\mathscr{L}_2] \subseteq \mathbf{P}_2^*$.

(5) $\forall p \in \mathscr{D}_2[\mathscr{L}_2], \forall r \in (\Sigma \times \Sigma)^{<\omega}\, [p[r] \neq \varnothing \Rightarrow p[r] \in \mathbf{P}_1^*]$.

Proof. These propositions are proved in a fashion similar to the proof of Lemma 6. Here we prove the essential part of (3), i.e., that $\forall p_1, p_2 \in \mathbf{P}_2[\mathrm{DIC}(p_1) \wedge \mathrm{DIC}(p_1 \,\tilde{\|}\, p_2)]$. Let us show by induction on $n \in \omega$ that the following holds for every $n \in \omega$:

$$\forall p_1, p_2 \in \mathbf{P}_2[\mathrm{DIC}^{(n)}(p_1) \wedge \mathrm{DIC}^{(n)}(p_2) \Rightarrow \mathrm{DIC}^{(n)}(p_1 \,\tilde{\|}\, p_2)]. \qquad (21)$$

Induction Base. Let $p_1, p_2 \in \mathbf{P}_2$ such that $\mathrm{DIC}^{(0)}(p_1)$ and $\mathrm{DIC}^{(0)}(p_2)$, and fix $\sigma \in \Sigma$. By the definition of $\mathrm{DIC}^{(0)}(\cdot)$, there exists $\mathscr{R}_i \subseteq \wp(\mathrm{sact}(p_i, \sigma) \cap \mathbf{C})$ such that

$$\forall \Gamma[\langle \sigma, \Gamma \rangle \in p_i \Leftrightarrow \exists R \in \mathscr{R}_i[\Gamma \cap R = \varnothing]] \qquad (i = 1, 2).$$

Let $\mathscr{R} = \{R_1 \cup R_2 : R_1 \in \mathscr{R}_1 \wedge R_2 \in \mathscr{R}_2 \wedge R_1 \cap \overline{R_2} = \varnothing\}$. Then one has, by the definitions of $\tilde{\|}$ and $\#$, that $\forall \Gamma[\langle \sigma, \Gamma \rangle \in p_1 \,\tilde{\|}\, p_2 \Leftrightarrow \exists R \in \mathscr{R}[\Gamma \cap R = \varnothing]]$, which implies that $\mathrm{DIC}^{(0)}(p_1 \,\tilde{\|}\, p_2)$.

Induction Step. For every $k \in \omega$, it is immediate by the definition of $\tilde{\|}$, that (21) with $n = k+1$ follows from (21) with $n = k$. ∎

4.4. *Correctness of $\mathscr{D}_2$ with Respect to $\mathscr{O}_2$*

The correctness of $\mathscr{D}_2$ w.r.t. $\mathscr{O}_2$ is established as that of $\mathscr{D}_1$ w.r.t. $\mathscr{O}_1$, by means of an intermediate model $\tilde{\mathscr{O}}_2$.

4.4.1. *Intermediate Model for $\mathscr{L}_2$ and Semantic Equivalence*

First, the intermediate model $\tilde{\mathscr{O}}_2$, which is an alternative formulation of $\mathscr{D}_2$, is defined in terms of the transition relation $\rightarrow_2$.

DEFINITION 25 (Intermediate Model $\tilde{\mathcal{O}}_2$ for $\mathcal{L}_2$). We have the unique mapping $\tilde{\mathcal{O}}_2: \mathcal{L}_2 \to \mathbf{P}_2$ satisfying the following condition (the existence and uniqueness of such a mapping are obtained as in Definition 11): For $s \in \mathcal{L}_2$,

$$\tilde{\mathcal{O}}_2[\![s]\!] = \bigcup \{(\langle \sigma, a, \sigma' \rangle) \cdot \tilde{\mathcal{O}}_2[\![s']\!] : \langle \sigma, a, \sigma' \rangle \in \Sigma \times \mathbf{A} \times \Sigma \wedge \langle s, \sigma \rangle \xrightarrow{a}_2 \langle \sigma', \sigma' \rangle\} \cup \{(\langle \sigma, \Gamma \rangle) : \langle \sigma, \Gamma \rangle \in \Sigma \times \wp(\mathbf{C}) \wedge \tau \notin \mathrm{act}(s, \sigma) \wedge \Gamma \cap \mathrm{sact}(s, \sigma) = \varnothing\}.$$

We have the distributivity of $\tilde{\|}$ in $\mathbf{P}_2$ as we had that in $\mathbf{P}_1$ (cf. Lemma 8).

LEMMA 20 (Distributivity of $\tilde{\|}$ in $\mathbf{P}_2$). *Let* $k, l \geq 1$, $p_1, ..., p_k$, $p'_1, ..., p'_l \in \mathbf{P}_2^*$:

$$\bigcup_{i \in \bar{k}} [p_i] \;\tilde{\|}\; \bigcup_{j \in \bar{l}} [p'_j] = \bigcup_{\langle i, j \rangle \in \bar{k} \times \bar{l}} [p_i \;\tilde{\|}\; p'_j].$$

Proof. Omitted (see Appendix 5 of [HBR90]). ∎

By means of the above lemma, we will establish the equivalence between $\mathcal{D}_2$ and $\tilde{\mathcal{O}}_2$ as we have established Lemma 7.

LEMMA 21 (Semantic Equivalence for $\mathcal{L}_2$). (1) *Let F be an operator of* $\mathcal{L}_2^*$ *with arity r, and let* $s_1, ..., s_r \in \mathcal{L}_2$. *Then one has*

$$\tilde{\mathcal{O}}_2[\![F(s_1, ..., s_r)]\!] = F^{\mathscr{I}_2}(\tilde{\mathcal{O}}_2[\![s_1]\!], ..., \tilde{\mathcal{O}}_2[\![s_r]\!]).$$

(2) *For* $s \in \mathcal{L}_2$, *one has* $\tilde{\mathcal{O}}_2[\![s]\!] = \mathcal{D}_2[\![s]\!]$.

Proof. (1) The proof is similar to that of Lemma 7. Here we prove the claim for the operator $\|$. For the other operators this is proved (more straightforwardly) in a similar fashion. Let $\mathbf{H}_2 = (\mathcal{L}_2 \times \mathcal{L}_2 \to \mathbf{P}_2)$, and let $F, G \in \mathbf{H}_2$ be defined as follows: For $s_1, s_2 \in \mathcal{L}_2$, $F(s_1, s_2) = \tilde{\mathcal{O}}_2[\![s_1 \parallel s_2]\!]$, $G(s_1, s_2) = \tilde{\mathcal{O}}_2[\![s_1]\!] \;\tilde{\|}\; \tilde{\mathcal{O}}_2[\![s_2]\!]$. Moreover, let $\mathcal{F}_2^{\|}: \mathbf{H}_2 \to \mathbf{H}_2$ be defined as follows: For $f \in \mathbf{H}_2$ and $s_1, s_2 \in \mathcal{L}_2$,

$$\mathcal{F}_2^{\|}(f)(s_1, s_2) = \mathcal{F}_2^{\lfloor}(f)(s_1, s_2) \cup \mathcal{F}_2^{\lfloor}(f)(s_2, s_1) \cup \mathcal{F}_2^{\rhd}(f)(s_1, s_2) \cup \mathcal{F}_2^{\rhd}(f)(s_2, s_1) \cup \mathcal{F}_2^{\#}(f)(s_1, s_2), \quad \text{where}$$

$$\mathcal{F}_2^{\lfloor}(f)(s_1, s_2) = \bigcup \{(\langle \sigma, a, \sigma' \rangle) \cdot f(s'_1, s_2) : \langle s_1, \sigma \rangle \xrightarrow{a}_2 \langle s'_1, \sigma' \rangle\}, \quad \text{and}$$

$$\mathcal{F}_2^{\rhd}(f)(s_1, s_2) = \bigcup \{(\langle \sigma, \tau, \sigma' \rangle) \cdot f(s'_1, s'_2) : \exists c, \exists v [\langle s_1, \sigma \rangle \xrightarrow{c!v}_2 \langle s'_1, \sigma \rangle \wedge \langle s_2, \sigma \rangle \xrightarrow{c?v}_2 \langle s'_2, \sigma' \rangle]\},$$

$$\mathcal{F}_2^{\#}(f)(s_1, s_2) = \{(\langle \sigma, \Gamma \rangle) : \tau \notin \mathrm{act}(s_1, \sigma) \wedge \tau \notin \mathrm{act}(s_1, \sigma) \wedge \mathrm{sact}(s_1, \sigma) \cap \overline{\mathrm{sact}(s_2, \sigma)} = \varnothing \wedge \Gamma \cap (\mathrm{sact}(s_1, \sigma) \cup \mathrm{sact}(s_2, \sigma)) = \varnothing\}.$$

Then, $\mathcal{F}_2^{\|}$ is a contraction.

Let $s_1, s_2 \in \mathscr{L}_2$. By the definition of $\tilde{\mathcal{O}}_2$ and $\rightarrow_2$, one has $F(s_1, s_2) = \mathscr{F}_2^{\|}(F)(s_1, s_2)$; i.e., $F = \text{fix}(\mathscr{F})$. Thus, for obtaining the desired result, it suffices to show that $G = \mathscr{F}_2^{\|}(G)$. By the definition of $\tilde{\|}$, one has

$$G(s_1, s_2) = \bigcup_{\langle i, j\rangle = \langle 1,2\rangle, \langle 2,1\rangle} [(\tilde{\mathcal{O}}_2[\![s_i]\!] \mathbin{\|\!\!\lfloor} \tilde{\mathcal{O}}_2[\![s_j]\!]) \cup (\tilde{\mathcal{O}}_2[\![s_i]\!] \rhd \tilde{\mathcal{O}}_2[\![s_j]\!])]$$
$$\cup (\tilde{\mathcal{O}}_2[\![s_1]\!] \mathbin{\#} \tilde{\mathcal{O}}_1[\![s_2]\!]).$$

Thus, for showing $G = \mathscr{F}_2^{\|}(G)$, it suffices to show that $(*)$ $(\tilde{\mathcal{O}}_2[\![s_i]\!] \mathbin{\|\!\!\lfloor} \tilde{\mathcal{O}}_2[\![s_j]\!]) = \mathscr{F}_2^{\|\!\lfloor}(G)(s_i, s_j)$ $(\langle i, j\rangle = \langle 1, 2\rangle, \langle 2, 1\rangle)$, $(\dagger)$ $(\tilde{\mathcal{O}}_2[\![s_i]\!] \rhd \tilde{\mathcal{O}}_2[\![s_j]\!]) = \mathscr{F}_2^{\rhd}(G)(s_1, s_2)$ $(\langle i, j\rangle = \langle 1, 2\rangle, \langle 2, 1\rangle)$, and $(\ddagger)(\tilde{\mathcal{O}}_1[\![s_1]\!] \mathbin{\#} \tilde{\mathcal{O}}_1[\![s_2]\!]) = \mathscr{F}_2^{\#}(G)(s_1, s_2)$. The fact $(*)$ can be shown as $(*)$ in the proof of Lemma 7(1); $(\dagger)$ is shown as follows:

$$\tilde{\mathcal{O}}_2[\![s_i]\!] \rhd \tilde{\mathcal{O}}_2[\![s_j]\!]$$

$$= \bigcup \{(\langle \sigma, \tau, \sigma'\rangle) \cdot (\tilde{\mathcal{O}}_2[\![s_i]\!][\langle \sigma, c!\,v, \sigma\rangle] \mathbin{\tilde{\|}} \tilde{\mathcal{O}}_2[\![s_j]\!][\sigma, c?\,v, \sigma'\rangle]):$$
$$\tilde{\mathcal{O}}_2[\![s_i]\!][\langle \sigma, c!\,v, \sigma\rangle] \neq \varnothing \wedge \tilde{\mathcal{O}}_2[\![s_j]\!][\langle \sigma, c?\,v, \sigma'\rangle] \neq \varnothing\}$$

(taking closure is omitted, since $\text{ASFin}^{(0)}(\mathcal{O}_2[\![s_k]\!])$ and $\text{OV Fin}^{(0)}(\mathcal{O}_2[\![s_k]\!])$ $(k = 1, 2)$ by Lemma 16(2) and (3), and therefore, the above set $\bigcup \{(\langle \sigma, \tau, \sigma'\rangle) \cdots\}$ is closed)

$$= \bigcup \Big\{(\langle \sigma, \tau, \sigma'\rangle) \cdot \Big(\bigcup \{\tilde{\mathcal{O}}_2[\![s_i']\!] : \langle s_i : \sigma\rangle \xrightarrow{c!v}_2 \langle s_i', \sigma\rangle\}$$
$$\mathbin{\tilde{\|}} \bigcup \{\tilde{\mathcal{O}}_2[\![s_j']\!] : \langle s_j, \sigma\rangle \xrightarrow{c?v}_2 \langle s_j', \sigma'\rangle\}\Big):$$
$$\exists s_i'[\langle s_i, \sigma\rangle \xrightarrow{c!v}_2 \langle s_i', \sigma\rangle] \wedge \exists s_j'[\langle s_j, \sigma\rangle \xrightarrow{c?v}_2 \langle s_2', \sigma'\rangle]\Big\}$$

$$= \bigcup \Big\{(\langle \sigma, \tau, \sigma'\rangle) \cdot \Big(\bigcup \{\tilde{\mathcal{O}}_2[\![s_i']\!] \mathbin{\tilde{\|}} \tilde{\mathcal{O}}_2[\![s_j']\!] :$$
$$\langle s_i, \sigma\rangle \xrightarrow{c!v}_2 \langle s_i', \sigma\rangle \wedge \langle s_j, \sigma\rangle \xrightarrow{c?v}_2 \langle s_j', \sigma'\rangle\}\Big):$$
$$\exists s_i'[\langle s_i, \sigma\rangle \xrightarrow{c!v}_2 \langle s_i', \sigma\rangle]$$
$$\wedge \exists s_j'[\langle s_j, \sigma\rangle \xrightarrow{c?v}_2 \langle s_j', \sigma'\rangle]\Big\} \qquad \text{(by Lemma 20)}$$

$$= \mathscr{F}_2^{\rhd}(G)(s_i, s_j).$$

For showing (‡), it suffices, by the definition of #, to show the following for every $\langle\sigma, \Gamma\rangle \in \Sigma \times \wp(\mathbf{C})$:

$$\exists\langle\sigma, \Gamma_1\rangle \in \tilde{\mathcal{O}}_2[\![s_1]\!], \exists\langle\sigma, \Gamma_2\rangle \in \tilde{\mathcal{O}}_2[\![s_2]\!][(\mathbf{C}\backslash\Gamma_1) \cap \overline{(\mathbf{C}\backslash\Gamma_2)} = \varnothing \wedge \Gamma \subseteq \Gamma_1 \cap \Gamma_2]$$
$$\Leftrightarrow \tau \notin \mathrm{act}(s_1, \sigma) \wedge \tau \notin \mathrm{act}(s_2, \sigma) \wedge \mathrm{sact}(s_1, \sigma) \cap \overline{\mathrm{sact}(s_2, \sigma)} = \varnothing$$
$$\wedge\ \Gamma \cap (\mathrm{sact}(s_1, \sigma) \cup \mathrm{sact}(s_2, \sigma)) = \varnothing. \tag{22}$$

The $\Leftarrow$-part of (22) is obtained by putting $\Gamma_1 = \mathbf{C}\backslash\mathrm{sact}(s_1, \sigma)$, $\Gamma_2 = \mathbf{C}\backslash\mathrm{sact}(s_2, \sigma)$. Let us show the $\Rightarrow$-part. Suppose the left-hand side of (22) holds, and fix such Γ_1, Γ_2. By the definition of $\tilde{\mathcal{O}}_2$, (**) $\tau \notin \mathrm{act}(s_1, \sigma)$. Moreover, $\Gamma_1 \cap \mathrm{sact}(s_1, \sigma) = \varnothing$, and therefore, (††) $\mathrm{sact}(s_1, \sigma) \subseteq \mathbf{C}\backslash\Gamma_1$. Similarly (‡‡) $\tau \notin \mathrm{act}(s_2, \sigma)$, and $\mathrm{sact}(s_2, \sigma) \subseteq \mathbf{C}\backslash\Gamma_2$, i.e., (***) $\overline{\mathrm{sact}(s_2, \sigma)} \subseteq \overline{\mathbf{C}\backslash\Gamma_2}$. By the left-hand side of (22), (††), and (**), one has (†††) $\mathrm{sact}(s_1, \sigma) \cap \overline{\mathrm{sact}(s_2, \sigma)} \subseteq \mathbf{C}\backslash\Gamma_1 \cap \overline{\mathbf{C}\backslash\Gamma_2} = \varnothing$. By the left-hand side of (22), $\Gamma \subseteq \Gamma_1 \subseteq \mathbf{C}\backslash\mathrm{sact}(s_1, \sigma)$, and therefore, (‡‡‡) $\Gamma \cap \mathrm{sact}(s_1, \sigma) = \varnothing$. Similarly (****) $\Gamma \cap \mathrm{sact}(s_2, \sigma) = \varnothing$. By (**), (‡‡), (†††), (‡‡‡), (****), one has the right-hand side of (22). Thus one has (22).

(2) Similar to the proof of the part (2) of Lemma 7. ▌

4.4.2. *Correctness of $\mathcal{D}_2$ with Respect to $\mathcal{O}_2$*

As a preliminary to the proof of the correctness, an *abstraction function* $\alpha_2 \colon \mathbf{P}_2 \to (\Sigma \to \wp_{\mathrm{nc}}((\mathbf{A} \times \Sigma)^{\leqslant\omega}))$ is defined as follows. Like α_1, this function is formulated in two ways, first as the fixed point of a higher-order mapping, and second as the set of histories.

DEFINITION 26 (Abstraction Function α_2 for $\mathcal{L}_2$). We have the unique mapping $\alpha_2 : \mathbf{P}_2^* \to (\Sigma \to \wp_{\mathrm{nc}}((\mathbf{A} \times \Sigma)^{\leqslant\omega}))$ satisfying the following (the existence and uniqueness of such a mapping are obtained as in Definition 12): For every $p \in \mathbf{P}_2^*$, $\sigma \in \Sigma$,

$$\alpha_2(p)(\sigma) = \bigcup \{(\langle a, \sigma'\rangle) \cdot \alpha_2(p[\langle\sigma, a, \sigma'\rangle])(\sigma') : \exists q \in \mathbf{Q}_2[(\langle\sigma, a, \sigma'\rangle) \cdot q \in p]\}$$
$$\cup\ \mathrm{if}(\exists\Gamma \in \wp(\mathbf{C})[(\langle\sigma, \Gamma\rangle) \in p], \{\varepsilon\}, \varnothing).$$

The abstraction function is characterized in another way. First, we need some preliminary definitions.

DEFINITION 27 (Histories of Elements of $\mathbf{Q}_2$). Let $q \in \mathbf{Q}_2 \cup (\Sigma \times \mathbf{A} \times \Sigma)^{<\omega}$.

(1) The sequence q is *executable*, written $\mathrm{Exec}_2(q)$, iff

$$\exists v \in \omega \cup \{\omega\}, \exists (\langle \sigma_i, a_i, \sigma'_i \rangle)_{i \in v} [q = (\langle \sigma_i, a_i, \sigma'_i \rangle)_{i \in v}$$
$$\wedge \forall i \in v [i+1 \in v \Rightarrow \sigma'_i = \sigma_{i+1}]]$$
$$\vee \exists k \in \omega, \exists (\langle \sigma_i, a_i, \sigma'_i \rangle)_{i \in k}, \exists \langle \sigma_k, \Gamma \rangle [q = (\langle \sigma_i, a_i, \sigma'_i \rangle)_{i \in k}$$
$$\cdot (\langle \sigma_k, \Gamma \rangle) \wedge \forall i \in k [\sigma'_i = \sigma_{i+1}]].$$

Let $\mathbf{E}_2 = \{q \in \mathbf{Q}_2 \cup (\Sigma \times \mathbf{A} \times \Sigma)^{<\omega} : \mathrm{Exec}_2(q)\}$. For $\sigma \in \Sigma$, let $\mathbf{E}_2\langle \sigma \rangle = \{q \in \mathbf{E}_2 \backslash \{\varepsilon\} : \mathrm{istate}_2(q) = \sigma\}$.

(2) Let q be executable. The *history* of q, denoted by $\mathrm{hist}_2(q)$, is defined by

$$\mathrm{hist}_2(q) = \begin{cases} (\langle a_i, \sigma'_i \rangle)_{i \in v} & \text{if } q = (\langle \sigma_i, a_i, \sigma'_i \rangle)_{i \in v}, \\ (\langle a_i, \sigma'_i \rangle)_{i \in k} & \text{if } q = (\langle \sigma_i, a_i, \sigma'_i \rangle)_{i \in k} \cdot (\langle \sigma_k, \Gamma \rangle). \end{cases}$$

The next lemma is shown in a fashion similar to Lemma 9.

LEMMA 22 (Another Formulation of the Abstraction Function α_2). (1) *For* $p \in \mathbf{P}_2^*$, $\sigma \in \Sigma$, *one has* $\alpha_2(p)(\sigma) = \{\mathrm{hist}_2(q) : q \in p \cap \mathbf{E}_2\langle \sigma \rangle\}$.

(2) $\forall k \geq 1, \forall p_1, ..., p_k \in \mathbf{P}_2^*, \forall \sigma [\alpha_2(\bigcup_{i \in \bar{k}} [p_i])(\sigma) = \bigcup_{i \in \bar{k}} [\alpha_2(p_i)(\sigma)]]$.

By means of this lemma, we have the correctness of $\mathscr{D}_2$.

LEMMA 23 (Correctness of $\mathscr{D}_2$). (1) $\alpha_2 \circ \tilde{\mathcal{O}}_2 = \mathcal{O}_2$.

(2) $\alpha_2 \circ \mathscr{D}_2 = \mathcal{O}_2$.

Proof. (1) By showing that $\alpha_2 \circ \tilde{\mathcal{O}}_2$ is the fixed point of $\Psi_2^{\mathcal{O}}$ defined in Definition 20.

(2) Immediate from (1) and Lemma 21(2). ∎

4.5. *Full Abstraction of $\mathscr{D}_2$ with Respect to $\mathcal{O}_2$*

As for $\mathscr{L}_1$, we present the following lemma to establish the full abstraction of $\mathscr{D}_2$;

LEMMA 24 (Uniform Distinction Lemma for $\mathscr{L}_2$). *Let* $\mathscr{X} \in (\wp_{\mathrm{f}}(\mathrm{IVar}) \backslash \{\varnothing\})$.

(1) *For every* $r \in (\Sigma_{\mathscr{X}} \times \mathbf{A} \times \Sigma_{\mathscr{X}})^{<\omega}$,

$$\forall p_1, p_2 \in \mathbf{P}_2^*[p_1[r] \neq \emptyset \wedge p_2[r] = \emptyset \Rightarrow \forall \sigma \in \Sigma_{\mathscr{X}}, \exists T \in \mathscr{L}_2[\alpha_2(p_1 \,\tilde{\|}\, \mathscr{D}_2[\![T]\!])(\sigma) \backslash \alpha_2(p_2 \,\tilde{\|}\, \mathscr{D}_2[\![T]\!])(\sigma) \neq \emptyset]]. \tag{23}$$

(2) *For every* $q \in (\Sigma_{\mathscr{X}} \times \mathbf{A} \times \Sigma_{\mathscr{X}})^{<\omega} \cdot (\Sigma_{\mathscr{X}} \times \wp(\mathbf{C}))$,

$$\forall p_1, p_2 \in \mathbf{P}_2^*[q \in p_1 \backslash p_2 \Rightarrow \forall \sigma \in \Sigma_{\mathscr{X}}, \exists T \in \mathscr{L}_2[\alpha_2(p_1 \,\tilde{\|}\, \mathscr{D}_2[\![T]\!])(\sigma) \backslash \alpha_2(p_2 \,\tilde{\|}\, \mathscr{D}_2[\![T]\!])(\sigma) \neq \emptyset]]. \tag{24}$$

The proof of this lemma is given later. First, note that the full abstraction of $\mathscr{D}_2$ follows immediately from Lemma 24, in the same way as Theorem 1 follows from Lemma 12.

THEOREM 2 (Full Abstraction of $\mathscr{D}_2$). *Let* $\mathbf{V}$ *be infinite. Then, for every* $s_1, s_2 \in \mathscr{L}_2$, *one has*

$$\mathscr{D}_2[\![s_1]\!] \neq \mathscr{D}_2[\![s_2]\!] \Rightarrow \exists T \in \mathscr{L}_2[\alpha_2(\mathscr{D}_2[\![s_1]\!] \,\|\, \mathscr{D}_2[\![T]\!]) \neq \alpha_2(\mathscr{D}_2[\![s_2]\!] \,\|\, \mathscr{D}_2[\![T]\!])].$$

We present the following lemma as a preliminary to the proof of Lemma 24. For its proof we assume that $\mathbf{V}$ is infinite.

LEMMA 25 (Testing Lemma for $\mathscr{L}_2$). *Let* $\mathscr{X} \in (\wp_f(\mathrm{IVar}) \backslash \{\emptyset\})$, $p \in \mathbf{P}_2^*$, $\langle \sigma', a, \sigma'' \rangle \in (\Sigma_{\mathscr{X}} \times \mathbf{A} \times \Sigma_{\mathscr{X}})$, $\sigma_0 \in \Sigma_{\mathscr{X}}$. *Then there are two finite sequences* $r_1, r_2 \in (\Sigma_{\mathscr{X}} \times \mathbf{A} \times \Sigma_{\mathscr{X}})^{<\omega}$ *such that the following hold*:

(1) $r_1 \cdot \langle \sigma', a, \sigma'' \rangle \cdot r_2 \in \mathbf{E}_2 \langle \sigma_0 \rangle$.

(2) *For every tester* $T' \in \mathscr{L}_2$, *there exists another tester* $T \in \mathscr{L}_2$ *such that the following hold*:

(i) $\mathscr{D}_2[\![T]\!][r_1 \cdot r_2] = \mathscr{D}_2[\![T']\!]$,

(ii) $\forall q' \in \mathbf{Q}_2[r_1 \cdot \langle \sigma', a, \sigma'' \rangle \cdot r_2 \cdot q' \in p \,\tilde{\|}\, \mathscr{D}_2[\![T]\!] \Rightarrow p[\langle \sigma', a, \sigma'' \rangle] \neq \emptyset \wedge q' \in p[\langle \sigma', a, \sigma'' \rangle] \,\tilde{\|}\, \mathscr{D}_2[\![T']\!]]$.

Proof. The proof is formulated by supposing that $\mathscr{X}$ is reduced to one variable: $\mathscr{X} = \{x\}$, as Lemma 13. However, the lemma still holds when $\mathscr{X}$ is composed of more than one variable, as Lemma 13. For $v \in \mathbf{V}$, let $\bar{\sigma}(v)$ be defined as in Lemma 13. The proof is given by distinguishing two cases according to whether $\sigma_0(x) = \sigma'(x)$.

Case 1. When $\sigma_0(x) = \sigma'(x)$, we can easily construct two sequences r_1, r_2 satisfying (1), (2) of Lemma 25 as follows: $r_1 = \varepsilon$ and $r_2 = \langle \sigma'', \tau, \bar{\sigma}(v_1) \rangle$, where v_1 is chosen such that

$$\text{(i)}\quad v_1 \neq \sigma''(x), \qquad \text{(ii)}\quad v_1 \notin \{v \in \mathbf{V} : \langle \sigma', a, \sigma'' \rangle \cdot \langle \sigma'', \tau, \bar{\sigma}(v) \rangle \in p^{[2]}\}. \tag{25}$$

Note that the right-hand side of (25)(ii) is finite by Definition 24, and therefore, there is v_1 satisfying (25). It is shown that (1) and (2) of Lemma 25 hold in a similar fashion to the corresponding part in the proof of Lemma 13.

Case 2. When $\sigma_0(x) \neq \sigma'(x)$, we can construct two sequences r_1, r_2, satisfying (1) and (2) of Lemma 25 as follows: $r_1 = \langle \sigma_0, \tau, \sigma' \rangle$, $r_2 = \langle \sigma'', \tau, \bar{\sigma}(v_1) \rangle$, where v_1 is chosen such that

$$\begin{cases} \text{(i)} & v_1 \notin \{v \in \mathbf{V} : \langle \sigma_0, \tau, \sigma'' \rangle \cdot \langle \sigma', a, \sigma'' \rangle \cdot \langle \sigma'', \tau, \bar{\sigma}(v) \rangle \in p^{[3]}\}, \\ \text{(ii)} & v_1 \neq \sigma'(x), \\ \text{(iii)} & v_1 \neq \sigma''(x), \\ \text{(iv)} & v_1 \notin \{v \in \mathbf{V} : \langle \sigma', a, \sigma'' \rangle \cdot \langle \sigma'', \tau, \bar{\sigma}(v) \rangle \in p^{[2]}\}. \end{cases} \tag{26}$$

Note that the right-hand sides of (26)(i) and (iv) are finite by Definition 24, and therefore, there is v_1 satisfying (26). In this case also, it is shown that (1) and (2) of Lemma 25 hold in a similar fashion to the corresponding part in the proof of Lemma 13. ∎

The following proposition follows immediately from Lemma 25 as Corollary 1 followed from Lemma 13; this corollary is to play a central role in the proof of Lemma 24.

COROLLARY 2. *Let* $\mathscr{X} \in (\wp_f(\text{IVar}) \backslash \{\varnothing\})$, $p \in \mathbf{P}_2^*$, $\langle \sigma', a, \sigma'' \rangle \in (\Sigma_{\mathscr{X}} \times \mathbf{A} \times \Sigma_{\mathscr{X}})$, *and* $\sigma_0 \in \Sigma_{\mathscr{X}}$. *Then, there are* $\rho_1, \rho_2 \in (\mathbf{A} \times \Sigma_{\mathscr{X}})^{<\omega}$ *such that for every* $T' \in \mathscr{L}_2$ *there exists* $T \in \mathscr{L}_2$ *such that, putting* $\sigma_1 = \text{last}(\rho_1 \cdot \sigma'' \cdot \rho_2)$, *the following hold*:

(1) *For every* $p' \in \mathbf{P}_2^*$, *one has*

$$\begin{aligned} \forall \rho' \in (\mathbf{A} \times \Sigma)^{\leqslant \omega}\ [&p'[\langle \sigma', a, \sigma'' \rangle] \neq \varnothing \\ &\wedge \rho' \in \alpha_2(p'[\langle \sigma', a, \sigma'' \rangle] \mathbin{\tilde{\|}} \mathscr{D}_2[\![T']\!])(\sigma_1) \\ &\Rightarrow \rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_2(p' \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma_0)]. \end{aligned} \tag{27}$$

(2) *For $p' = p$, one has the converse of* (28). *That is,*

$$\forall \rho' \in (\mathbf{A} \times \Sigma)^{\leqslant \omega} [\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_2(p \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma_0)$$
$$\Rightarrow p[\langle \sigma', a, \sigma'' \rangle] \neq \emptyset \wedge \rho' \in \alpha_2(p[\langle \sigma', a, \sigma'' \rangle] \mathbin{\tilde{\|}} \mathscr{D}_2[\![T']\!])(\sigma_1)]. \quad (28)$$

Proof of Lemma 24. Let $\mathscr{X} \in (\wp_f(\mathrm{IVar}) \backslash \{\emptyset\})$.

Part (1). The first part is proved by means of Corollary 2, as Lemma 12(1) was proved by means of Corollary 1, by induction on the length of $r \in (\Sigma_{\mathscr{X}} \times \mathbf{A} \times \Sigma_{\mathscr{X}})^{<\omega}$.

Part (2). We will prove that (24) holds for every $q \in (\Sigma_{\mathscr{X}} \times \mathbf{A} \times \Sigma_{\mathscr{X}})^{<\omega} \cdot (\Sigma_{\mathscr{X}} \times \wp(\mathbf{C}))$, by induction on the length of q. The proof is similar to the proof of the corresponding part of Lemma 12 except for the induction base, which is established by means of the method of [BKO88] with some adaptation to the present setting; the induction step can be established using the testing method (Corollary 2).

Induction Base. Let $\mathrm{lgt}(q) = 1$ and $q = (\langle \sigma', \Gamma' \rangle)$. Suppose $q \in p_1$ and $q \notin p_2$, and let $\sigma \in \Sigma_{\mathscr{X}}$. We will construct a tester T such that $(\langle \tau, \sigma' \rangle) \in \alpha_2(p_1 \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma) \backslash \alpha_2(p_2 \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma)$. Since p_2 satisfies the disjointness inaction condition, there exists $\mathscr{R}_2$ such that (*) $\mathscr{R}_2 \subseteq \wp(\mathrm{sact}(p_2, \sigma) \cap \mathbf{C})$ and (†) $\forall \Gamma \in \wp(\mathbf{C})[(\langle \sigma', \Gamma \rangle) \in p_2 \Leftrightarrow \exists R \in \mathscr{R}_2[\Gamma \cap R = \emptyset]]$. Fix such an $\mathscr{R}_2$, and let (‡) $\Gamma'' = \mathrm{sact}(p_2, \sigma) \cap \Gamma'$. By (†) and the fact that $q \notin p_2$, one has $\forall R' \in \mathscr{R}_2[\Gamma' \cap R' \neq \emptyset]$. The set $\mathrm{sact}(p_2, \sigma)$ is finite since $\mathrm{ASFin}(p_2)$, which implies that Γ'' is finite. Let $\Gamma'' = \{\gamma_1, ..., \gamma_n\}$. Since $\mathscr{X}$ is finite and nonempty, we can put $\mathscr{X} = \{x_1, ..., x_r\}$ as in the proof of Lemma 12. Let us set $T \equiv (x_1 := \sigma'(x_1))$; ...; $(x_r := \sigma'(x_r))$; T', and $T' \equiv \mathbf{0} + \phi(\overline{\gamma_1}) + \cdots + \phi(\overline{\gamma_n})$, where $\phi(\gamma) = (c\,!\,v)$; $\mathbf{0}$ if $\gamma = c!$ with $v \in \mathbf{V}$ arbitrary, and $\phi(\gamma) = (c\,?\,x)$; $\mathbf{0}$ if $\gamma = c?$ with $x \in \mathrm{IVar}$ arbitrary. With this tester T, we will show that $(\langle \tau, \sigma'_1 \rangle, ..., \langle \tau, \sigma'_r \rangle) \in \alpha_2(p_1 \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma) \backslash \alpha_2(p_2 \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma)$, where $\sigma'_i = \sigma[(\sigma'(x_1), ..., \sigma'(x_i))/(x_1, ..., x_i)]$ $(i \in r+1)$.

First, let us show that $(\langle \tau, \sigma'_1 \rangle, ..., \langle \tau, \sigma'_r \rangle) \in \alpha_2(p_2 \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma)$. Under the assumption that $q \in p_1$, one has (**) $(\langle \sigma', \Gamma' \rangle) \in p_1$. Moreover, by the definition of T', one has that (††) $(\langle \sigma', \mathbf{C} \backslash \overline{\Gamma''} \rangle) \in \mathscr{D}_2[\![T']\!]$. Moreover,

$$(\mathbf{C} \backslash \Gamma') \cap \overline{(\mathbf{C} \backslash (\mathbf{C} \backslash \overline{\Gamma''}))} = (\mathbf{C} \backslash \Gamma') \cap \Gamma''$$
$$= (\mathbf{C} \backslash \Gamma') \cap \mathrm{sact}(p_2, \sigma) \cap \Gamma' \text{ (by (‡))} = \emptyset.$$

By this (**), (††), and the definitions of $\tilde{\|}$ and $\#$, one has that $(\langle \sigma'_0, \tau, \sigma'_1 \rangle, ..., \langle \sigma'_{r-1}, \tau, \sigma'_r \rangle, \langle \sigma', \emptyset \rangle) \in p_1 \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!]$, i.e., (‡‡): $(\langle \tau, \sigma'_1 \rangle, ..., \langle \tau, \sigma'_r \rangle) \in \alpha_2(p_1 \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma)$.

Next let us show, by contradiction, that $(\langle \tau, \sigma'_1 \rangle, ..., \langle \tau, \sigma'_r \rangle) \notin \alpha_2(p_2 \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma)$. Assume, to the contrary, that (***) $(\langle \tau, \sigma'_1 \rangle, ..., \langle \tau, \sigma'_r \rangle) \in \alpha_2(p_2 \mathbin{\tilde{\|}} \mathscr{D}_2[\![T]\!])(\sigma)$. Then, by the definition of α_2, one has that

$(\langle\sigma'_0, \tau, \sigma'_1\rangle, ..., \langle\sigma'_{r-1}, \tau, \sigma'_r\rangle, \langle\sigma', \varnothing\rangle) \in p_2 \tilde{\|} \mathscr{D}_2[\![T]\!]$. Hence, $(\langle\sigma', \varnothing\rangle) \in p_2 \tilde{\|} \mathscr{D}_2[\![T]\!][(\langle\sigma'_0, \tau, \sigma'_1\rangle, ..., \langle\sigma'_{r-1}, \tau, \sigma'_r\rangle)] = p_2 \tilde{\|} \mathscr{D}_2[\![T']\!]$. By this and the definitions of $\tilde{\|}$ and $\#$, there exist $\Gamma_1, \Gamma_2 \in \wp(\mathbf{C})$ such that

$$\begin{aligned} &\text{(i)} \quad (\langle\sigma', \Gamma_1\rangle) \in p_2, \\ &\text{(ii)} \quad (\langle\sigma', \Gamma_2\rangle) \in \mathscr{D}_2[\![T']\!], \\ &\text{(iii)} \quad (\mathbf{C}\backslash\Gamma_1) \cap \overline{(\mathbf{C}\backslash\Gamma_2)} = \varnothing. \end{aligned} \tag{29}$$

Moreover, there exists $R' \in \mathscr{R}_2$ such that $\Gamma_1 \cap R' = \varnothing$. Fix such R'. Then (†††) $\mathbf{C}\backslash\Gamma_1 \supseteq R'$. By the fact that $(\langle\sigma', \Gamma'\rangle) \notin p_2$, one has that (‡‡‡) $\Gamma' \cap R' \neq \varnothing$. By (29)(ii), one has that $\Gamma_2 \cap \overline{\Gamma''} = \varnothing$, i.e., $\mathbf{C}\backslash\Gamma_2 \supseteq \overline{\Gamma''}$, and therefore (∗∗∗∗) $\overline{\mathbf{C}\backslash\Gamma_2} \supseteq \Gamma''$. Thus

$$\begin{aligned} (\mathbf{C}\backslash\Gamma_1) \cap \overline{\mathbf{C}\backslash\Gamma_2} &\supseteq R' \cap \Gamma'' \qquad \text{(by (†††) and (∗∗∗∗))} \\ &= R' \cap (\text{sact}(p_2, \sigma) \cap \Gamma') \qquad \text{(by (‡))} \\ &= R' \cap \Gamma' \ (\text{since } R' \subseteq \text{sact}(p_2, \sigma) \text{ by } (*)) \neq \varnothing \qquad \text{(by (‡‡‡))}. \end{aligned}$$

This contradicts (29)(iii). Hence (∗∗∗) is false, and therefore, one has that $(\langle\tau, \sigma'_1\rangle, ..., \langle\tau, \sigma'_r\rangle) \notin \alpha_2(p_2 \tilde{\|} \mathscr{D}_2[\![T]\!])(\sigma)$. By this and (††), one has that $(\langle\tau, \sigma'_a\rangle, ..., \langle\tau, \sigma'_r\rangle) \in \alpha_2(p_1 \tilde{\|} \mathscr{D}_2[\![T]\!])(\sigma)\backslash\alpha_2(p_2 \tilde{\|} \mathscr{D}_2[\![T]\!])(\sigma)$.

Induction Step. By means of Corollary 2, the induction step is established, in a similar fashion to the induction step of the proof of Lemma 12(1).

4.6. *Comparison of $\mathscr{D}_2$ and Roscoe's Model for Occam*

Roscoe, in [Ros84], constructed a denotational model for a large subset of occam. The language in [Ros84] is similar to $\mathscr{L}_2$ in many respects. However there are several differences between the two: One major difference is that, unlike individual variables in $\mathscr{L}_2$, variables in occam (except read-only ones) are not shared by two or more parallel processes, and therefore, intermediate states of one process cannot directly affect another process. Thus, in [Ros84], a denotational model $\mathscr{C}$ can be constructed (for the language) without taking account of intermediate states: The model $\mathscr{C}$ is constructed as a hybrid of the failures model for CSP (proposed in [BHR84] and improved in [BR84]), and the conventional model for sequential languages which defines the meaning of a program as a relation between initial and final states. We expect that a model for $\mathscr{L}_2$ can be constructed along the lines of $\mathscr{C}$, and will be more abstract than $\mathscr{D}_2$ in nature. However, it will not be compositional w.r.t. $\|$, since processes of $\mathscr{L}_2$ have shared variables.

5. Concluding Remarks

We conclude this paper with some remarks about possible extensions of the reported results and related works. There are two directions for such extensions. One is to investigate fully abstract models for other languages, e.g., a nonuniform concurrent language with *process creation* and (a form of) *local variables* as the language $\mathscr{L}_3$ in [BR91]. The other is to investigate fully abstract denotational models for the same language $\mathscr{L}_1$ (or $\mathscr{L}_2$) w.r.t. other operational models, which might be more abstract than the one treated in this paper.

For instance, it might be possible to construct a fully abstract denotational model for an operational model $\mathscr{B}'$ for $\mathscr{L}_1$ which is defined by slightly modifying $\mathscr{B}$ in Sectin 3.6.3 as follows: For every statement s and state σ, $\mathscr{B}'[\![s]\!](\sigma) = \{\sigma' : \exists s'[\langle s, \sigma)(\rightarrow_1)^* \langle s', \sigma' \rangle \wedge \neg\exists \langle s'', \sigma'' \rangle [\langle s', \sigma' \rangle \rightarrow_1 \langle s'', \sigma'' \rangle]]\} \cup \text{if}(\exists(\langle s_n, \sigma_n \rangle)_{n \in \omega} [\langle s_0, \sigma_0 \rangle = \langle s, \sigma \rangle \wedge \forall n \in \omega [\langle s_n, \sigma_n \rangle \rightarrow_1 \langle s_{n+1}, \sigma_{n+1} \rangle]], \{\perp\}, \emptyset)$. It was shown in [AP86] that there is no fully abstract denotational model w.r.t. $\mathscr{B}'$ if the language has *countable nondeterminism.* However, it is still to be investigated whether there is a fully abstract denotational model w.r.t. $\mathscr{B}'$, since the language $\mathscr{L}_1$ does not have counable nondeterminism. It seems that $\mathscr{D}_1$ is not fully abstract w.r.t. $\mathscr{B}'$; at least, we cannot establish the full abstraction w.r.t. $\mathscr{B}'$ as we have done w.r.t. $\mathscr{O}_1$, since there are $s_1, s_2 \in \mathscr{L}_1$ such that $\mathscr{D}_1[\![s_1]\!] \neq \mathscr{D}_1[\![s_2]\!]$, but $\forall T \in \mathscr{L}_1[\mathscr{B}'[\![s_1 \parallel T]\!] = \mathscr{B}'[\![s_2 \parallel T]\!]]$. This is easily verified by putting $s_1 \equiv \mathbf{0}$ and $s_2 \equiv (x := x); \mathbf{0}$.

For $\mathscr{L}_2$, a language for communicating concurrent systems, there are several possible operational models besides $\mathscr{O}_2$, defined in Section 4. There are several dimensions for classifying operational model for such a language; such a classification and comparative study of these models were presented in [Gla90]. One of those dimensions is the dichotomy of *linear time* versus *branching time*: a model is called a linear time model, if it identifies processes differing only in the branching structure of their execution paths; otherwise it is called a branching time model. Another dimension is the dichotomy of *weak* versus *strong*: a model is called weak, if it identifies processes differing only in their internal or silent actions (denoted by τ in this paper); otherwise it is called strong. Also, there are two kinds of languages, i.e., *uniform* languages and *nonuniform* languages. By combination of these criteria, one has eight types of operational models, and for each of them, one has the problem of constructing a fully abstract denotational model, or of characterizing somehow the fully abstract compositional model. The results on these problems obtained so far are summarized in Table 1.

As described in the introduction, fully abstract model for uniform languages w.r.t. strong operational models of the linear time variety were

TABLE I
Results on Fully Abstract Models for Communicating Processes

Linear Time	Strong	Uniform	[BKO88]: Characterization of a fully abstract compositional model.*1 [Rut89]: Construction of a fully abstract denotational model.*2
		Nonuniform	This paper: Construction of a fully abstract denotational model w.r.t. an operational model *with states*.*3 ?: With respect to an operational model *without states*.*4
	Weak	Uniform	[Hor91]: Characterization of fully abstract models for a CCS-like language.*5
		Nonuniform	?*6
Branching Time	Strong	Uniform	[Mil80, Mil85, Mil89]: Characterization of a fully abstract compositional model for CCS.*7 [GV88]: Characterization of fully abstract compositional models in general.*8 [Rut90]: Construction of fully abstract denotational models.*9
		Nonuniform	?
	Weak	Uniform	[Mil80, Mil85, Mil89]: Characterization of a fully abstract compositional model. · 10
		Nonuniform	?

investigated in [BKO88] and [Rut89] (cf.. *1, *2 in Table 1). The operational model $\mathcal{O}_2$ for a nonuniform language introduced in Section 4 is a strong model of the linear time variety. Also, it involves information about *states*. A fully abstract denotational model w.r.t. this is presented in this paper (cf. *3 in Table 1).

We can define a more abstract operational model $\mathcal{O}_2^*$ for $\mathcal{L}_2$ by ignoring states as follows: For every statement s and state σ, $\mathcal{O}_2^*[\![s]\!](\sigma) = \bigcup \{(a) \cdot \mathcal{O}_2^*[\![s']\!](\sigma') : \langle s, \sigma \rangle \xrightarrow{a}_2 \langle s', \sigma'' \rangle\} \cup \text{if}(\tau \notin \text{act}(s, \sigma), \{\varepsilon\}, \emptyset)$. It is to be investigated whether $\mathcal{D}_2$ is fully abstract w.r.t. $\mathcal{O}_2^*$ (cf *4 in Table 1). It seems more difficult to construct fully abstract denotational models w.r.t. weak operational models. A weak operational model $\mathcal{O}_2^{**}$ for $\mathcal{L}_2$ is defined by means of $\mathcal{O}_2^*$ as follows: For every statement s and state σ, $\mathcal{O}_2^{**}[\![s]\!](\sigma) = \{\rho \backslash \tau : \rho \in \mathcal{O}_2^*[\![s]\!](\sigma)\}$, where $\rho\backslash\tau$ is the result of ignoring τ's in $\rho \in (\mathbf{C} \cup \{\tau\})^{\leq \omega}$. In [Hor91], fully abstract models for CCS-like languages were constructed w.r.t. weak linear semantics with divergence, in the uniform setting (cf. *5 in Table 1); it remains for future research to construct such models in the nonuniform setting (cf. *6 in Table 1). A related discussion is found in the last section of [BKO88].

In [Mil80, Mil85, Mil89], Milner showed that a strong operational model for CCS of the branching time variety is compositional (cf. $*7$ in Table 1). Moreover, it was shown in [GV88] that branching time and strong operational models are in general compositional under certain conditions (cf. $*8$ in Table 1). Denotational models equivalent to those operational models were presented in [Rut90]; the denotational models are fully abstract w.r.t. the operational models by definition (cf. $*9$ in Table 1).

In [Mil80], [Mil85], and [Mil89], Milner characterized a fully abstract compositional model for CCS w.r.t. *observation equivalence* $\approx$ (cf. $*10$ in Table 1). This relation $\approx$ is a weak operational equivalence relation of the branching time variety. Milner characterized *observation congruence* $\approx^c$, which is the coarsest congruence relation included in $\approx$, as follows: For every two statements s_1 and s_2, $s_1 \approx^c s_2$ iff $\forall a \in \mathrm{Act}[\bigwedge_{\langle i,j\rangle = \langle 1,2\rangle, \langle 2,1\rangle} [\forall s'[s_i \xrightarrow{a} s' \Rightarrow \exists s''[s_j(\xrightarrow{\tau})^* \xrightarrow{a} (\xrightarrow{\tau})^* s'' \wedge s' \approx s'']]]]$, where Act is the set of all actions including τ (cf. [Mil89, Definition 7.2]). While this model is not denotational in the sense explained in the introduction, it seems worthwhile to investigate whether such a characterization is possible in the linear time setting.

The full abstraction problem can be treated in another framework, i.e., in the setting of complete partial ordered sets or complete lattices. For a treatment of the full abstraction problem for a concurrent language in this setting see [HP79]. In [Hen88], which is based on [DH83, Hen83, Hen 85], Henessy showed in detail the full abstraction of a denotational model consisting of *acceptance trees* equipped with a complete partial order, w.r.t. *testing equivalence.*

For a survey of the full abstraction problem for sequential languages, see [BCL85]. In [St86], the general question concerning the existence of fully abstract models was treated in an algebraic context.

Acknowledgments

We thank the Amsterdam Concurrency Group, including Frank de Boer, Arie de Bruin, Jean-Marie Jacquet, Peter Knijnenberg, Joost Kok, Erik de Vink, and Jeroen Warmerdam for helpful discussions.

Received August 6, 1990; final manuscript received July 15, 1992

References

[AP86] Apt, K., and Plotkin, G. (1986), Countable nondeterminism and random assignment, *J. Assoc. Comput. Mach.* **33**, 724–767.

[ABKR89] AMERICA, P., DE BAKKER, J. W., KOK, J. N., AND RUTTEN, J. J. M. M. (1989), Denotational semantics of a parallel object-oriented language, *Inform. and Comput.* **83**, 152–205.

[AR89] AMERICA, P., AND RUTTEN, J. J. M. M. (1989), Solving reflexive domain in a category of complete metric spaces, *J. Comput. System Sci.* **39**, 343–375.

[Bak91] DE BAKKER, J. W. (1991), Comparative semantics for flow of control in logic programming without logic, *Inform. and Comput.* **94**, 123–179.

[BM88] DE BAKKER, J. W., AND MEYER, J.-J. CH. (1988), Metric semantics for concurrency, *BIT* **28**, 504–529.

[BR91] DE BAKKER, J. W., AND RUTTEN, J. J. M. M. (1991), Concurrency semantics based on metric domain equations, *in* "Topology and Category Theory in Computer Science" (G. M. Reed, A. W. Roscoe, R. F. Wachter, Eds.), pp. 113–151, Oxford Univ. Press, London.

[BZ82] DE BAKKER, J. W., AND ZUCKER, J. I. (1982), Processes and the denotational semantics of concurrency, *Inform. and Control* **54**, 70–120.

[BKO88] BERGSTRA, J. A., KLOP, J. W., AND OLDEROG, E.-R. (1988), Readies and failures in the algebra of communicating processes, *SIAM J. Comput.* **17**, No. 6, 1134–1177.

[BCL85] BERRY, G., CURIEN, P. L., AND LEVY, J. (1985), Full abstraction for sequential languages: The state of the art, *in* "Algebraic Methods in Semantics" (M. Nivat and J. C. Reynolds, Eds.), pp. 90–132, Cambridge Univ. Press, London/New York.

[BHR84] BROOKES, S. D., HOARE, C. A. R., AND ROSCOE, A. W. (1984), A theory of communicating sequential processes, *J. Assoc. Comput. Mach.* **31**, 560–599.

[BR84] BROOKES, S. D., AND ROSCOE, A. W. (1984), An improved failures model for communicating processes, *in* "Lecture Notes in Computer Science," Vol. 197, pp. 281–305, Springer-Verlag, Berlin/New York.

[DH83] DE NICOLA, R., AND HENNESSY M. (1983), Testing equivalence and processes, *Theoret. Comput. Sci.* **34**, 83–133.

[Dug66] DUGUNDJI, J. (1966), "Topology," Allyn & Bacon, Boston.

[Eng77] ENGELKING, R. (1977), "General Topology," Polish Scientific Publishers.

[Gla90] VAN GLABBEEK, R. J. (1990), "Comparative Concurrency Semantics and Refinement of Actions," Ph.D. Thesis, Free University of Amsterdam.

[GV88] GROOTE, J. F., AND VAANDRAGER, F. (1988), "Structured Operational Semantics and Bisimulation as a Congruence," Technical Report CS-R8845, Centre for Mathematics and Computer Science, Amsterdam, to appear in *Inform. and Comput.*; extended abstract *in* "Proceedings 16th ICALP, Stresa," pp. 423–438, Lecture Notes in Computer Science, Vol. 372, Springer-Verlag, Berlin/New York.

[Hen83] HENNESSY, M. (1983), Synchronous and asynchronous experiments on processes, *Inform. and Control* **59**, 36–83.

[Hen85] HENNESSY, M. (1985), Acceptance trees, *J. Assoc. Comput. Mach.* **32**, 896–928.

[Hen88] HENNESSY, M. (1988), "Algebraic Theory of Processes," MIT Press, Cambridge, MA.

[HP79] HENNESSY, M., AND PLOTKIN, G. D. (1979), Full abstraction for a simple parallel programming language, *in* "Proceedings, 8th MFCS" (J. Bečvář, Ed.), pp. 108–120, Lecture Notes in Computer Science, Vol. 74, Springer-Verlag, Berlin/New York.

[Hor91] HORITA, E. (1991), Fully abstract models for communicating processes with respect to weak linear semantics with divergence, *IEICE Trans. Inform. Systems* **E75-D**, No. 1, 64–77.

[HBR90] HORITA, E., DE BAKKER, J. W., AND RUTTEN, J. J. M. M. (1990), "Fully Abstract Denotational Models for Nonuniform Concurrent Languages," CWI Report CS-R9027, Amsterdam.

[KR90] KOK, J. N., AND RUTTEN, J. J. M. M. (1990), Contraction in comparing concurrency semantics, in *Theoret. Comput. Sci.* **76**, 179–222.

[Mil73] MILNER, R. (1973), Processes: A mathematical model of computing agents, *in* "Proceedings of Logic Colloquium 73" (H. E. Rose and J. C. Shepherdson, Eds.), pp. 157–173, North-Holland, Amsterdam.

[Mil77] MILNER, R. (1977), Fully abstract models of typed lambda-calculi, *Theoret. Comput. Sci.* **4**, 1–22.

[Mil80] MILNER, R. (1980), "A Calculus of Communicating Systems," Lecture Notes in Computer Science, Vol. 92, Springer-Verlag, Berlin/New York.

[Mil85] MILNER, R. (1985), Lectures on a calculus for communicating systems, *in* "Seminar on Concurrency" (S. D. Brookes, A. W. Roscoe, and G. Winskel, Eds.), pp. 197–220, Lecture Notes in Computer Science, Vol. 197, Springer-Verlag, Berlin/New York.

[Mil89] MILNER, R. (1989), "Communication and Concurrency," Prentice–Hall International, Englewood Cliffs, NJ.

[Mu85] MULMULEY, K., (1985), "Full Abstraction and Semantic Equivalence," Ph.D. Thesis, Report CMU-CS-85-148, Computer Science Department, Carnegie Mellon University, Pittsburgh.

[Niv79] NIVAT, M. (1979), Infinite words, infinite trees, infinite computations, *in* "Foundations of Computer Science III, Part 2" (J. W. de Bakker and J. van Leeuwen, Eds.), Mathematical Centre Transactions, Vol. 109, Centre for Mathematics and Computer Science.

[Plo81] PLOTKIN, G. D. (1981), "A structured Approach to Operational Semantics," Report DAIMI FN-19, Computer Science Department, Aarhus University.

[Ros84] ROSCOE, A. W., (1984), Denotational semantics for Occam, *in* "Seminar on Concurrency" (S. D. Brookes, A. W. Roscoe, and G. Winskel, Eds.), pp. 306–329, Lecture Notes in Computer Science, Vol. 197, Springer-Verlag, Berlin/New York.

[Rut89] RUTTEN, J. J. M. M. (1989), Correctness and full abstraction of metric semantics for concurrency, *in* "Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency" (J. W. de Bakker, W. P. de Roever, G. Rozenberg, Eds.), pp. 628–658, Lecture Notes in Computer Science, Vol. 354, Springer-Verlag, Berlin/New York.

[Rut90] RUTTEN J. J. M. M. (1990), Deriving denotational models for bisimulation from structured operational semantics, *in* "Programming Concepts and Methods, Proceedings of the IFIP Working Group 2.2/2.3 Working Conference" (M. Broy and C. B. Jones, Eds.), pp. 148–170, North-Holland, Amsterdam.

[St86] STOUGHTON, A. (1986), Fully Abstract Models of Programming Languages," Ph.D. Thesis, Report CST-40-86, Department of Computer Science, University of Edinburgh.

Printed in Belgium
Uitgever: Academic Press, Inc.
Verantwoordelijke uitgever voor België:
Hubert Van Maele
Altenastraat 20, B-8310 Sint-Kruis