Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

E. Horita, J.W. de Bakker, J.J.M.M. Rutten

# Fully abstract denotational models for nonuniform concurrent languages

Computer Science/Department of Software Technology Report CS-R9027 June

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

# Fully Abstract Denotational Models for Nonuniform Concurrent Languages

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

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

**Abstract:** This paper investigates *full abstractness* of denotational models with respect to operational ones for two concurrent languages. The languages are *nonuniform* in the following sense: They have individual variables which store values, and the elementary actions are (mainly) value assignments to these variables. The first language $\mathcal{L}_1$ has *parallel composition* but no communication, whereas the second one $\mathcal{L}_2$ has CSP-like *communications* in addition.

For each of $\mathcal{L}_i$ $(i = 1, 2)$, an operational model $\mathcal{O}_i$ is introduced in terms of a Plotkin-style transition system, while a denotational model $\mathcal{D}_i$ for $\mathcal{L}_i$ is defined compositionally using interpreted operations of the language, with meanings of recursive programs as fixed points in appropriate complete metric spaces.

The full abstractness 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 $s_1 \| T$ and $s_2 \| T$ are distinct.

A combinatorial method for constructing such $T$ is proposed. Thereby the full abstractness of $\mathcal{D}_1$ and $\mathcal{D}_2$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ respectively is established. That is, $\mathcal{D}_i$ is most abstract of those models $\mathcal{C}$ which are compositional and satisfy $\mathcal{O}_i = \alpha \circ \mathcal{C}$ for some abstraction function $\alpha$ $(i = 1, 2)$.

**1980 Mathematics Subject Classification:** 68B10, 68C01.
**1986 Computing Reviews Categories:** D.3.1, F.3.2, F.3.3.
**Key Words & Phrases:** domain equations, metric spaces, concurrency, imperative languages, denotational semantics, operational semantics, correctness, full abstractness, linear time, branching time.

# 1 Introduction

This paper investigates *full abstractness* of denotational models with respect to operational ones for two concurrent languages. The languages are *nonuniform* in the following sense: They have individual variables which store values, and the elementary actions are (mainly) value assignments to these variables. The first language $\mathcal{L}_1$ has *parallel composition* but no communication, whereas the second one $\mathcal{L}_2$ has CSP-like *communications* in addition. Both of the two languages have recursion.

For each of $\mathcal{L}_i$ $(i = 1, 2)$, an operational model $\mathcal{O}_i$ for $\mathcal{L}_i$ is introduced in terms of a Plotkin-style transition system, while a denotational model $\mathcal{D}_i$ for $\mathcal{L}_i$ is defined compositionally using interpreted operations of the language and some fixed point method for defining the meanings of recursive programs.

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

Report CS-R9027
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

The *full abstractness problem* for programming languages was first raised by Milner in [Mil 73]. In general, a model $\mathcal{D}$ for a language $\mathcal{L}$ is called *fully abstract* with respect to another model $\mathcal{O}$, if it makes *just enough* distinctions in order to be correct (and thus compositional) with respect to $\mathcal{O}$. In other words, it is fully abstract with respect to $\mathcal{O}$, if

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

where a *context* is a statement consisting of the language constructs of $\mathcal{L}$ and a *place-holder* (or a *hole*) $\xi$, and $C[s]$ denotes the result of substituting $s$ for $\xi$ in $C$.[1] If $\mathcal{D}$ is fully abstract with respect to $\mathcal{O}$, then $\mathcal{D}$ is most abstract of those models $\mathcal{C}$ which are compositional and satisfy $\mathcal{O} = \alpha \circ \mathcal{C}$ for some abstraction function $\alpha$, i.e., for each of these $\mathcal{C}$'s, there is an *abstraction function* $\beta$ such that $\beta \circ \mathcal{C} = \mathcal{D}$.

As was shown in [BKO 88], for a language which can be formulated as the set of terms generated by a single-sorted signature and has *no recursion*, there is a unique fully abstract compositional model with respect to a given operational model.

However, as was shown in [AP 86], for a language *with recursion*, there is not necessarily a fully abstract *denotational* model with respect to a given operational model, though it is unique if it exists as for a language without recursion. Here we claim that a compositional model $\mathcal{D}$ should satisfy the following two conditions, to be called *denotational*:

**(i)** Every interpreted operation is *continuous* with respect to some topology on the semantic domain.

**(ii)** The meaning of every recursive program $X \Leftarrow b(X)$ is obtained as the limit of the iteration sequence
$$p_0, f(p_0), f^2(p_0), \cdots,$$
where $f$ is obtained as the interpretation of the body $b(X)$ and $p_0$ is some initial point.

The function $f$ is continuous by **(i)**, and therefore one has
$$f(\lim_n[f^n(p_0)]) = \lim_n[f(f^n(p_0))] = \lim_n[f^n(p_0)],$$
i.e., the meaning of $X$ is a fixed point of $f$.

In Section 2, some mathematical preliminaries on complete metric spaces, especially on spaces consisting (of sets) of streams, are given.

Section 3 and Section 4 constitute the main body of our paper.

In Section 3, the first language $\mathcal{L}_1$ is introduced; an operational model $\mathcal{O}_1$ is presented in terms of a Plotkin-style transition system; a denotational model $\mathcal{D}_1$ for $\mathcal{L}_1$ 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 $\mathcal{D}_1$ with respect to $\mathcal{O}_1$ is established, as in [Rut 89] and [BR 90], by means of the fixed point method introduced in [KR 88].

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

> Given two statements $s_1, s_2 \in \mathcal{L}_1$ 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] ...... (*)

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 finite sequence
> $$r = (\langle\sigma_1, \sigma_1'\rangle, \cdots, \langle\sigma_n, \sigma_n'\rangle),$$
> we can construct a tester $T$ and an executable sequence $\tilde{r} = (\langle\tilde{\sigma}_1, \tilde{\sigma}_1'\rangle, \cdots, \langle\tilde{\sigma}_k, \tilde{\sigma}_k'\rangle)$ with $k \geq n$

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

[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 $s$.

such that for every process $p$, the parallel composition $p \tilde{\|} \mathcal{D}_1[\![T]\!]$ can execute $\tilde{r}$ if and only if there is some sequence $q$ such that $(\langle\sigma_1,\sigma_1'\rangle,\cdots,\langle\sigma_n,\sigma_n'\rangle)\cdot q \in p$.

Intuitively, for such $T$ and $\tilde{r}$, every process $p$ is *forced* to execute the steps $\langle\sigma_1,\sigma_1'\rangle,\cdots,\langle\sigma_n,\sigma_n'\rangle$ (maybe not consecutively but in this order), when $p \tilde{\|} \mathcal{D}_1[\![T]\!]$ executes the the steps $(\langle\tilde{\sigma}_1,\tilde{\sigma}_1'\rangle,\cdots,\langle\tilde{\sigma}_k,\tilde{\sigma}_k'\rangle)$ consecutively.

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

If $s_1$ and $s_2$ are distinct in their denotational meanings, then 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 $r$ as above, one has

$\tilde{r} \in \mathcal{D}_1[\![s_1]\!] \tilde{\|} \mathcal{D}_1[\![T]\!]$ and $\tilde{r} \notin \mathcal{D}_1[\![s_2]\!] \tilde{\|} \mathcal{D}_1[\![T]\!]$.

Thus one has a difference between the operational meanings of the two statements $s_1 \| T$ and $s_2 \| T$.

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

In Section 4, the second language $\mathcal{L}_2$ is introduced; an operational model $\mathcal{O}_2$ for $\mathcal{L}_2$ is given as in Section 3.

The domain of a denotational model $\mathcal{D}_2$ for $\mathcal{L}_2$ is a kind of *failures model*, which was introduced in [BHR 84], and is adapted here to the nonuniform setting. Each element of the domain is a set consisting of such elements that are represented as

$\langle((\langle s_i,a_i,s_i'\rangle)_i,\langle s'',C\rangle\rangle$,

where $s_i$, $s_i'$, and $s''$ are states, $a_i$ is an action, and $C$ is a set of *communication sorts*. These elements are called *failures*; the parts $(\langle s_i,a_i,s_i'\rangle)_i$ and $\langle s'',C\rangle$ are called a *trace* and a *refusal*, respectively.

First, the correctness of $\mathcal{D}_2$ is established as in Section 3. Then, the full abstractness of $\mathcal{D}_2$ is established by a combination of the testing method and the method proposed by Bergstra, Klop, and Olderog in [BKO 88] to establish the full abstractness of a *failures model* for a uniform language without recursion. This method was adapted by Rutten in [Rut 89] so as 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.

Given two statements $s_1$ and $s_2$ of $\mathcal{L}_2$ 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 [BKO 88].

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

Some mathematical proofs are given in the appendices.

# 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*, and *closed set* to be known (the reader might consult [Dug 66] or [Eng 77]).

Let $\langle M_1,d_1\rangle$ and $\langle M_2,d_2\rangle$ be two metric spaces. The function $f: M_1 \to M_2$ is called *contracting* (or a *contraction*), if

$$\exists\varepsilon \in [0,1)[\forall x,y \in M_1[d_2(f(x),f(y)) \le \varepsilon\cdot d_1(x,y)]].$$

We call $M_1$ and $M_2$ *isometric* (notation: $M_1 \cong M_2$) if there exists a bijective mapping $f: M_1 \to M_2$ such that

$$\forall x,y \in M_1[d_2(f(x),f(y)) = d_1(x,y)].$$

The following fact is known as Banach's Theorem: Let $\langle M, d\rangle$ be a complete metric space and $f : M \to M$ be a contraction. Then $f$ has a unique fixed point, that is, there exists a unique $x \in M$ such that $f(x) = x$. For a contraction $f$, the unique fixed point $f$ is denoted by $\mathrm{fix}(f)$.

The following notations are used.

**Notation 1**

**(1)** The usual $\lambda$-notation is used for denoting functions: 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 $E(x)$.

**(2)** For a set $X$, the cardinality of $X$ is denoted by $\sharp(X)$, and the set of nonempty subsets of $X$ is denoted by $\wp_+(X)$. 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 $\omega$. For $n \in \omega$, $\bar{n} = \{m \in \omega : 1 \leq m \leq n\}$.

**(3)** The empty sequence is denoted by $\epsilon$. $\blacksquare$

**Notation 2**

**(1)** For a set $A$, the set of finite sequences of elements of $A$ is denoted by $A^{<\omega}$, and the set of nonempty finite sequences of elements of $A$ is denoted by $A^+$. The set of finite or infinite (with length $\omega$) of sequences of elements of $A$ is denoted by $A^{\leq\omega}$.

For $a \in A$, the sequence $(a)$ consisting only of $a$ is sometimes denoted by $a$.

**(2)** For a sequence $q \in A^{\leq\omega}$, $\mathrm{lgt}(q)$ denotes the length of $q$.

**(3)** For $n \in \omega$ and a sequence $q \in A^{\leq\omega}$, the *truncation* of $q$ at level $n$, written as $q^{[n]}$, is the prefix of $q$ with length $n$ if $\mathrm{lgt}(n) \geq n$, or $q$ otherwise. For a set of sequences $p \subseteq A^{\leq\omega}$, let $p^{[n]} = \{q^{[n]} : q \in p\}$. $\blacksquare$

We use the following operations 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 $\langle M, d\rangle$, $\langle M_1, d_1\rangle, \cdots, \langle M_n, d_n\rangle$ be metric spaces.

**(1)** With $M_1 \uplus \cdots \uplus M_n$ we denote the *disjoint union* of $M_1, \cdots, M_n$, which can be defined as $\bigcup_{j \in \bar{n}}[\{j\} \times M_j]$.

We define a metric $d_{\mathrm{U}}$ on $M_1 \uplus \cdots \uplus M_n$ as follows. For every $x, y \in M_1 \uplus \cdots \uplus M_n$,

$$d_{\mathrm{U}}(x,y) = \begin{cases} d_j(x,y) & \text{if } \exists j \in \bar{n}[x, y \in \{j\} \times M_j], \\ 1 & \text{otherwise.} \end{cases}$$

**(2)** We define a metric $d_{\mathrm{P}}$ on the Cartesian product $M_1 \times \cdots \times M_n$ as follows.
For $(x_1, \cdots, x_n), (y_1, \cdots, y_n) \in M_1 \times \cdots \times M_n$

$$d_{\mathrm{P}}((x_1, \cdots, x_n), (y_1, \cdots, y_n)) = \max_{j \in \bar{n}}[d_j(x_j, y_j)].$$

**(3)** For $X \subseteq M$, the *closure* of $X$ is denoted by $X^{\mathrm{cls}}$.

**(4)** Let $\wp_{\mathrm{cl}}(M) = \{X \in \wp(M) : X \text{ is closed}\}$. We define a metric $d_{\mathrm{H}}$ on $\wp_{\mathrm{cl}}(M)$, called the *Hausdorff distance*, as follows.
For every $X, Y \in \wp_{\mathrm{cl}}(M)$,

$$d_{\mathrm{H}}(X,Y) = \max\{\sup_{x \in X}[\underline{d}(x, Y)], \sup_{y \in Y}[\underline{d}(y, X)]\},$$

where $\underline{d}(x,Z) = \inf_{z\in Z}[d(x,z)]$ for every $Z \subseteq M$, $x \in X$. (We use the convention that $\sup\emptyset = 0$ and $\inf\emptyset = 1$.)

The space $\wp_{nc}(M) = \{X \in \wp(M) : X \text{ is closed and nonempty}\}$ is supplied with a metric by taking the restriction of $d_H$ to it.

**(5)** For a real number $\varepsilon \in [0,1)$, we define
$id_\varepsilon(\langle M,d\rangle) = \langle M,d'\rangle$,
where $d'(x,y) = \varepsilon \cdot d(x,y)$, for every $x,y \in M$.

**(6)** An arbitrary set $A$ can be supplied with a metric $d_A$, called the *discrete metric*, defined by

$$d_A(x,y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y. \end{cases}$$

The space $\langle A, d_A\rangle$ is an ultra-metric space. ∎

Complete metric spaces consisting of streams are introduced as solutions of appropriate domain equations as in [BZ 82] and [AR 89]. The metric on spaces of streams turns out identical with the one defined in terms of projections, which are introduced below.

**Definition 2** *(Projection Functions on Domain of Streams)*
Let $\langle Q, d_Q\rangle$ be the unique complete metric space which satisfies the following domain equation:

$$\mathbf{Q} \cong B \uplus (A \times \mathrm{id}_{1/2}(\mathbf{Q})).$$

The existence and uniqueness of such a $\mathbf{Q}$ has been shown in [BZ 82] and [AR 89], respectively. Remark that $\mathrm{id}_{1/2}$ is necessary for the associated functor with this domain equation to be contractive, which condition ensures the uniqueness of the solution (see [AR 89]).

Actually the space $\langle (A^{<\omega} \times B) \uplus A^\omega, d_Q\rangle$ is the solution of this equation, where $d_Q$ is defined, in terms of truncation, by

$$d_Q(q_1,q_2) = \begin{cases} (1/2)^{\min\{n:(q_1)^{[n]} \neq (q_2)^{[n]}\}-1} & \text{if } \exists n[(q_1)^{[n]} \neq (q_2)^{[n]}], \\ 0 & \text{otherwise.} \end{cases}$$

Here and in the sequel, we treat an element $\langle r,b\rangle$ of $A^{<\omega} \times B$ as a sequence with length $\mathrm{lgt}(r)+1$ instead of a pair of $r$ and $b$.

**(1)** For $n \in \omega$, the $n$-th projection function $\pi_n : \mathbf{Q} \to \mathbf{Q}$ is defined as follows. First, fix an arbitrary element $b_0$ of $B$. For $q \in \mathbf{Q}$,

**(i)** $\pi_0(q) = b_0$,

**(ii)** $\pi_{n+1}(q) = \begin{cases} b & \text{if } q \in B, \\ a \cdot \pi_n(q') & \text{if } q = a \cdot q'. \end{cases}$

**(2)** Let $\mathbf{P} = \wp_{nc}(\mathbf{Q})$. For $n \in \omega$ and $p \in \mathbf{P}$, let
$\tilde{\pi}_n(p) = \{\pi_n(q) : q \in p\}$. ∎

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

The metric $d_Q$ is formulated also in terms of projection as follows:

**Lemma 1**

**(1)** *For* $q_1, q_2 \in \mathbf{Q}$,

$$d_Q(q_1,q_2) = \begin{cases} (1/2)^{\min\{n:\pi_n(q_1) \neq \pi_n(q_2)\}-1} & \textit{if } \exists n[\pi_n(q_1) \neq \pi_n(q_2)], \\ 0 & \textit{otherwise.} \end{cases}$$

**(2)** *For* $p_1, p_2 \in \mathbf{P}$,

$$d_P(p_1,p_2) = \begin{cases} (1/2)^{\min\{n:\tilde{\pi}_n(p_1)\neq\tilde{\pi}_n(p_2)\}-1} & \text{if } \exists n[\tilde{\pi}_n(p_1) \neq \tilde{\pi}_n(p_2)], \\ 0 & \text{otherwise.} \end{cases}$$

**(3)** $\forall n \in \omega, \exists \varepsilon > 0$
$[\forall p_1, p_2 \in \mathbf{P}[d_P(p_1,p_2) \leq \varepsilon \Rightarrow \tilde{\pi}_n(p_1) = \tilde{\pi}_n(p_2)]\ ]$. ∎

**Proof.** See Appendix A.1. ∎

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 exist $n \in \omega$ and $\mathbf{P}'' \subseteq \mathbf{P}$ such that

$$\forall p \in \mathbf{P}[p \in \mathbf{P}' \Leftrightarrow \tilde{\pi}_n(p) \in \mathbf{P}'']. \quad ∎ \tag{1}$$

A property defined for elements of $\mathbf{P}$ is called *finitely characterized*, if the set consisting of those elements of $\mathbf{P}$ which have the property is finitely characterized. The next example presents such a property.

**Example 1** Fix $n \in \omega$. An element $p \in \mathbf{P}$ said to be *nonempty at level* $n$, if $p^{[n]} \cap A^n \neq \emptyset$. Let $\mathbf{P}' = \{p \in \mathbf{P} : p \text{ is nonempty at level } n\}$. Then it is immediate that
$\forall p \in \mathbf{P}[p \in \mathbf{P}' \Leftrightarrow \pi_{n+1}(p) \in \mathbf{P}']$.
Thus $\mathbf{P}'$ is finitely characterized, and therefore, the property "being *nonempty at level* $n$" is finitely characterized. Remark that $\mathbf{P}''$ in (1) 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. We shall use this lemma in the proof of full abstractness to show that the denotational meaning of each statement is a member of a desired set.

**Lemma 2**

**(1)** *If* $\mathbf{P}' \subseteq \mathbf{P}$ *is finitely characterized, then* $\mathbf{P}'$ *is closed in* $\mathbf{P}$.

**(2)** *If* $\mathcal{P} \subseteq \wp(\mathbf{P})$ *and* $\forall \mathbf{P}' \in \mathcal{P}[\mathbf{P}'$ *is finitely characterized*],
*then* $\bigcap \mathcal{P}$ *is closed in* $\mathbf{P}$. ∎

**Proof.** **(1)** Suppose $(p_i)_{i\in\omega}$ converges and $\forall i \in \omega[p_i \in \mathbf{P}']$.
Let $p = \lim_{i\in\omega}[p_i]$. Since $\mathbf{P}'$ is finitely characterized, there exist $n \in \omega$ and $\mathbf{P}'' \subseteq \mathbf{P}$ such that

$$\forall p \in \mathbf{P}[p \in \mathbf{P}' \Leftrightarrow \tilde{\pi}_n(p) \in \mathbf{P}'']. \tag{2}$$

Fix such $n$ and $\mathbf{P}''$. By Lemma 1 (3), there exists $\varepsilon > 0$ such that

$$\forall p_1, p_2 \in \mathbf{P}[d_P(p_1,p_2) \leq \varepsilon \Rightarrow \tilde{\pi}_n(p_1) = \tilde{\pi}_n(p_2)]. \tag{3}$$

Fix such $\varepsilon$.
$\exists N \in \omega[d_P(p_N, p) \leq \varepsilon]$.
Fix such $N$. By (3),

$$\tilde{\pi}_n(p_N) = \tilde{\pi}_n(p). \tag{4}$$

By (2) and the fact that $p_N \in \mathbf{P}'$, one has
$\tilde{\pi}_n(p_N) \in \mathbf{P}'$.
By this and (4), one has
$\tilde{\pi}_n(p) \in \mathbf{P}'$.
It follows from this and (2) that
$p \in \mathbf{P}'$.
**(2)** This part follows immediately from **(1)** and the fact that the intersection of closed sets is closed. ∎

# 3 A Nonuniform Language with Parallel Composition

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

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

Then a denotational model $\mathcal{D}_1$ 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 $\mathcal{D}_1$ with respect to $\mathcal{O}_1$ is established, as in [Rut 89] and [BR 90], by means of the fixed point method introduced in [KR 88].

Finally, full abstractness of $\mathcal{D}_1$ 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 abstractness of $\mathcal{D}_1$ with respect to $\mathcal{O}_1$ is established.

## 3.1 The Language $\mathcal{L}_1$

The language $\mathcal{L}_1$ 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.

Remark that sequential composition as in [BKO 88] is not included in this language: We use *prefixing* of assignment statements as in [Mil 80], 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 $\mathcal{L}_1$, but has general sequential composition instead of prefixing.

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

**Notation 3**

**(1)** Let $(v \in)\mathbf{V}$ denote some abstract domain of values.

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

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

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

**(5)** Let $(b \in)\mathrm{BExp}$ 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 $\sigma$ are denoted by $[\![e]\!](\sigma)$ and $[\![b]\!](\sigma)$, respectively. It is also assumed that IVar is finite; the full abstractness of a denotational model is established under this assumption.

Let $X$ ranges over RVar, the set of recursion variables, and let $\xi$ range over SVar, the set of statement variables. Remark 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 $\mathcal{L}_1$ is introduced as a subset of $\mathcal{L}_1^*$, a language with place holders.

**Definition 4** *(Language $\mathcal{L}_1$)*

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

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

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

Let $\mathrm{FV}(S)$ denote the set of statement variables contained in $S$.

**(2)** Let $(s \in)\mathcal{L}_1$ be the set of statements with no statement variable. That is, $\mathcal{L}_1 = \{S \in \mathcal{L}_1^* : \mathrm{FV}(S) = \emptyset\}$.

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

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

**(4)** We assume that each recursion variable $X$ is associated with an element $g_X$ of $\mathcal{G}_1$ 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 $D = \{\langle X, g_X\rangle\}_{X \in \mathrm{RVar}}$.

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

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

**Notation 4** Let $\mathcal{L}^*$ be a language formulated as the set of terms generated by a signature $\mathcal{S}$ and a variable set $\{\xi_i\}$.

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

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

**(3)** Let $\mathcal{I}$ be an *interpretation*, i.e., a set of interpreted operations for the signature $\mathcal{S}$ with an underlying domain $\mathbf{P}$ (see bib:rutt:90 for a formal definition of an interpretation for a signature); let $S_{(\xi_1,\cdots,\xi_n)}$ be a context.

For $p_1,\cdots,p_n \in \mathbf{P}$, the notation $[\![S]\!]^{\mathcal{I}}[(\xi_1,\cdots,\xi_n)/(p_1,\cdots,p_n)]$ denotes the interpretation of $S$ under $\mathcal{I}$ with the assignment of the value $p_i$ to the variable $\xi_i$, $i \in \bar{n}$. More simply, we sometimes write $[\![S_{(\xi_1,\cdots,\xi_n)}]\!]^{\mathcal{I}}(p_1,\cdots,p_n)$ for $[\![S]\!]^{\mathcal{I}}[(p_1,\cdots,p_n)/(\xi_1,\cdots,\xi_n)]$. ∎

## 3.2 Operational Model $\mathcal{O}_1$ for $\mathcal{L}_1$

The operational model $\mathcal{O}_1$ rests on a transition system $\rightarrow_1$ of the style of [Plo 81]. The transition relation $\rightarrow_1 \subseteq (\mathcal{L}_1 \times \Sigma) \times (\mathcal{L}_1 \times \Sigma)$ is defined as follows. For $s_1, s_2 \in \mathcal{L}_1$ and $\sigma_1, \sigma_2 \in \Sigma$, $(\langle s_1,\sigma_1\rangle, \langle s_2,\sigma_2\rangle) \in \rightarrow_1$ is written as $\langle s_1,\sigma_1\rangle \longrightarrow_1 \langle s_2,\sigma_2\rangle$ 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 (5). For $\sigma \in \Sigma$, $x \in \mathrm{IVar}$, and $v \in \mathbf{V}$, the notation $\sigma[v/x]$ is used to denote a state $\sigma'$ which is the same as $\sigma$ except that $\sigma'(x) = v$.

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

(1) $\langle (x := e); s, \sigma \rangle \longrightarrow_1 \langle s, \sigma[\llbracket e \rrbracket(\sigma)/x] \rangle$.

(2) $$\frac{\langle s_1, \sigma \rangle \longrightarrow_1 \langle s, \sigma' \rangle}{\langle \mathrm{If}(b, s_1, s_2), \sigma \rangle \longrightarrow_1 \langle s, \sigma' \rangle} \quad \text{If } \llbracket e \rrbracket(\sigma) = \mathrm{tt}$$

$$\frac{\langle s_2, \sigma \rangle \longrightarrow_1 \langle s, \sigma' \rangle}{\langle \mathrm{If}(b, s_1, s_2), \sigma \rangle \longrightarrow_1 \langle s, \sigma' \rangle} \quad \text{If } \llbracket e \rrbracket(\sigma) = \mathrm{ff}$$

(3) $$\frac{\langle s_1, \sigma \rangle \longrightarrow_1 \langle s, \sigma' \rangle}{\begin{array}{l}\langle s_1 + s_2, \sigma \rangle \longrightarrow_1 \langle s, \sigma' \rangle \\ \langle s_2 + s_1, \sigma \rangle \longrightarrow_1 \langle s, \sigma' \rangle\end{array}}$$

(4) $$\frac{\langle s_1, \sigma \rangle \longrightarrow_1 \langle s, \sigma' \rangle}{\begin{array}{l}\langle s_1 \parallel s_2, \sigma \rangle \longrightarrow_1 \langle s \parallel s_2, \sigma' \rangle \\ \langle s_2 \parallel s_1, \sigma \rangle \longrightarrow_1 \langle s_2 \parallel s, \sigma' \rangle\end{array}}$$

(5) 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:

$$\frac{\langle g_X, \sigma \rangle \longrightarrow_1 \langle s', \sigma' \rangle}{\langle X, \sigma \rangle \longrightarrow_1 \langle s', \sigma' \rangle} \quad \blacksquare$$

The transition relation $\rightarrow_1$ is *finitely branching* in the following sense.

**Lemma 3**
$\forall s \in \mathcal{L}_1, \forall \sigma \in \Sigma[\ \{\langle s', \sigma' \rangle \in \mathcal{L}_1 \times \Sigma : \langle s, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle\}$ *is finite* $]$. ∎

**Proof.** A statement $s$ is said to be *finitely branching*, if

$$\forall \sigma \in \Sigma[\ \{\langle s', \sigma' \rangle \in \mathcal{L}_1 \times \Sigma : \langle s, \sigma \rangle \rightarrow_1 \langle s', \sigma' \rangle\} \text{ is finite }].$$

First, it is shown

$$\forall g \in \mathcal{G}_1[\ g \text{ is finitely branching }] \qquad (5)$$

by induction on the structure $g$ using the fact that

$$\forall x \in \mathrm{IVar}, \forall e \in \mathrm{VExp}, \forall s \in \mathcal{L}_1[\ \text{“}(x := e); s\text{” is finitely branching }],$$

and

$$\begin{array}{l}\forall s_1, s_2 \in \mathcal{L}_1[\ \text{if both } s_1 \text{ and } s_2 \text{ are finitely branching,} \\ \text{then “}s_1 + s_2\text{” and “}s_1 \parallel s_2\text{” are also finitely branching }].\end{array} \qquad (6)$$

It follows from (5) and (6) that

$$\forall s \in \mathcal{L}_1[\ s \text{ is finitely branching }]$$

by induction on the structure of $s$. ∎

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 $\mathcal{L}_1$)*

(1) Let $\mathbf{M}_1^{\mathcal{O}} = (\mathcal{L}_1 \rightarrow (\Sigma \rightarrow \wp_{\mathrm{nc}}(\Sigma^{\leq \omega})))$, and let $\Psi_1^{\mathcal{O}} : \mathbf{M}_1^{\mathcal{O}} \rightarrow \mathbf{M}_1^{\mathcal{O}}$ be as follows:

For $f \in \mathbf{M}_1^{\mathcal{O}}$, $s \in \mathcal{L}_1$, and $\sigma \in \Sigma$,

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

The right-hand side of the above equation is closed by Lemma 3, and therefore, $\Psi_1^{\mathcal{O}}$ 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

$$\mathcal{O}_1 : \mathcal{L}_1 \rightarrow (\Sigma \rightarrow \wp_{\mathrm{nc}}(\Sigma^{\leq\omega})),$$

and for each $s \in \mathcal{L}_1$ and $\sigma \in \Sigma$,

$$\mathcal{O}_1[\![s]\!](\sigma) = \begin{cases} \bigcup\{\sigma' \cdot \mathcal{O}_1[\![s']\!](\sigma') : \langle s, \sigma\rangle \rightarrow_1 \langle s', \sigma'\rangle\} & \text{if } \exists\langle s', \sigma'\rangle[\langle s, \sigma\rangle \rightarrow_1 \langle s', \sigma'\rangle], \\ \{\epsilon\} & \text{otherwise.} \end{cases}$$ ∎

Remark that $\mathcal{O}_2$ is not compositional as the following example shows.

**Example 2** Let $x \in \mathrm{IVar}$. Then

$\mathcal{O}_1[\![(x := 0); (x := x + 1); \mathrm{Stop}]\!] = \mathcal{O}_1[\![(x := 0); (x := 1); \mathrm{Stop}]\!] = (\lambda\sigma : \{(\sigma[0/x], \sigma[1/x])\})$,

but

$\mathcal{O}_1[\![((x := 0); (x := x + 1); \mathrm{Stop}) \parallel ((x := 2); \mathrm{Stop})]\!]$
$\neq \mathcal{O}_1[\![((x := 0); (x := 1); \mathrm{Stop}) \parallel ((x := 2); \mathrm{Stop})]\!]$. ∎

## 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
$\mathbf{Q}_1 \cong \Sigma \uplus ((\Sigma \times \Sigma) \times \mathrm{id}_{1/2}(\mathbf{Q}_1))$.
One has

$$\mathbf{Q}_1 \cong ((\Sigma \times \Sigma)^{<\omega} \times \Sigma) \cup (\Sigma \times \Sigma)^{\omega}.$$

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

**(3)** The *initial state* of a sequence $q \in \mathbf{Q}_1 \cup (\Sigma \times \Sigma)^{+}$, written as $\mathrm{istate}_1(q)$, is given by

$$\mathrm{istate}_1(q) = \begin{cases} \sigma & \text{if } q = (\langle\sigma, \sigma'\rangle) \cdot q', \\ \sigma'' & \text{if } q = (\sigma''). \end{cases}$$

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

**(5)** Let $p \in \wp_{\mathrm{nc}}(\mathbf{Q}_1)$, 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 level $n$ for every $n \in \omega$.

**(6)** The set $\mathbf{P}_1$, the domain of processes for $\mathcal{L}_1$, is given by
$\mathbf{P}_1 = \{p \in \wp_{\text{nc}}(\mathbf{Q}_1) : p \text{ is uniformly nonempty}\}$. ∎

**Remark 1** A subset $\mathbf{P}$ of $\wp_{\text{nc}}(\mathbf{Q}_1)$ is said to be *closed under taking remainders* if

$$\forall p \in \mathbf{P}, \forall r [p[r] \neq \emptyset \Rightarrow p[r] \in \mathbf{P}].$$

It is immediate that $\mathbf{P}_1$ is the smallest set of those subsets of $\wp_{\text{nc}}(\mathbf{Q}_1)$ which are closed under taking remainders and include $\{p \in \wp_{\text{nc}}(\mathbf{Q}_1) : p \text{ is uniformly nonempty at level } 0\}$. ∎

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 $\wp_{\text{nc}}(\mathbf{Q}_1)$, and therefore $\mathbf{P}_1$ is a complete metric space with the original metric of $\wp_{\text{nc}}(\mathbf{Q}_1)$ restricted to it.* ∎

**Proof.** For $n \in \omega$, let

$$P_n = \{p \in \wp_{\text{nc}}(\mathbf{Q}_1) : p \text{ is uniformly nonempty at level } n\}$$

It is immediate that

$$\forall p \in \wp_{\text{nc}}(\mathbf{Q}_1)[p \in P_n \Leftrightarrow \tilde{\pi}_{n+1}(p) \in P_n].$$

Thus $P_n$ is finitely characterized.
Remark that $\mathbf{P}_1 = \bigcap_{n\in\omega}[P_n]$. Hence, by Lemma 3, $\mathbf{P}_1$ is closed. ∎

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

**Definition 8** *(Interpretation $\mathcal{I}_1$ for Signature of $\mathcal{L}_1$)*

**(1)** $\text{stop}_1 = \{(\sigma) : \sigma \in \Sigma\}$.

**(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 operation "$(x := e); \cdot$ " on statements, is defined as follows:

For every $p \in \mathbf{P}_1$,
$\text{asg}_1(x, e)(p) = \{\langle \sigma, \sigma[\llbracket e \rrbracket(\sigma)/x]\rangle \cdot p : \sigma \in \Sigma\}$,

where $(\langle \sigma, \sigma[\llbracket e \rrbracket(\sigma)/x]\rangle) \cdot p$ denotes the concatenation of $(\langle \sigma, \sigma[\llbracket e \rrbracket(\sigma)/x]\rangle)$ 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 operation "$\text{If}(b, \cdot, \cdot)$" on statements, is defined as follows:

For every $\forall p_1, p_2 \in \mathbf{P}_1$, $\text{if}(b)(p_1, p_2) = \bigcup_{\sigma\in\Sigma}[\text{if}(\llbracket b \rrbracket(\sigma) = \text{tt}, p_1\langle\sigma\rangle, p_2\langle\sigma\rangle)]$.

**(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 $p \cap \text{stop}_1$ is called the *deadlock 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 deadlock part is the intersection of the deadlock 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 deadlock 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 $p_1, p_2 \in \mathbf{P}_1$,

$$p_1 \mathbin{\tilde{\|}} p_2 = (p_1 \mathbin{\|\!\!\_} p_2) \cup (p_2 \mathbin{\|\!\!\_} p_1) \cup (p_1 \# p_2),$$
$$p_1 \mathbin{\|\!\!\_} p_2 = \bigcup\{\langle\sigma,\sigma'\rangle \cdot (p_1[\langle\sigma,\sigma'\rangle] \mathbin{\tilde{\|}} p_2) : p_1[\langle\sigma,\sigma'\rangle] \neq \emptyset\}. \tag{7}$$

Formally the operation $\tilde{\|}$ is defined as the fixed point of a suitably defined contraction: Let $\mathbf{M}_1^{\|} = (\mathbf{P}_1 \times \mathbf{P}_1 \to \mathbf{P}_1)$, and let $\Omega_1^{\|}w : \mathbf{M}_1^{\|} \to \mathbf{M}_1^{\|}$ 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

$$\Omega_1^{\|\!\_}(F)(p_1,p_2) = \bigcup\{\langle\sigma,\sigma'\rangle \cdot F(p_1[\langle\sigma,\sigma'\rangle],p_2) : p_1[\langle\sigma,\sigma'\rangle] \neq \emptyset\}.$$

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 $\neg\exists\sigma'[\Omega_1^{\|}(F)(p_1,p_2)[\langle\sigma,\sigma'\rangle] \neq \emptyset]$. Then, by the definition of $\Omega_1^{\|}$, one has $\neg\exists\sigma'[p_1[\langle\sigma,\sigma'\rangle] \neq \emptyset]$ and $\neg\exists\sigma'[p_2[\langle\sigma,\sigma'\rangle] \neq \emptyset]$. Thus, by the fact that $p_1$ and $p_2$ are uniformly nonempty at level 0, one has $\langle\sigma\rangle \in (p_1 \# p_2)$.

Moreover $\Omega_1^{\|}(F)(p_1,p_2)$ is uniformly nonempty at level $n \geq 1$, 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 $\tilde{\|} = \mathrm{fix}(\Omega_1^{\|})$, and $\|\!\_ = \Omega_1^{\|\!\_}(\tilde{\|})$.

**(6)** Let

$$\mathcal{I}_1 = \{\mathrm{stop}_1, \{\mathrm{asg}_1(x,e) : \langle x,e\rangle \in \mathrm{IVar} \times \mathrm{VExp}\}, \{\mathrm{if}(b) : b \in \mathrm{BExp}\}, \tilde{+}, \tilde{\|}\}. \quad \blacksquare$$

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

**Lemma 5**

**(1)** $\langle\sigma,\sigma'\rangle \cdot q \in p_1 \mathbin{\tilde{\|}} p_2 \Rightarrow (q \in (p_1[\langle\sigma,\sigma'\rangle] \mathbin{\tilde{\|}} p_2)) \vee (q \in (p_1 \mathbin{\tilde{\|}} p_2[\langle\sigma,\sigma'\rangle]))$.

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

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

**Definition 9** *(Denotational Model $\mathcal{D}_1$ for $\mathcal{L}_1$)*
The model $\mathcal{D}_1 : \mathcal{L}_1 \to \mathbf{P}_1$ is defined by induction on the structure of $s \in \mathcal{L}_2$.

**(1)** First, for each recursion variable $X$, $\mathcal{D}_1[\![X]\!]$ 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 $\mathbf{M}_1^{\mathcal{D}} = (\mathrm{RVar} \to \mathbf{P}_1)$, and let $\Pi_1 : \mathbf{M}_1^{\mathcal{D}} \to \mathbf{M}_1^{\mathcal{D}}$ be defined as follows:

For $\mathbf{p} \in \mathbf{M}_1^{\mathcal{D}}$, $X \in \mathrm{RVar}$,

$$\Pi_1(\mathbf{p})(X) = [\![g_X]\!]^{\mathcal{I}_1}[(\mathbf{p}(Y_1^X),\cdots,\mathbf{p}(Y_{l(X)}^X))/(Y_1^X,\cdots,Y_{l(X)}^X)],$$

where $\{Y_1^X,\cdots,Y_{l(X)}^X\}$ is the set of recursion variables contained in $g_X$. (See Notation 4 for the notation $[\![g_X]\!]^{\mathcal{I}_1}(\cdots)$.)

The mapping $\Pi_1$ is a contraction from $\mathbf{M}_1^{\mathcal{D}}$ to $\mathbf{M}_1^{\mathcal{D}}$. Let $\mathbf{p}_0 = \mathrm{fix}(\Pi_1)$. For $X \in \mathrm{RVar}$, let us define $X^{\mathcal{D}_1}$, the denotational meaning of $X$, by:
$X^{\mathcal{D}_1} = \mathbf{p}_0(X)$.

**(2)** Next, $\mathcal{D}_1 : \mathcal{L}_1 \to \mathbf{P}_1$ is defined by induction on the structure of $s \in \mathcal{L}_1$ as follows:

For each operation $F$ of $\mathcal{L}_1$ with arity $r$, and $s_1, \cdots, s_r \in \mathcal{L}_1$, let
$\mathcal{D}_1[\![F(s_1, \cdots, s_r)]\!] = F^{\mathcal{I}_1}(\mathcal{D}_1[\![s_1]\!], \cdots, \mathcal{D}_1[\![s_r]\!])$,
where $F^{\mathcal{I}_1}$ is the interpreted operation corresponding to $F$. ∎

A property so-called *image finiteness* for elements of $\mathbf{P}_1$ is introduced. It is shown that the denotational meaning of each statement in $\mathcal{L}_1$ has this property; this fact is used for establishing the full abstractness of $\mathcal{D}_1$.

**Definition 10** *(Image Finiteness for Elements of* $\mathbf{P}_1$*)*

**(1)** Let $p \in \mathbf{P}_1$ and $n \in \omega$.

The process $p$ is *image finite at level n*, notation: $\mathrm{IFin}_1^{(n)}(p)$ iff

$$\forall r \in (\Sigma \times \Sigma)^n, \forall \sigma[\ \{\sigma' \in \Sigma : r \cdot \langle \sigma, \sigma' \rangle \in p^{[n+1]}\} \text{ is finite } ].$$

The process $p$ is *image finite*, notation: $\mathrm{IFin}_1(p)$ iff

$$\forall n \in \omega[\mathrm{IFin}_1^{(n)}(p)].$$

**(2)** $\mathbf{P}_1^* = \{p \in \mathbf{P}_1 : \mathrm{IFin}_1(p)\}$. ∎

**Remark 2** It is immediate that $\{p \in \mathbf{P}_1 : \mathrm{IFin}_1(p)\}$ is the smallest set of those subsets of $\mathbf{P}_1$ which are closed under taking remainders and include $\{p \in \mathbf{P}_1 : \mathrm{IFin}_1^{(0)}(p)\}$. ∎

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

**Lemma 6**

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

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

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

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

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

**Proof.** **(1)** This part is shown as Lemma 4.
**(2)** This part follows immediately from the definition of $\mathbf{P}_1^*$.
**(3)** This part follows immediately from the definitions of the operations in Definition 8.
**(4)** Let $\mathbf{stop} = (\lambda X \in \mathrm{RVar} : \mathrm{stop}_1)$. By the definition of $\mathrm{stop}_1$, one has $\mathbf{stop} \in (\mathrm{RVar} \to \mathbf{P}_1^*)$.
By **(3)**, for every $n \in \omega$, one has

$$(\Pi_1)^n(\mathbf{stop}) \in (\mathrm{RVar} \to \mathbf{P}_1^*),$$

where $(\Pi_1)^n$ is the $n$-times iteration of $\Pi_1$.
The set $\mathbf{P}_1^*$ is a complete metric space by **(1)** , and therefore, by Banach's Theorem one has

$$\mathbf{p}_0 = \lim_n[(\Pi_1)^n(\mathbf{stop})] \in (\mathrm{RVar} \to \mathbf{P}_1^*).$$

Hence, for each $X \in \mathrm{RVar}$, one has

$$\mathcal{D}_1[\![X]\!] = \mathbf{p}_0(X) \in \mathbf{P}_1^*.$$

From this and **(3)**, it follows that

$$\forall s \in \mathcal{L}_1[\mathcal{D}_1[\![s]\!] \in \mathbf{P}_1^*].$$

**(5)** This part follows immediately from **(2)** and **(4)**. ∎

## 3.4 Correctness of $\mathcal{D}_1$ with respect to $\mathcal{O}_1$

The correctness of the denotational model is shown as in [Rut 89]: For the denotational model $\mathcal{D}_1$, an alternative formulation, called an *intermediate model*, is given, in terms of the same transition system which was used for the definition of $\mathcal{O}_1$. Let $\tilde{\mathcal{O}}_1$ be the intermediate model. Then the correctness is proved by showing that, for an appropriate abstraction function $\alpha_1$, both $\alpha_1 \circ \tilde{\mathcal{O}}_1$ and $\mathcal{O}_1$ are a fixed point of the same contraction, which by Banach's Theorem has a unique fixed point.

### 3.4.1 Intermediate Model for $\mathcal{L}_1$ and Semantic Equivalence

First, the intermediate model $\tilde{\mathcal{O}}_1$, which is an alternative formulation of $\mathcal{D}_1$, is defined in terms of the transition $\longrightarrow_1$.

**Definition 11** *(Intermediate Model $\tilde{\mathcal{O}}_1$ for $\mathcal{L}_1$)*

**(1)** Let $\mathbf{M}_1 = (\mathcal{L}_1 \to \mathbf{P}_1)$, and let $\Psi_1 : \mathbf{M}_1 \to \mathbf{M}_1$ be defined as follows:

For $F \in \mathbf{M}_1$, $s \in \mathcal{L}_1$,

$$\begin{aligned}\Psi_1(F)(s) = \ & \bigcup\{(\langle\sigma,\sigma'\rangle)\cdot F(s') : \sigma \in \Sigma \wedge \langle s,\sigma\rangle \longrightarrow_1 \langle s',\sigma'\rangle\} \\ & \cup\{(\sigma) : \sigma \in \Sigma \wedge \neg\exists\langle s',\sigma'\rangle[\langle s,\sigma\rangle \longrightarrow_1 \langle s',\sigma'\rangle]\}.\end{aligned}$$

The right-hand side of the above equation is closed by Lemma 3;
$\Psi_1$ is a contraction 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

$$\begin{aligned}\tilde{\mathcal{O}}_1[\![s]\!] = \ & \bigcup\{(\langle\sigma,\sigma'\rangle)\cdot \tilde{\mathcal{O}}_1[\![s']\!] : \sigma \in \Sigma \wedge \langle s,\sigma\rangle \longrightarrow_1 \langle s',\sigma'\rangle\} \\ & \cup\{(\sigma) : \sigma \in \Sigma \wedge \neg\exists\langle s',\sigma\rangle[\langle s,\sigma\rangle \longrightarrow_1 \langle s',\sigma'\rangle]\}. \ \blacksquare\end{aligned}$$

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 operation of $\mathcal{L}_1$ with arity $r$, and let $s_1, \cdots, s_r \in \mathcal{L}_1$. Then it holds that*
$\tilde{\mathcal{O}}_1[\![F(s_1,\cdots,s_r)]\!] = F^{\mathcal{I}_1}(\tilde{\mathcal{O}}_1[\![s_1]\!],\cdots,\tilde{\mathcal{O}}_1[\![s_r]\!])$.

**(2)** *For $s \in \mathcal{L}_1$, it holds that*
$\tilde{\mathcal{O}}_1[\![s]\!] = \mathcal{D}_1[\![s]\!]$. $\blacksquare$

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

**Lemma 8** *(Distributivity of $\tilde{\|}$ in $\mathbf{P}_1$)*
*For $k, l \geq 1$, and $p_1, \cdots, p_k, p_1', \cdots, p_l' \in \mathbf{P}_1$,*

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

**Proof.** See Appendix A.2. $\blacksquare$

**Proof of Lemma 7.**
**(1)** Here we prove the assertion for the operation $\|$. For the other operations this is proved in a similar fashion.

Let $\mathbf{H}_1 = (\mathcal{L}_1 \times \mathcal{L}_1 \to \mathbf{P}_1)$, and let $F, G \in \mathbf{H}_1$ be defined as follows:

For $s_1, s_2 \in \mathcal{L}_1$,

$$F(s_1, s_2) = \tilde{\mathcal{O}}_1[\![s_1 \parallel s_2]\!],$$

$$G(s_1, s_2) = \tilde{\mathcal{O}}_1[\![s_1]\!] \,\tilde{\|}\, \tilde{\mathcal{O}}_1[\![s_2]\!].$$

Let $\mathcal{F} : \mathbf{H}_1 \to \mathbf{H}_1$ be defined as follows:

For $f \in \mathbf{H}_1$ and $s_1, s_2 \in \mathcal{L}_1$,

$$\begin{aligned}\mathcal{F}(f)(s_1, s_2) = \; & \bigcup\{(\langle\sigma, \sigma'\rangle) \cdot f(s_1', s_2) : \langle s_1, \sigma\rangle \to_1 \langle s_1', \sigma'\rangle\} \\ & \cup \bigcup\{(\langle\sigma, \sigma'\rangle) \cdot f(s_1, s_2') : \langle s_2, \sigma\rangle \to_1 \langle s_2', \sigma'\rangle\} \\ & \cup \{(\sigma) : \neg\exists\langle s_1', \sigma'\rangle[\langle s_1, \sigma\rangle \to_1 \langle s_1', \sigma'\rangle] \\ & \qquad \wedge \neg\exists\langle s_2', \sigma'\rangle[\langle s_2, \sigma\rangle \to_1 \langle s_2', \sigma'\rangle]\ \}.\end{aligned}$$

Then $\mathcal{F}$ is a contraction.

Let $s_1, s_2 \in \mathcal{L}_1$. By the definitions of $\tilde{\mathcal{O}}_1$ and $\to_1$, and Lemma 3, it holds that

$$F(s_1, s_2) = \mathcal{F}(F)(s_1, s_2).$$

That is, $F = \text{fix}(\mathcal{F})$.

On the other hand, by the definition of $\tilde{\|}$ , one has

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

Moreover,

$$\begin{aligned}& \tilde{\mathcal{O}}_1[\![s_1]\!] \,\lfloor\!\lfloor\, \tilde{\mathcal{O}}_1[\![s_2]\!] \\ & = \bigcup\{(\langle\sigma, \sigma'\rangle) \cdot (\tilde{\mathcal{O}}_1[\![s_1]\!][\langle\sigma, \sigma'\rangle] \,\tilde{\|}\, \tilde{\mathcal{O}}_1[\![s_2]\!]) : \tilde{\mathcal{O}}_1[\![s_1]\!][\langle\sigma, \sigma'\rangle] \neq \emptyset\} \\ & = \bigcup\{(\langle\sigma, \sigma'\rangle) \cdot (\bigcup\{\tilde{\mathcal{O}}_1[\![s_1']\!] : \langle s_1, \sigma\rangle \to_1 \langle s_1', \sigma'\rangle\} \,\tilde{\|}\, \tilde{\mathcal{O}}_1[\![s_2]\!]) : \exists s_1'[\langle s_1, \sigma\rangle \to_1 \langle s_1', \sigma'\rangle]\} \\ & = \bigcup\{(\langle\sigma, \sigma'\rangle) \cdot (\bigcup\{\tilde{\mathcal{O}}_1[\![s_1']\!] \,\tilde{\|}\, \tilde{\mathcal{O}}_1[\![s_2]\!] : \langle s_1, \sigma\rangle \to_1 \langle s_1', \sigma'\rangle\} : \exists s_1'[\langle s_1, \sigma\rangle \to_1 \langle s_1', \sigma'\rangle]\} \\ & \quad \text{(by Lemma 8)} \\ & = \bigcup\{(\langle\sigma, \sigma'\rangle) \cdot (\tilde{\mathcal{O}}_1[\![s_1']\!] \,\tilde{\|}\, \tilde{\mathcal{O}}_1[\![s_2]\!]) : \langle s_1, \sigma\rangle \to_1 \langle s_1', \sigma'\rangle\}.\end{aligned}$$

Thus

$$\tilde{\mathcal{O}}_1[\![s_1]\!] \,\lfloor\!\lfloor\, \tilde{\mathcal{O}}_1[\![s_2]\!] = \bigcup\{(\langle\sigma, \sigma'\rangle) \cdot G(s_1', s_2) : \langle s_1, \sigma\rangle \to_1 \langle s_1', \sigma'\rangle\}.$$

Similarly

$$\tilde{\mathcal{O}}_1[\![s_2]\!] \,\lfloor\!\lfloor\, \tilde{\mathcal{O}}_1[\![s_1]\!] = \bigcup\{(\langle\sigma, \sigma'\rangle) \cdot G(s_2', s_1) : \langle s_2, \sigma\rangle \to_1 \langle s_2', \sigma'\rangle\}.$$

By the definition of $\#$ ,

$$\begin{aligned}& \tilde{\mathcal{O}}_1[\![s_1]\!] \# \tilde{\mathcal{O}}_1[\![s_2]\!] \\ & = \{(\sigma) : \neg\exists\langle s_1', \sigma'\rangle[\langle s_1, \sigma\rangle \to_1 \langle s_1', \sigma'\rangle] \wedge \neg\exists\langle s_2', \sigma'\rangle[\langle s_2, \sigma\rangle \to_1 \langle s_2', \sigma'\rangle]\ \}.\end{aligned}$$

Thus it holds that

$$G(s_1, s_2) = \mathcal{F}(G)(s_1, s_2).$$

Summing up, $G$ is also the fixed point of $\mathcal{F}$. Thus, by Banach's Theorem, one has $F = G$, i.e.,

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

**(2)** First, let us show, for $X \in \text{RVar}$, that

$$\tilde{\mathcal{O}}_1[\![X]\!] = \mathcal{D}_1[\![X]\!]. \tag{8}$$

Let $\langle X, g_X\rangle \in D$. Then,

$$\begin{aligned}& \tilde{\mathcal{O}}_1[\![X]\!] \\ & = \tilde{\mathcal{O}}_1[\![g_X]\!] \quad \text{(by the definition of } \tilde{\mathcal{O}}_1) \\ & = [\![g_X]\!]^{\mathcal{I}_1}[(\tilde{\mathcal{O}}_1[\![Y_1^X]\!], \cdots, \tilde{\mathcal{O}}_1[\![Y_{l(X)}^X]\!])/(Y_1^X, \cdots, Y_{l(X)}^X)]. \quad \text{(by (1))}\end{aligned} \tag{9}$$

Here $\{Y_1^X, \cdots, Y_{l(X)}^X\}$ is the set of recursion variables contained in $g_X$.

Hence $(\lambda X \in \text{RVar} : \tilde{\mathcal{O}}_1[\![X]\!])$ is the fixed point of $\Pi_1$ defined Definition 9. Therefore by the definition of $\mathcal{D}_1[\![X]\!]$, one has (8).

It follows from this and **(1)**, by induction on the structure of $s \in \mathcal{L}_1$, that

$$\forall s \in \mathcal{L}_1[\tilde{\mathcal{O}}_1[\![s]\!] = \mathcal{D}_1[\![s]\!]]. \quad \blacksquare$$

### 3.4.2 Correctness of $\mathcal{D}_1$ with respect to $\mathcal{O}_1$

An *abstraction function* $\alpha_1 : \mathbf{P}_1 \to (\Sigma \to \wp_{\rm nc}(\Sigma^{\leq\omega}))$ 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* are to be defined shortly.

**Definition 12** *(Abstraction Function $\alpha_1$for $\mathcal{L}_1$)*

**(1)** Let $\mathbf{M}_1^\alpha = (\mathbf{P}_1 \to (\Sigma \to \wp_{\rm nc}(\Sigma^{\leq\omega})))$, and let $\Delta_1 : \mathbf{M}_1^\alpha \to \mathbf{M}_1^\alpha$ be defined as follows:

For $F \in \mathbf{M}_1^\alpha$, $p \in \mathbf{P}_1$, and $\sigma \in \Sigma$,

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

Remark that the right-hand side of the above equation is nonempty, since $p$ is uniformly nonempty at level 0. Thus the mapping $\Delta_1$ 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

$$\begin{aligned}\alpha_1(p)(\sigma) = \ & \bigcup\{(\sigma') \cdot \alpha_1(p[\langle\sigma,\sigma'\rangle])(\sigma') : \exists q \in \mathbf{Q}_1[(\langle\sigma,\sigma'\rangle) \cdot q \in p]\} \\ & \cup \mathrm{if}((\sigma) \in p, \{\epsilon\}, \emptyset). \ \blacksquare\end{aligned}$$

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 $\sigma_i$. After this execution, the process is in the state $\sigma_i'$. 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 $\langle\sigma_0,\sigma_0'\rangle, \langle\sigma_1,\sigma_1'\rangle, \cdots$, and therefore is called *executable*.

**Definition 13** *(Histories of Elements of $\mathbf{P}_1$)*
Let $q \in \mathbf{Q}_1 \cup (\Sigma \times \Sigma)^+$.

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

$$\begin{aligned}&\exists(\langle\sigma_i,\sigma_i'\rangle)_{i\in I}[q = (\langle\sigma_i,\sigma_i'\rangle)_{i\in I} \wedge \forall i \in I[i > 0 \Rightarrow \sigma_{i-1}' = \sigma_i]] \vee \\ &\exists k \in \omega, \exists(\langle\sigma_i,\sigma_i'\rangle)_{i\in\bar{k}}, \exists\sigma_{k+1}[q = (\langle\sigma_i,\sigma_i'\rangle)_{i\in\bar{k}} \cdot (\sigma_{k+1}) \wedge \forall i \in \bar{k}[\sigma_i' = \sigma_{i+1}]\ ],\end{aligned}$$

where $I$ is $\omega$ or $\bar{n}$ for some $n \in \omega$.

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

$$\mathrm{hist}_1(q) = \begin{cases} (\sigma_i')_{i\in(I\setminus\{0\})} & \text{if } q = (\langle\sigma_i,\sigma_i'\rangle)_{i\in I}, \\ (\sigma_i')_{i\in\bar{k}} & \text{if } q = (\langle\sigma_i,\sigma_i'\rangle)_{i\in\bar{k}} \cdot (\sigma_{k+1}), \end{cases}$$

where $I$ is $\omega$ or $\bar{n}$ for some $n \in \omega$. $\blacksquare$

Now we can give another formulation of $\alpha_1$ as follows:

**Lemma 9** *(Another Formulation of Abstraction Function $\alpha_1$)*

**(1)** *For $p \in \mathbf{P}_1, \sigma \in \Sigma$, it holds that*
$\alpha_1(p)(\sigma) = \{\mathrm{hist}_1(q) : q \in p \wedge \mathrm{Exec}_1(q) \wedge \mathrm{istate}_1(q) = \sigma\}$.

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

**Proof.** See Appendix A.3. ■

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

**Lemma 10** *(Correctness of* $\mathcal{D}_1$*)*

**(1)** $\alpha_1 \circ \tilde{\mathcal{O}}_1 = \mathcal{O}_1$

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

**Proof.** **(1)** Let us define $\mathcal{O}'_1 : \mathcal{L}_1 \to (\Sigma \to \wp_{\rm nc}(\Sigma^{\leq\omega}))$ by $\mathcal{O}'_1 = \alpha_1 \circ \tilde{\mathcal{O}}_1$.
Fix $s \in \mathcal{L}_1$ and $\sigma \in \Sigma$. Let us show

$$\mathcal{O}'_1[\![s]\!](\sigma) = \Psi^{\mathcal{O}}_1(\mathcal{O}'_1)(s)(\sigma), \tag{10}$$

where $\Psi^{\mathcal{O}}_1$ is the higher-order function introduced in Definition 6. We distinguish two cases according to whether $\exists\langle s',\sigma'\rangle[\langle s,\sigma\rangle \to_1 \langle s',\sigma'\rangle]$.
**Case 1.** Suppose $\neg\exists\langle s',\sigma'\rangle[\langle s,\sigma\rangle \to_1 \langle s',\sigma'\rangle]$. Then, by the definition of $\Psi^{\mathcal{O}}_1$, both sides of (10) are equal to $\{\epsilon\}$, and therefore, (10) holds.
**Case 2.** Suppose $\exists\langle s',\sigma'\rangle[\langle s,\sigma\rangle \to_1 \langle s',\sigma'\rangle]$. Then

$$\begin{aligned}
&\mathcal{O}'_1[\![s]\!](\sigma)\\
&= \bigcup\{(\sigma')\cdot\alpha_1(\tilde{\mathcal{O}}_1[\![s]\!][\langle\sigma,\sigma'\rangle])(\sigma') : \tilde{\mathcal{O}}_1[\![s]\!][\langle\sigma,\sigma'\rangle] \neq \emptyset\}\\
&= \bigcup\{(\sigma')\cdot\alpha_1(\bigcup\{\tilde{\mathcal{O}}_1[\![s']\!] : \langle s,\sigma\rangle \to_1 \langle s',\sigma'\rangle\})(\sigma') : \exists s'[\langle s,\sigma\rangle \to_1 \langle s',\sigma'\rangle]\}\\
&= \bigcup\{(\sigma')\cdot\bigcup\{\alpha_1(\tilde{\mathcal{O}}_1[\![s']\!])(\sigma') : \langle s,\sigma\rangle \to_1 \langle s',\sigma'\rangle\} : \exists s'[\langle s,\sigma\rangle \to_1 \langle s',\sigma'\rangle]\}\\
&\qquad \text{(by Lemma 9 (2))}\\
&= \bigcup\{(\sigma')\cdot\alpha_1(\tilde{\mathcal{O}}_1[\![s']\!])(\sigma') : \langle s,\sigma\rangle \to_1 \langle s',\sigma'\rangle\}\\
&= \bigcup\{(\sigma')\cdot\mathcal{O}'_1[\![s']\!](\sigma') : \langle s,\sigma\rangle \to_1 \langle s',\sigma'\rangle\}\\
&= \Psi^{\mathcal{O}}_1(\mathcal{O}'_1)(s)(\sigma). \quad \text{(by the definition of } \Psi^{\mathcal{O}}_1\text{)}
\end{aligned}$$

Hence (10) holds.
Thus (10) holds for every $s$ and $\sigma$. This implies that

$$\alpha_1 \circ \tilde{\mathcal{O}}_1 = \mathcal{O}'_1 = \mathrm{fix}(\Psi^{\mathcal{O}}_1) = \mathcal{O}_1.$$

**(2)** This part follows from **(1)** and Lemma 7 (2). ■

### 3.5 Full Abstractness of $\mathcal{D}_1$ with respect to $\mathcal{O}_1$

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

> Given two statements $s_1, s_2 \in \mathcal{L}_1$ 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. .......(∗)

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

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

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

If $s_1$ and $s_2$ are distinct in their denotational meanings, then there exists some sequence $r$ such that $p_1[r] \neq \emptyset$ but $p_2[r] = \emptyset$ (or vice versa). By constructing a tester $T$ and an executable sequence $\tilde{r}$ for $r$ as above, one has

$\tilde{r} \in \mathcal{D}_1[\![s_1]\!] \,\tilde{\|}\, \mathcal{D}_1[\![T]\!]$ and $\tilde{r} \notin \mathcal{D}_1[\![s_2]\!] \,\tilde{\|}\, \mathcal{D}_1[\![T]\!]$.

Thus one has a difference between the operational meanings of the two statements $s_1 \| T$ and $s_2 \| T$.

First, the notion of *full abstractness* is defined.

**Definition 14** *(Full Abstractness)*
Let $\mathcal{L}$ be a language and $\mathcal{O}$ an operational model for $\mathcal{L}$. A denotational model $\mathcal{D}$ is said to be *fully abstract* with respect to the operational model $\mathcal{O}$ iff

$\forall s_1, s_2 \in \mathcal{L}_1[\forall S \in \mathcal{L}_1^*, \forall \xi \in \mathrm{SVar}[\ \mathcal{O}[\![S_{(\xi)}[s_1]]\!] = \mathcal{O}[\![S_{(\xi)}[s_2]]\!]\ ] \Leftrightarrow \mathcal{D}[\![s_1]\!] = \mathcal{D}[\![s_2]\!]\ ]$. ∎

For a language $\mathcal{L}$ which can be formulated as the set of terms generated by a single-sorted signature, and an operational model $\mathcal{O}$ for it, a fully abstract compositional model for $\mathcal{L}$ with respect to $\mathcal{O}$ is unique in the following sense and exists if $\mathcal{L}$ has no recursion, as was shown in [BKO 88].

**Lemma 11** *(Uniqueness and Existence of Fully Abstract Compositional Model)*
*If two compositional models $\mathcal{D}$ and $\mathcal{D}'$ are fully abstract with respect to $\mathcal{O}$, then there is an isomorphism from $\mathcal{D}[\mathcal{L}]$ to $\mathcal{D}'[\mathcal{L}]$, i.e.,*

$\exists \varphi : \mathcal{D}[\mathcal{L}] \to \mathcal{D}'[\mathcal{L}][\varphi$ *is a bijection* $\wedge$
$\forall F[F$ *is an operation in* $\mathcal{L}$ *with arity* $r \Rightarrow$
$\forall p_1, \cdots, p_r \in \mathcal{D}[\mathcal{L}][\varphi(F^{\mathcal{D}}(p_1, \cdots, p_r)) = F^{\mathcal{D}'}(\varphi(p_1), \cdots, \varphi(p_r))]]]$.

*In other words, the fully abstract compositional model is unique except for isomorphism.*
*Moreover, there exists a fully abstract compositional model, if $\mathcal{L}$ has no recursion.* ∎

**Proof.** See Proposition 7.1.1 in [BKO 88]. ∎

As a preliminary to the proof of the full abstractness of $\mathcal{D}_1$ with respect to $\mathcal{O}_1$, we present the next lemma. Roughly, the claim of the lemma is the that

if $p_1 \neq p_2$, then $p_1$ and $p_2$ are *uniformly distinct* , i.e., then one has, for *every* $\sigma \in \Sigma$,
$\exists T \in \mathcal{L}_1[\alpha_1(p_1 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma) \neq \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma)]$,
where $T$ is a tester.

Though it is sufficient for the full abstractness that

if $p_1 \neq p_2$, then $p_1$ and $p_2$ are distinct, i.e., then one has, for *some* $\sigma \in \Sigma$,
$\exists T \in \mathcal{L}_1[\alpha(p_1 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma) \neq \alpha(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma)]$,

we need this stronger assertion for the sake of inductive proof.

**Lemma 12** *(Uniform Distinction Lemma for $\mathcal{L}_1$)*

**(1)** *Let $r \in (\Sigma \times \Sigma)^+$, and $p_1, p_2 \in \mathbf{P}_1^*$.*
*If $p_1[r] \neq \emptyset$ and $p_2[r] = \emptyset$, then it holds that*
$\forall \sigma \in \Sigma, \exists T \in \mathcal{L}_1[\alpha_1(p_1 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma) \setminus \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma) \neq \emptyset]$.

**(2)** *Let $q \in (\Sigma \times \Sigma)^{<\omega} \times \Sigma$, and $p_1, p_2 \in \mathbf{P}_1^*$.*
*If $q \in p_1 \setminus p_2$, then it holds that*
$\forall \sigma \in \Sigma, \exists T \in \mathcal{L}_1[\alpha_1(p_1 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma) \setminus \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma) \neq \emptyset]$. ∎

The proof of this lemma is given later. First, the full abstractness of $\mathcal{D}_1$ is derived from it.

**Theorem 1** *(Full Abstractness of $\mathcal{D}_1$)*
$\forall s_1, s_2 \in \mathcal{L}_1[\mathcal{D}_1[\![s_1]\!] \neq \mathcal{D}_1[\![s_2]\!] \Rightarrow \exists T \in \mathcal{L}_1[\alpha_1(\mathcal{D}_1[\![s_1]\!] \,\tilde{\|}\, \mathcal{D}_1[\![T]\!]) \neq \alpha_1(\mathcal{D}_1[\![s_2]\!] \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])]]$. ∎

**Proof.** Let $p_1 = \mathcal{D}_1[\![s_1]\!]$, $p_2 = \mathcal{D}_1[\![s_2]\!]$, and suppose $p_1 \neq p_2$. Then either

$$\exists q \in \mathbf{Q}_1[q \in p_1 \wedge q \notin p_2]$$

or

$$\exists q \in \mathbf{Q}_1[q \notin p_1 \wedge q \in p_2].$$

We consider the first case; the result is obtained in the second case in the same fashion.

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 $n \in \omega$ such that $p_2[q^{[n]}] = \emptyset$. Assume 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 that $q \notin p_2$. Hence,

$$\exists n \in \omega[p_2[q^{[n]}] = \emptyset].$$

Let $r = q^{[n]}$. Then

$$p_1[r] \neq \emptyset \wedge p_2[r] = \emptyset.$$

From this and Lemma 12 (1), it follows that

$$\exists T \in \mathcal{L}_1[\alpha_1(p_1 \tilde{\|} \mathcal{D}_1[\![T]\!]) \neq \alpha_1(p_2 \tilde{\|} \mathcal{D}_1[\![T]\!])].$$

**Case 2.** If $q$ is finite, then it follows from Lemma 12 (2) that

$$\exists T \in \mathcal{L}_1[\alpha_1(p_1 \tilde{\|} \mathcal{D}_1[\![T]\!]) \neq \alpha_1(p_2 \tilde{\|} \mathcal{D}_1[\![T]\!])].$$

Thus in both cases, one has the desired result. ∎

Testers for proving Lemma 12 (1) are constructed by induction on $r$; for the part (2) they are constructed in the same fashion.

The following lemma is used to construct testers for $r$ with length $n+1$ by means of testers for $r$ with length $n$, as well as to construct testers for $r$ with length 1. It is assumed that $\mathbf{V}$ is infinite for its proof.

**Lemma 13** *(Testing Lemma for $\mathcal{L}_1$)*
*For a process $p \in \mathbf{P}_1^*$, and states $\sigma', \sigma'', \sigma_0 \in \Sigma$, there are two finite sequences $r_1, r_2 \in (\Sigma \times \Sigma)^{<\omega}$ such that the following hold:*

**(i)** *The sequence $r_1 \cdot \langle \sigma', \sigma'' \rangle \cdot r_2$ is executable and its initial state is $\sigma_0$,*

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

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

$$p[\langle \sigma', \sigma'' \rangle] \neq \emptyset \wedge q' \in p[\langle \sigma', \sigma'' \rangle] \tilde{\|} \mathcal{D}_1[\![T']\!] \Leftrightarrow r_1 \cdot \langle \sigma', \sigma'' \rangle \cdot r_2 \cdot q' \in p \tilde{\|} \mathcal{D}_1[\![T]\!]. \quad \blacksquare \tag{11}$$

The proof is formulated by supposing that IVar is reduced to one variable: IVar $= \{x\}$, which simplifies the proof allowing us to identify a state $\sigma$ with its value $\sigma(x)$ in $\mathbf{V}$. However, the lemma still holds when IVar is composed of more than one variable, as established in Appendix A.4.

Under this assumption $\Sigma$ is identified with $\mathbf{V}$ and the states $\sigma'$, $\sigma''$ and $\sigma_0$ are represented by $v', v'', v_0 \in \mathbf{V}$, respectively.

Trying to construct such $T$, we first observe that the composition $p \tilde{\|} \mathcal{D}_1[\![T]\!]$ must be in the state $v'$ when $p$ executes the step $\langle v', v'' \rangle$.

Therefore, if $v_0 \neq v'$, then $\mathcal{D}[\![T]\!]$ must execute the step $\langle v, v' \rangle$, and therefore, $T$ must have an assignment "$x := v'$" in it. Moreover, we need a trick for forbidding $p$ to execute the step $\langle v_0, v' \rangle$ for $\mathcal{D}[\![T]\!]$ and forbidding $\mathcal{D}[\![T]\!]$ to execute the step $\langle v', v'' \rangle$ for $p$. A tester $T$ with these properties can be constructed in the following format:

**Definition 15** *(Format for Testers)*
For $v, v_0, v_1, v_2 \in \mathbf{V}$, and $T' \in \mathcal{L}_1$, let

$$F(v, v_0, v_1, v_2, T') \equiv \text{If}(x = v_0,\ (x := v); (x := v_1); T',\ (x := v_2); \text{Stop}). \blacksquare$$

**Proof of Lemma 13.**
The proof is given by distinguishing two cases according to whether $v_0 = v'$.
**Case 1.** When $v_0 = v'$, we can easily construct two sequences $r_1, r_2$ satisfying **(i)** and **(ii)** in Lemma 13 as follows:

$$r_1 = \epsilon,\quad r_2 = \langle v'', v_1 \rangle,$$

where $v_1$ is chosen such that

$$\begin{cases} \text{(i) } v_1 \neq v'', \\ \text{(ii) } v_1 \neq \{v \in \mathbf{V} : \langle v', v'' \rangle \cdot \langle v'', v \rangle \in p^{[2]}\}. \end{cases} \tag{12}$$

Remark that the right-hand side of (12) (ii) is finite by Lemma 6 (4), and therefore there is $v_1$ satisfying (12).

It is immediate that **(i)** holds. Let us show **(ii)**. For every $T' \in \mathcal{L}_1$, let

$$T \equiv (x := v_1); T'.$$

It is immediate that (ii) (a) holds. Let us show (ii) (b), i.e., (11) holds for every $q' \in \mathbf{Q}_1$.
($\Rightarrow$) This part is immediate, since if $q' \in p[\langle v', v'' \rangle] \tilde{\|} \mathcal{D}_1[\![T']\!]$, then the parallel composition $p \tilde{\|} \mathcal{D}_1[\![T]\!]$ can executes the sequence

$$r_1 \cdot \langle v', v'' \rangle \cdot r_2 \cdot q' = \langle v', v'' \rangle \cdot \langle v'', v_1 \rangle \cdot q'$$

with the first two step $\langle v', v'' \rangle$, and $\langle v'', v_1 \rangle$ stemming from $p$ and $\mathcal{D}_1[\![T]\!]$, respectively.
($\Leftarrow$) Suppose

$$\langle v', v'' \rangle \cdot \langle v'', v_1 \rangle \cdot q' \in p \tilde{\|} \mathcal{D}_1[\![T]\!].$$

Let us show that the first two steps $\langle v', v'' \rangle$ and $\langle v'', v_1 \rangle$ must stem from $p$ and $\mathcal{D}_1[\![T]\!]$, respectively.

The first step cannot stem from $\mathcal{D}_1[\![T]\!]$ by (12) (i). Also, the second step cannot stem from $p$ by (12) (ii). Thus one has the desired result.
**Case 2.** When $v_0 \neq v'$, we can construct two sequences $r_1$, $r_2$ satisfying **(i)** and **(ii)** in Lemma 13 as follows:

$$r_1 = \langle v_0, v' \rangle,\quad r_2 = \langle v'', v_1 \rangle, \tag{13}$$

where $v_1$ is chosen such that

$$\begin{cases} \text{(i) } v_1 \neq \{v \in \mathbf{V} : \langle v_0, v'' \rangle \cdot \langle v', v'' \rangle \cdot \langle v'', v \rangle \in p^{[3]}\}, \\ \text{(ii) } v_1 \neq v', \\ \text{(iii) } v_1 \neq v'', \\ \text{(iv) } v_1 \neq \{v \in \mathbf{V} : \langle v', v'' \rangle \cdot \langle v'', v \rangle \in p^{[2]}\}. \end{cases} \tag{14}$$

Remark that the right-hand sides of (14) (i) and (iv) are finite, since $p$ is image finite by Lemma 6 (4). Therefore there is $v_1$ satisfying (14).

It is immediate that **(i)** holds. Let us show **(ii)**. For every $T' \in \mathcal{L}_1$, let

$$T \equiv F(v', v_0, v_1, v_2, T'),$$

where $v_2$ is chosen such that

$$\begin{cases} \text{(i) } v_2 \neq v'', \\ \text{(ii) } v_2 \neq v_1. \end{cases} \tag{15}$$

In this case also, it is immediate that **(ii) (a)** holds. Let us show **(ii) (b)**, i.e., (11) holds for every $q' \in \mathbf{Q}_1$. First, put

$$t' = \mathcal{D}_1[\![T']\!],$$
$$t = \mathcal{D}_1[\![T]\!].$$

**(⇒) This part is immediate, since if $q' \in p[\langle v', v''\rangle] \tilde{\|} t'$, then the parallel composition $p \tilde{\|} t$ executes the sequence**

$$r_1 \cdot \langle v', v''\rangle \cdot r_2 \cdot q' = \langle v_0, v'\rangle \cdot \langle v', v''\rangle \cdot \langle v'', v_1\rangle \cdot q'$$

with the first three steps $\langle v_0, v'\rangle$, $\langle v', v''\rangle$, and $\langle v'', v_1\rangle$ stemming from $t$, $p$, and $t$, respectively.
(⇐) Suppose

$$\langle v_0, v'\rangle \cdot \langle v', v''\rangle \cdot \langle v'', v_1\rangle \cdot q' \in p \tilde{\|} t.$$

Let us show that the first three steps $\langle v_0, v'\rangle$, $\langle v', v''\rangle$, and $\langle v'', v_1\rangle$ must stem from $t$, $p$, and $t$, respectively.

First, let us show by contradiction that the first step $\langle v_0, v'\rangle$ cannot stem from $p$.

Assume that the first step stems from $p$, i.e.,

$$\langle v', v''\rangle \cdot \langle v'', v_1\rangle \cdot q' \in p[\langle v_0, v'\rangle] \tilde{\|} t.$$

Then the second step $\langle v', v''\rangle$ must stem from either of $p[\langle v_0, v'\rangle]$ or $t$; let us show that it can stem from neither of them.

Suppose that the second step stems from $t$, i.e.,

$$\langle v'', v_1\rangle \cdot q' \in p[\langle v_0, v'\rangle] \tilde{\|} t[\langle v', v''\rangle].$$

Then $t[\langle v', v''\rangle] \neq \emptyset$, and therefore, under the assumption

$$v_0 \neq v',$$

the assignment "$x := v_2$" must be executed in the second step, which yields $v'' = v_2$. However this contradicts (15) (i). Thus

$$\langle v'', v_1\rangle \cdot q' \in p[\langle v_0, v'\rangle \cdot \langle v', v''\rangle] \tilde{\|} t.$$

The third step $\langle v'', v_1\rangle$ cannot stem from $p[\langle v_0, v'\rangle \cdot \langle v', v''\rangle]$, since, by (14) (i),

$$p[\langle v_0, v'\rangle \cdot \langle v', v''\rangle \cdot \langle v'', v_1\rangle] = \emptyset.$$

Thus the third step must stem from $t$, which implies $v_1 = v'$ or $v_1 = v_2$. However both are impossible by (14) (ii) and (15) (ii), respectively.

Summing up, the first step cannot stem from $p$, and therefore, it must stem from $t$. Thus one has

$$\langle v', v''\rangle \cdot \langle v'', v_1\rangle \cdot q' \in p \tilde{\|} t[\langle v_0, v'\rangle].$$

Then, let us show that the second step $\langle v', v''\rangle$ cannot stem from $t[\langle v_0, v'\rangle]$. If it stems from $t[\langle v_0, v'\rangle]$, then

$$t[\langle v_0, v'\rangle \cdot \langle v', v''\rangle] \neq \emptyset,$$

which implies, by the form of $T$, that

$$v'' = v_1.$$

This contradicts (14) (iii). Thus the second step must stem from $p$, and therefore

$$\langle v'', v_1\rangle \cdot q' \in p[\langle v', v''\rangle] \tilde{\|} t[\langle v_0, v'\rangle].$$

Finally, the third step $\langle v'', v_1\rangle$ cannot stem from $p[\langle v', v''\rangle]$, since $p[\langle v', v''\rangle \cdot \langle v'', v_1\rangle] = \emptyset$ by (14) (iv). Thus the third step must stem from $t[\langle v_0, v'\rangle]$, and therefore

$$q' \in p[\langle v', v''\rangle] \tilde{\|} t[\langle v_0, v'\rangle \cdot \langle v'', v_1\rangle],$$

that is,

$$q' \in p[\langle v', v''\rangle] \tilde{\|} \mathcal{D}_1[\![T']\!]. \quad \blacksquare$$

**Remark 3** Remark that if $v_0 \neq v'$ and $v' \neq v''$, then a simpler tester

$$T \equiv (x := v'); (x := v_1); T',$$

with $v_1$ satisfying (14), is sufficient to establish the above lemma.

However if $v_0 \neq v'$ and $v' = v''$, then we need that tester defined by (52), for excluding the possibility that the first three steps of the parallel composition may stem from $p$, $t$, and $t$, respectively. ▮

The following proposition follows immediately from Lemma 13; this corollary is to play a central role in the proof of Lemma 12.

**Corollary 1** *Let $p \in \mathbf{P}_1^*$, $\langle \sigma', \sigma'' \rangle \in \Sigma^2$, and $\sigma_0 \in \Sigma$.*

*There are two finite sequences $\rho_1, \rho_2 \in \Sigma^{<\omega}$ such that for every tester $T' \in \mathcal{L}_1$ there exists another tester $T \in \mathcal{L}_1$ such that the following holds:*

> *Let $\sigma_1$ be the last element of the finite sequence $\rho_1 \cdot \sigma'' \cdot \rho_2$. Then the following holds for every $\rho' \in \Sigma^{\leq\omega}$:*
>
> $$p[\langle \sigma', \sigma'' \rangle] \neq \emptyset \wedge \rho' \in \alpha_1(p[\langle \sigma', \sigma'' \rangle] \tilde{\|} \mathcal{D}_1[\![T']\!])(\sigma_1) \Leftrightarrow \\ \rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p \tilde{\|} \mathcal{D}_1[\![T]\!])(\sigma_0). \quad \blacksquare \tag{16}$$

**Proof.** Take $r_1$, $r_2$, and $T$ as in Lemma 13, and put $\rho_1 = \text{hist}_1(r_1)$, $\rho_2 = \text{hist}_1(r_2)$, and let $\sigma_1$ be the last element of $\rho_1 \cdot \sigma'' \cdot \rho_2$. Let us show (16) holds for every $\rho' \in \Sigma^{\leq\omega}$.
($\Rightarrow$) Suppose $p[\langle \sigma', \sigma'' \rangle] \neq \emptyset$ and
$\rho' \in \alpha_1(p[\langle \sigma', \sigma'' \rangle] \tilde{\|} \mathcal{D}_1[\![T']\!])(\sigma_1)$.
Then, by Lemma 9 (i), there exists executable $q' \in p[\langle \sigma', \sigma'' \rangle] \tilde{\|} \mathcal{D}_1[\![T']\!]$ such that $\text{istate}_1(q') = \sigma_1$ and $\text{hist}_1(q') = \rho'$. Fix such $q'$. By **(i)** in Lemma 13, $r_1 \cdot \langle \sigma', \sigma'' \rangle \cdot r_2 \cdot q'$ is executable and its initial state is $\sigma_0$. Thus, by the $\Rightarrow$-part of (11), one has
$\text{hist}_1(r_1 \cdot \langle \sigma', \sigma'' \rangle \cdot r_2 \cdot q') \in \alpha_1(p \tilde{\|} \mathcal{D}_1[\![T]\!])(\sigma_0)$,
that is,
$\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p \tilde{\|} \mathcal{D}_1[\![T]\!])(\sigma_0)$.
($\Leftarrow$) Suppose
$\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p \tilde{\|} \mathcal{D}_1[\![T]\!])(\sigma_0)$.
Then, by by Lemma 9 (i), there exists executable $q'$ such that

$$\begin{cases} \text{(i) } \text{istate}_1(q') = \sigma_1 \\ \text{(ii) } \text{hist}_1(q') = \rho'. \end{cases} \tag{17}$$

Fix such $q'$. By the $\Leftarrow$-part of (11), one has
$p[\langle \sigma', \sigma'' \rangle] \neq \emptyset$
and
$q' \in p[\langle \sigma', \sigma'' \rangle] \tilde{\|} \mathcal{D}_1[\![T']\!]$.
Thus, by (17), one has
$\rho' = \text{hist}_1(q') \in \alpha_1(p[\langle \sigma', \sigma'' \rangle] \tilde{\|} \mathcal{D}_1[\![T']\!])(\sigma_1)$. $\blacksquare$

We now prove Lemma 12. By means of Lemma 13, we can construct appropriate testers by induction on the length of $r \in (\Sigma \times \Sigma)^+$ and on the length of $q \in (\Sigma \times \Sigma)^{<\omega} \times \Sigma$ for proving **(1)** and **(2)** of Lemma 12 respectively as follows:

**Proof of Lemma 12.**
**(1)** Let us establish the first part of Lemma 12. To this end, we will prove, by induction on the length of $r \in (\Sigma \times \Sigma)^+$, that the following holds:

> For every $p_1, p_2 \in \mathbf{P}_1^*$, if $p_1[r] \neq \emptyset$ and $p_2[r] = \emptyset$, then
>
> $$\forall \sigma_0 \in \Sigma, \exists T \in \mathcal{L}_1[\alpha_1(p_1 \tilde{\|} \mathcal{D}_1[\![T]\!])(\sigma_0) \setminus \alpha_1(p_2 \tilde{\|} \mathcal{D}_1[\![T]\!])(\sigma_0) \neq \emptyset].$$

**Induction Base:** Suppose $\text{lgt}(r) = 1$, and $r = (\langle \sigma', \sigma'' \rangle)$. Let $p_1, p_2 \in \mathbf{P}_1^*$ such that

$$\begin{cases} \text{(i) } p_1[r] \neq \emptyset, \\ \text{(ii) } p_2[r] = \emptyset. \end{cases} \tag{18}$$

Finally let $\sigma_0 \in \Sigma$.
By Corollary 1, there are $\rho_1$, $\rho_2$, $T$ such that the following holds:

> Let $\sigma_1$ be the last element of $\rho_1 \cdot \sigma'' \cdot \rho_2$. Then it holds for every $\rho' \in \Sigma^{\leq\omega}$ that

$$p_2[\langle\sigma',\sigma''\rangle] \neq \emptyset \wedge \rho' \in \alpha_1(p_2[\langle\sigma',\sigma''\rangle] \,\tilde{\|}\, \text{stop}_1)(\sigma_1) \\ \text{iff } \rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0). \tag{19}$$

By (18) (i), there exists $\rho' \in \alpha_1(p_1[r] \,\tilde{\|}\, \text{stop}_1)(\sigma_1)$. Let us fix such $\rho'$. By (19), on has
$\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p_1 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0)$.
Next let us show by contradiction that
$\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \notin \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0)$.
Assume that

$$\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0). \tag{20}$$

Then, by (19), one has
$p_2[\langle\sigma',\sigma''\rangle] \neq \emptyset$,
which contradicts (18) (ii). Hence,
$\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \notin \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0)$.
**Induction Step:** Assume that the claim holds for every $r$ such that $\text{lgt}(r) \leq k$. Fix arbitrary $r$ of length $k+1$, say $r = \langle\sigma',\sigma''\rangle \cdot \tilde{r}$, and let $p_1, p_2$ such that

$$\text{(i) } p_1[r] \neq \emptyset, \\ \text{(ii) } p_2[r] = \emptyset. \tag{21}$$

Finally let $\sigma_0 \in \Sigma$. Two cases should be taken into account. On the one hand, if $p_2[\langle\sigma',\sigma''\rangle] = \emptyset$, then it is possible to construct $T_2$ as in the induction base, such that
$\exists \rho' \in \Sigma^{\leq\omega}[\rho' \in \alpha_1(p_1 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0) \wedge \rho' \notin \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0)]$.
On the other hand, if $p_2[\langle\sigma',\sigma''\rangle] \neq \emptyset$, then one has, by (21), denoting $p_1[\langle\sigma',\sigma''\rangle]$ and $p_2[\langle\sigma',\sigma''\rangle]$ by $p_1'$ and $p_2'$ respectively, that

$$p_1'[\tilde{r}] \neq \emptyset \wedge p_2'[\tilde{r}] = \emptyset. \tag{22}$$

By Corollary 1, there are $\rho_1, \rho_2$ such that for every $T' \in \mathcal{L}_1$ there exists $T$ such that the following holds:

Let $\sigma_1$ be the last element of $\rho_1 \cdot \sigma'' \cdot \rho_2$. Then it holds, for every $\rho' \in (\Sigma)^{\leq\omega}$, that

$$p_2[\langle\sigma',\sigma''\rangle] \neq \emptyset \wedge \rho' \in \alpha_1(p[\langle\sigma',\sigma''\rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!])(\sigma_1) \\ \text{iff } \rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0). \tag{23}$$

By the induction hypothesis and (22), there are $T'$ and $\rho'$ such that

$$\rho' \in \alpha_1(p_1' \,\tilde{\|}\, \mathcal{D}_1[\![T']\!])(\sigma_1) \setminus \alpha_1(p_2' \,\tilde{\|}\, \mathcal{D}_1[\![T']\!])(\sigma_1). \tag{24}$$

Let $\rho = \rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho'$, and take $T$ such that (23) holds. By (23) and (24), one has
$\rho \in \alpha_1(p_1 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0)$.
Let us now show by contradiction that
$\rho \notin \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0)$.
Assume that

$$\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0). \tag{25}$$

Then, it follows from (23) that
$\rho' \in \alpha_1(p_2' \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_1)$,
which contradicts (24).
Summing up, in this case too, there is $\rho$ such that
$\rho \in \alpha_1(p_1 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0) \setminus \alpha_1(p_2 \,\tilde{\|}\, \mathcal{D}_1[\![T]\!])(\sigma_0)$.
**(2)** Let us establish the second part of Lemma 12, To this end, we will prove, by induction on the length of $q$, that for $q \in (\Sigma \times \Sigma)^{<\omega} \times \Sigma$, the following holds:

For every $p_1, p_2 \in \mathbf{P}_1^*$, if $q \in (p_1 \setminus p_2)$ then
$\forall \sigma_0 \in \Sigma, \exists T \in \mathcal{L}_1[\alpha_1(p_1 \tilde{\|} \mathcal{D}_1[\![T]\!])(\sigma_0) \setminus \alpha_1(p_2 \tilde{\|} \mathcal{D}_1[\![T]\!])(\sigma_0) \neq \emptyset]$.

**Induction Base:** Suppose $\mathrm{lgt}(q) = 1$, and $q = (\sigma')$. Assume

$$q \in p_1 \setminus p_2. \tag{26}$$

Let $\sigma_0 \in \Sigma$, and let

$$\begin{aligned} &T \equiv (x := \sigma'); \mathrm{Stop}, \\ &t = \mathcal{D}_1[\![T]\!]. \end{aligned} \tag{27}$$

By the definition of $\tilde{\|}$, one has
$(\langle \sigma_0, \sigma' \rangle, \sigma') \in p_1 \tilde{\|}\, t$, i.e., $(\sigma') \in \alpha_1(p_1 \tilde{\|}\, t)(\sigma_0)$.
Let us now prove by contradiction that
$(\langle \sigma_0, \sigma' \rangle, \sigma') \notin p_2 \tilde{\|}\, t$.
Indeed, if
$(\langle \sigma_0, \sigma' \rangle, \sigma') \in p_2 \tilde{\|}\, t$,
then one of the following statements should hold:

**(i)** $(\sigma') \in p_2[\langle \sigma_0, \sigma' \rangle] \tilde{\|}\, t$,
**(ii)** $(\sigma') \in p_2 \tilde{\|}\, t[\langle \sigma_0, \sigma' \rangle]$.

However, by the definition of $\tilde{\|}$, both **(i)** and **(ii)** are impossible, since $(\sigma') \notin t$ and $(\sigma') \notin p_2$, respectively.
Summing up, one has
$(\langle \sigma_0, \sigma' \rangle, \sigma') \notin p_2 \tilde{\|}\, t$, i.e., $(\sigma') \notin \alpha_1(p_2 \tilde{\|}\, t)(\sigma_0)$.
**Induction Step:** Similar to the induction step of **(1)**. ■

## 3.6 Relation between $\mathcal{D}_1$ and Other Denotational Models

In [BR 90], another denotational model $\mathcal{D}_1'$ for for a language, which is like $\mathcal{L}_1$ but has general sequential composition instead of prefixing, was proposed. The model $\mathcal{D}_1'$ was presented on the basis of the domain:
$\mathbf{P}_1' = \wp_{\mathrm{nc}}(\mathbf{Q}_1')$,
where
$\mathbf{Q}_1' \cong \{\epsilon\} \cup (\Sigma \to (\Sigma \times \mathbf{Q}_1'))$.
The outline of $\mathcal{D}_1'$ is as follows; the interpretation of the parallel composition is omitted, since this is not necessary for the present purpose.

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

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

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

However $\mathcal{D}_1'$ is not fully abstract with respect to $\mathcal{O}_1$ as the next example shows.

**Example 3** Let
$s_1 \equiv ((x := 0); \mathrm{Stop}) + ((x := 1); \mathrm{Stop})$,
$s_2 \equiv \mathrm{If}(x = 0, (x := 0); \mathrm{Stop}, (x := 1); \mathrm{Stop})$
$\quad + \mathrm{If}(x = 0, (x := 1); \mathrm{Stop}, (x := 0); \mathrm{Stop})$.
Then,

$$\mathcal{D}_1[\![s_1]\!] = \mathcal{D}_1[\![s_2]\!] = \{\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle\} \times \mathrm{stop}_1. \tag{28}$$

On the other hand

$$\mathcal{D}'_1[\![s_1]\!] = \{q_1, q_2\},$$

where, writing sets to denote functions,

$$q_1 = \{\langle 0, \langle 0, \epsilon\rangle\rangle, \langle 1, \langle 0, \epsilon\rangle\rangle\},$$
$$q_2 = \{\langle 0, \langle 1, \epsilon\rangle\rangle, \langle 1, \langle 1, \epsilon\rangle\rangle\}.$$

Also,

$$\mathcal{D}'_1[\![s_2]\!] = \{q'_1, q'_2\},$$

where

$$q'_1 = \{\langle 0, \langle 0, \epsilon\rangle\rangle, \langle 1, \langle 1, \epsilon\rangle\rangle\},$$
$$q'_2 = \{\langle 0, \langle 1, \epsilon\rangle\rangle, \langle 1, \langle 0, \epsilon\rangle\rangle\}.$$

Hence

$$\mathcal{D}'_1[\![s_1]\!] \neq \mathcal{D}'_1[\![s_2]\!]. \tag{29}$$

If $\mathcal{D}'_1$ is also fully abstract, then one has

$$\forall s_1, s_2 \in \mathcal{L}_1[\mathcal{D}_1[\![s_1]\!] = \mathcal{D}_1[\![s_2]\!] \Leftrightarrow \mathcal{D}'_1[\![s_1]\!] = \mathcal{D}'_1[\![s_2]\!]],$$

which contradicts (28) and (29). Hence $\mathcal{D}'_1$ cannot be fully abstract. ∎

# 4 A Nonuniform Language with Parallel Composition and Communication

The second language $\mathcal{L}_2$ 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 $\mathcal{L}_2$ is given as in Section 3.

The domain of a denotational model $\mathcal{D}_2$ for $\mathcal{L}_2$ is a kind of *failures model*, which was introduced in [BHR 84], adapted to the nonuniform setting. Each element of the domain is a set consisting of such elements as

$$\langle(\langle s_i, a_i, s'_i\rangle)_i, \langle s'', C\rangle\rangle,$$

where $s_i$, $s'_i$, and $s''$ are states, $a_i$ is an action and $C$ is a set of *communication sorts*. These elements are called *failures*; the parts $(\langle s_i, a_i, s'_i\rangle)_i$ and $\langle s'', C\rangle$ are called a *trace* and a *refusal*, respectively.

First, the correctness of $\mathcal{D}_2$ is established as in Section 3. Then, the full abstractness of $\mathcal{D}_2$ is established by a combination of the testing method introduced in Section 3 and the method proposed by Bergstra, Klop, and Olderog in [BKO 88] to establish the full abstractness of a *failures model* for a uniform language without recursion. This method was adapted by Rutten in [Rut 89] so as 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 abstractness of the denotational model for $\mathcal{L}_2$ is established as follows: Given two statements $s_1$ and $s_2$ of $\mathcal{L}_2$ 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 testing method, otherwise we can construct a tester by the method of Bergstra, Klop, and Olderog.

## 4.1 The Language $\mathcal{L}_2$

In addition to all constructs of $\mathcal{L}_1$, the language $\mathcal{L}_2$ 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 16** *(Language $\mathcal{L}_2$)*
The set of statements of the nonuniform concurrent language $(S \in)\ \mathcal{L}_2^*$ is defined by the following BNF-syntax:

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

Here $X$ ranges over RVar, the set of recursion variables; $\xi$ range 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

$(s \in)\mathcal{L}_2 = \{S \in \mathcal{L}_2^* : \mathrm{FV}(S) = \emptyset\}$.

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

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

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

As for $\mathcal{L}_1$, $\mathcal{L}_2^*$ and $\mathcal{L}_2$ can be formulated as the set of terms and the set of closed terms generated by a signature $\mathcal{S}_2$, respectively.

## 4.2 Operational Model $\mathcal{O}_2$ for $\mathcal{L}_2$

An operational model $\mathcal{O}_2$ for $\mathcal{L}_2$ is defined in terms of a transition relation $\longrightarrow_2$. The following definition is given as a preliminary to the definition of $\longrightarrow_2$.

**Definition 17** *(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)** The function sort : $\mathbf{A} \rightarrow$ ASort is defined as follows:

For $a \in \mathbf{A}$,

$$\mathrm{sort}(a) = \begin{cases} \gamma & \text{if } \exists v[a = \langle \gamma, v\rangle], \\ \tau & \text{otherwise.} \end{cases}$$ ∎

The transition relation $\longrightarrow_2 \subseteq (\mathcal{L}_2 \times \Sigma) \times \mathbf{A} \times (\mathcal{L}_2 \times \Sigma)$ is defined as follows. For $s_1, s_2 \in \mathcal{L}_2, \sigma_1, \sigma_2 \in \Sigma$, and $a \in \mathbf{A}$, we write $\langle s_1, \sigma_1\rangle \xrightarrow{a}_2 \langle s_2, \sigma_2\rangle$ for $(\langle s_1, \sigma_1\rangle, a, \langle s_2, \sigma_2\rangle) \in \longrightarrow_2$; 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 18** *(Transition Relation $\longrightarrow_2$)*

The transition relation $\longrightarrow_2$ is defined as the smallest relation satisfying the following rules (1) to (7).

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

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

**(3)** For every $v \in \mathbf{V}$, we have the following axiom:

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

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

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

**(5)**
$$\frac{\langle s_1,\sigma\rangle \xrightarrow{a}_2 \langle s,\sigma'\rangle}{\begin{array}{l}\langle s_1 + s_2,\sigma\rangle \xrightarrow{a}_2 \langle s,\sigma'\rangle \\ \langle s_2 + s_1,\sigma\rangle \xrightarrow{a}_2 \langle s,\sigma'\rangle\end{array}}$$

**(6)** **(i)** For every action $a \in \mathbf{A}$, we have

$$\frac{\langle s_1,\sigma\rangle \xrightarrow{a}_2 \langle s,\sigma'\rangle}{\begin{array}{l}\langle s_1 \parallel s_2,\sigma\rangle \xrightarrow{a}_2 \langle s \parallel s_2,\sigma'\rangle \\ \langle s_2 \parallel s_1,\sigma\rangle \xrightarrow{a}_2 \langle s_2 \parallel s,\sigma'\rangle\end{array}}$$

**(ii)** For every $c \in \mathrm{Chan}$ and $v \in \mathbf{V}$, we have

$$\frac{\langle s_1,\sigma\rangle \xrightarrow{c!v}_2 \langle s_1',\sigma\rangle \quad \langle s_2,\sigma\rangle \xrightarrow{c?v}_2 \langle s_2',\sigma'\rangle}{\begin{array}{l}\langle s_1 \parallel s_2,\sigma\rangle \xrightarrow{\tau}_2 \langle s_1' \parallel s_2',\sigma'\rangle \\ \langle s_2 \parallel s_1,\sigma\rangle \xrightarrow{\tau}_2 \langle s_2' \parallel s_1'\sigma'\rangle\end{array}}$$

**(7)** For each $\langle X, g_X\rangle \in D$,

$$\frac{\langle g_X,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle}{\langle X,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle} \quad \blacksquare$$

The transition relation is *image finite* in the following sense.

**Lemma 14**

**(1)** $\forall s \in \mathcal{L}_2, \forall \sigma \in \Sigma, \forall a \in \mathbf{A}$
$[\{\langle s',\sigma'\rangle \in \mathcal{L}_2 \times \Sigma : \langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle\}$ *is finite*$]$.

**(2)** $\forall s \in \mathcal{L}_2, \forall \sigma \in \Sigma$
$[\{\mathrm{sort}(a) : \exists \langle s',\sigma'\rangle \in \mathcal{L}_2 \times \Sigma[\langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle]\}$ *is finite*$]$.

**(3)** $\forall s \in \mathcal{L}_2, \forall \sigma \in \Sigma, \forall c \in \mathrm{Chan}$
$[\{v \in \mathbf{V} : \exists \langle s',\sigma'\rangle \in \mathcal{L}_2 \times \Sigma[\langle s,\sigma\rangle \xrightarrow{c!v}_2 \langle s',\sigma'\rangle]\}$ *is finite*$]$. ∎

**Proof.** These are shown in a similar fashion 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 19** *(Operational Model $\mathcal{O}_2$ for $\mathcal{L}_2$)*

**(1)** Let $\mathrm{act} : \mathcal{L}_2 \times \Sigma \rightarrow \wp(\mathbf{A})$ be defined by
$\mathrm{act}(s,\sigma) = \{a \in \mathbf{A} : \exists\langle s',\sigma'\rangle \in \mathcal{L}_2 \times \Sigma[\langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle]\}$.

**(2)** Let $\mathbf{M}_2^{\mathcal{O}} = (\mathcal{L}_2 \rightarrow (\Sigma \rightarrow \wp_{\mathrm{nc}}((\mathbf{A} \times \Sigma)^{\leq\omega})))$, and let $\Psi_2^{\mathcal{O}} : \mathbf{M}_2^{\mathcal{O}} \rightarrow \mathbf{M}_2^{\mathcal{O}}$ be defined by:

For $f \in \mathbf{M}_2^{\mathcal{O}}$, $s \in \mathcal{L}_2$, and $\sigma \in \Sigma$,

$$\begin{array}{rl}\Psi_2^{\mathcal{O}}(f)(s)(\sigma) = & \bigcup\{\langle a,\sigma'\rangle \cdot f(s')(\sigma') : \langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle\} \\ & \cup \mathrm{If}(\tau \notin \mathrm{act}(s,\sigma), \{\epsilon\}, \emptyset).\end{array}$$

The right-hand side of the above equation is obviously nonempty and closed by Lemma 14 (1), and therefore, $\Psi_2^{\mathcal{O}}$ is a contraction from $\mathbf{M}_2^{\mathcal{O}}$ to $\mathbf{M}_2^{\mathcal{O}}$.

**(3)** Let the operational model $\mathcal{O}_2$ be the unique fixed point of $\Psi_2^{\mathcal{O}}$. By the definition, one has

$$\mathcal{O}_2 : \mathcal{L}_2 \to (\Sigma \to \wp_{\rm nc}((\mathbf{A} \times \Sigma)^{\leq\omega})),$$

and for each $s \in \mathcal{L}_2$ and $\sigma \in \Sigma$,

$$\begin{aligned}\mathcal{O}_2[\![s]\!](\sigma) = \ & \bigcup\{\langle a, \sigma'\rangle \cdot \mathcal{O}_2[\![s']\!](\sigma') : \langle s, \sigma\rangle \xrightarrow{a}_2 \langle s', \sigma'\rangle\} \\ & \cup \mathrm{If}(\tau \notin \mathrm{act}(s, \sigma), \{\epsilon\}, \emptyset). \quad \blacksquare\end{aligned}$$

## 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 [BHR 84], adapted to the nonuniform setting. Each element of the domain is a set consisting of elements
$\langle (\langle s_i, a_i, s'_i\rangle)_i, \langle s'', C\rangle\rangle$,
where $s_i$, $s'_i$, and $s''$ are states, $a_i$ is an action and $C$ is a set of *communication sorts*. These elements are called *failures*. Formally the domain $\mathbf{P}_2$ is defined by:

**Definition 20** *(Denotational Semantic Domain $\mathbf{P}_2$ for $\mathcal{L}_2$)*

**(1)** Let $\mathbf{Q}_2$ be the unique solution of
$\mathbf{Q}_2 \cong (\Sigma \times \wp(\mathbf{C})) \uplus ((\Sigma \times \mathbf{A} \times \Sigma) \times \mathrm{id}_{1/2}(\mathbf{Q}_2))$.
One has

$$\mathbf{Q}_2 \cong ((\Sigma \times \mathbf{A} \times \Sigma)^{<\omega} \times (\Sigma \times \wp(\mathbf{C}))) \cup (\Sigma \times \mathbf{A} \times \Sigma)^{\omega}.$$

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

**(3)** For $p \in \wp_{\rm nc}(\mathbf{Q}_2)$ and $\sigma \in \Sigma$,
$p\langle\sigma\rangle = \{q \in p : \exists\Gamma \in \wp(\mathbf{C})[q = (\langle\sigma, \Gamma\rangle)] \vee \exists a \in \mathbf{A}, \exists\sigma' \in \Sigma, \exists q' \in \mathbf{Q}_1[q = (\langle\sigma, a, \sigma'\rangle) \cdot q']\}$.

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

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

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

**(5)** Let $\mathbf{P}_2$, the domain of processes for $\mathcal{L}_2$, be given by
$\mathbf{P}_2 = \{p \in \wp_{\rm nc}(\mathbf{Q}_2) : p \text{ is uniformly nonempty}\}$.

**(6)** $\mathrm{act}(p, \sigma) = \{a \in \mathbf{A} : \exists\sigma'[p[\langle\sigma, a, \sigma\rangle] \neq \emptyset]\}$.

**(7)** For $\gamma \in \mathbf{C}$,

$$\overline{\gamma} = \begin{cases} c? & \text{if } \gamma = c!, \\ c! & \text{if } \gamma = c?. \end{cases}$$

For $\Gamma \in \wp(\mathbf{C})$,

$$\overline{\Gamma} = \{\overline{\gamma} : \gamma \in \Gamma\}. \quad \blacksquare$$

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

**Lemma 15** *The set $\mathbf{P}_2$ is closed in $\wp_{\rm nc}(\mathbf{Q}_2)$, and therefore, $\mathbf{P}_2$ is a complete metric space with the original metric of $\wp_{\rm nc}(\mathbf{Q}_2)$.* $\blacksquare$

**Proof.** This is proved in a similar fashion to Lemma 4. ■

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

**Definition 21** *(Interpretation $\mathcal{I}_2$ for Signature of $\mathcal{L}_2$)*

**(1)** $\mathrm{stop}_2 = \{(\langle\sigma,\Gamma\rangle) : \langle\sigma,\Gamma\rangle \in \Sigma \times \wp(\mathbf{C})\}$.

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

For $p \in \mathbf{P}_2$,
$\mathrm{asg}_2(x,e)(p) = \{(\langle\sigma,\tau,\sigma[[\![e]\!](\sigma)/x]\rangle)\cdot p : \sigma \in \Sigma\}$.

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

For $p \in \mathbf{P}_2$,
$\mathrm{output}(c,e)(p)$
$= \{(\langle\sigma,\langle c!,[\![e]\!](\sigma)\rangle,\sigma\rangle)\cdot p : \sigma \in \Sigma\} \cup \{(\langle\sigma,\Gamma\rangle) : \sigma \in \Sigma \wedge \Gamma \subseteq \mathbf{C}\setminus\{!c\}\}$.

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

For $p \in \mathbf{P}_2$,
$\mathrm{input}(c,x)(p)$
$= \{(\langle\sigma,c?v,\sigma[v/x]\rangle)\cdot p : \sigma \in \Sigma \wedge v \in \mathbf{V}\} \cup \{(\langle\sigma,\Gamma\rangle) : \sigma \in \Sigma \wedge \Gamma \subseteq \mathbf{C}\setminus\{c?\}\}$.

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

For $p_1, p_2 \in \mathbf{P}_2$,
$\mathrm{if}(b)(p_1,p_2) = \bigcup_{\sigma\in\Sigma}[\mathrm{if}([\![b]\!](\sigma) = \mathrm{tt}, p_1\langle\sigma\rangle, p_2\langle\sigma\rangle)]$.

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

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

The downward closedness of each $p$ follows from the fact that $p$ satisfies the condition named *disjointness deadlock condition* to be introduced in Definition 23. It is shown in Lemma 17 that the denotational meaning of each $s \in \mathcal{L}_2$ satisfies this condition.

**(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 $p_1, p_2 \in \mathbf{P}_2$,
$p_1 \,\tilde{\|}\, p_2 = p_1 \,\tilde{\lfloor\!\lfloor}\, p_2 \cup p_2 \,\tilde{\lfloor\!\lfloor}\, p_1 \cup p_1 \triangleright p_2 \cup p_2 \triangleright p_1 \cup (p_1 \# p_2)$.

Here

$$p_1 \,\tilde{\lfloor\!\lfloor}\, p_2 = \bigcup\{\langle\sigma,a,\sigma'\rangle\cdot(p_1[\langle\sigma,a,\sigma'\rangle] \,\tilde{\|}\, p_2) : \exists q \in \mathbf{Q}_2[\langle\sigma,a,\sigma'\rangle\cdot q \in p_1]\},$$

$$\begin{aligned} p_1 \triangleright p_2 \quad = \quad & (\bigcup\{(\langle\sigma,\tau,\sigma'\rangle)\cdot(p_1[\langle\sigma,c!v,\sigma\rangle] \,\tilde{\|}\, p_2[\langle\sigma,c?v,\sigma'\rangle]) : \\ & p_1[\langle\sigma,c!v,\sigma\rangle] \neq \emptyset \wedge p_2[\langle\sigma,c?v,\sigma'\rangle] \neq \emptyset\,\})^{\mathrm{cls}}, \end{aligned} \tag{30}$$

and

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

Remark that taking closure in the right-hand side of (30) is necessary as Example 4 shows below.

**(8)** Let

$$\mathcal{I}_2 = \{\text{stop}_2, \{\text{asg}_2(x,e) : \langle x,e\rangle \in \text{IVar}\times\text{VExp}\}, \{\text{if}(b) : b \in \text{BExp}\}, \tilde{+}, \tilde{\|}, \{\text{output}(c,e) : c \in \text{Chan} \wedge e \in \text{VExp}\}, \{\text{input}(c,x) : c \in \text{Chan} \wedge x \in \text{IVar}\}\}. \blacksquare$$

**Example 4** Let us assume, for simplicity, that IVar $= \{x\}$ and $\mathbf{V} = \{v\}$. Then the set of states consists only of one state denoted by $v$. Moreover assume that Chan $= \{c_i : i \in \omega\}$ and $c_i \neq c_j$ for $i \neq j$. Let $p_1$ and $p_2$ be defined as follows:
$p_1 = \{q_n : n \in \omega\}$,
$p_2 = \{(\langle v, c_n?v, v\rangle, \langle v, \emptyset\rangle) : n \in \omega\}$,
where

$$q_n = \langle v, c_n!v, v\rangle \cdot \underbrace{\langle v, c_0!v, v\rangle \cdots \langle v, c_0!v, v\rangle}_{n} \cdot \langle v, \emptyset\rangle.$$

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 23. Nevertheless, it is shown that the right-hand side of (30) without taking closure is not closed as follows. This set is $\{q'_n : n \in \omega\}$, where

$$q'_n = \langle v, \tau, v\rangle \cdot \underbrace{\langle v, c_0!v, v\rangle \cdots \langle v, c_0!v, v\rangle}_{n} \cdot \langle v, \emptyset\rangle.$$

This is not closed, since the infinite sequence

$$(\langle v, \tau, v\rangle, \langle v, c_0!v, v\rangle, \langle v, c_0!v, v\rangle, \cdots)$$

is a member of its closure but is not a member of it. ∎

The next lemma follows immediately from Definition 8 (7).

**Lemma 16** $\forall p_1, p_2 \in \mathbf{P}_2 [p_1 \,\tilde{\|}\, p_2 = p_2 \,\tilde{\|}\, p_1]$. ∎

In terms of the interpretation $\mathcal{I}_2$, the denotational model $\mathcal{D}_2$ is defined as follows:

**Definition 22** *(Denotational Model $\mathcal{D}_2$ for $\mathcal{L}_2$)*
The model $\mathcal{D}_2 : \mathcal{L}_2 \to \mathbf{P}_2$ is defined by induction on the structure of $s \in \mathcal{L}_2$ as in Definition 9.

**(1)** Let $D = \{\langle X, g_X\rangle\}_{X\in\text{RVar}}$ be the set of declarations.
Let $\mathbf{M}_2^{\mathcal{D}} = (\text{RVar} \to \mathbf{P}_2)$, and let $\Pi_2 : \mathbf{M}_2^{\mathcal{D}} \to \mathbf{M}_2^{\mathcal{D}}$ be defined as follows:

For $\mathbf{p} \in \mathbf{M}_2^{\mathcal{D}}$, and $X \in$ RVar,

$$\Pi_2(\mathbf{p})(X) = (\llbracket g_X \rrbracket^{\mathcal{I}_2}[(\mathbf{p}(Y_1^X), \cdots, \mathbf{p}(Y_{l(X)}^X))/(Y_1^X, \cdots, Y_{l(X)}^X)]),$$

where $\{Y_1^X, \cdots, Y_{l(X)}^X\}$ is the set of recursion variables contained in $g_X$. The mapping $\Pi_2$ is a contraction from $\mathbf{M}_2^{\mathcal{D}}$ to $\mathbf{M}_2^{\mathcal{D}}$.

Let $\mathbf{p}_0 = \text{fix}(\Pi_2)$. For $X \in$ RVar, let us define $X^{\mathcal{D}_2}$, the denotational meaning of $X$, by:
$X^{\mathcal{D}_2} = \mathbf{p}_0(X)$, $X \in$ RVar.

**(2)** For each operation $F$ of $\mathcal{L}_2$ with arity $r$, and $s_1, \cdots, s_r \in \mathcal{L}_2$, let
$\mathcal{D}_2\llbracket F(s_1, \cdots, s_r)\rrbracket = F^{\mathcal{I}_2}(\mathcal{D}_2\llbracket s_1\rrbracket, \cdots, \mathcal{D}_2\llbracket s_r\rrbracket)$,
where $F^{\mathcal{I}_2}$ is the interpreted operation corresponding to $F$. ∎

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

**Definition 23** *(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* $n$, notation: $\mathrm{IFin}_2^{(n)}(p)$ iff

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

The process $p$ is *image finite*, notation: $\mathrm{IFin}_2(p)$ iff

$$\forall n \in \omega[\mathrm{IFin}_2^{(n)}(p)].$$

**(2)** The process $p$ is *finite with respect to action sorts at level* $n$, notation: $\mathrm{ASFin}^{(n)}(p)$ iff

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

The process $p$ is *finite with respect to action sorts*, notation: $\mathrm{ASFin}(p)$ iff

$$\forall n \in \omega[\mathrm{ASFin}^{(n)}(p)].$$

**(3)** The process $p$ is *finite with respect to output values at level* $n$, notation: $\mathrm{OVFin}^{(n)}(p)$ iff

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

The process $p$ is *finite with respect to output values*, notation: $\mathrm{OVFin}(p)$ iff

$$\forall n \in \omega[\mathrm{OVFin}^{(n)}(p)].$$

**(4)** The process $p$ satisfies the *disjointness deadlock condition at level* $n$, notation: $\mathrm{DDC}^{(n)}(p)$ iff

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

The process $p$ satisfies the *disjointness deadlock condition*, notation: $\mathrm{DDC}(p)$ iff

$$\forall n \in \omega[\mathrm{DDC}^{(n)}(p)].$$

**(5)** $\mathbf{P}_2^* = \{p \in \mathbf{P}_2 : \mathrm{IFin}_2(p) \wedge \mathrm{ASFin}(p) \wedge \mathrm{OVFin}(p) \wedge \mathrm{DDC}(p)\ \}$. ∎

**Remark 4** Though the condition $\mathrm{DDC}(\cdot)$ might seem too complicated, it is characterized in terms of a simpler condition $\mathrm{D}(\cdot)$ defined as follows:

First, as follows from Definition 24 and Lemma 19 (2), for $s \in \mathcal{L}_2$, the refusals $\langle \sigma, \Gamma \rangle$ in $\mathcal{D}_2[\![s]\!]$ are characterized in terms of $\mathrm{act}(s, \sigma)$ by:

$$\forall \sigma[\exists \Gamma[\langle \sigma, \Gamma \rangle \in p] \Rightarrow \mathrm{sort}[\mathrm{act}(s, \sigma)] \subseteq \mathbf{C} \wedge \forall \Gamma[\langle \sigma, \Gamma \rangle \in \mathcal{D}_2[\![s]\!] \Leftrightarrow \Gamma \cap \mathrm{sort}[\mathrm{act}(s, \sigma)] = \emptyset]]. \quad (31)$$

Let us define $\mathrm{D}(p)$ for $p \in \mathbf{P}_2$ by:

$$\mathrm{D}(p) \Leftrightarrow \forall\sigma[\exists\Gamma[\langle\sigma,\Gamma\rangle \in p] \Rightarrow \exists R \subseteq \mathrm{sort}[\mathrm{act}(s,\sigma)] \cap \mathbf{C}, \forall\Gamma[\langle\sigma,\Gamma\rangle \in p \Leftrightarrow \Gamma \cap R = \emptyset]].$$

Then, for every $s \in \mathcal{L}_2$, one has $\mathrm{D}(\mathcal{D}_2[\![s]\!])$ by (31) putting $R = \mathrm{sort}[\mathrm{act}(s,\sigma)]$. It is immediate that $\{p \in \mathbf{P}_2 : \mathrm{DDC}^{(0)}(p)\}$ is characterized as the smallest set of those subsets of $\mathbf{P}_2$ which are closed under set-theoretical union and include $\{p \in \mathbf{P}_2 : \mathrm{D}(p)\}$. Also, $\{p \in \mathbf{P}_2 : \mathrm{DDC}(p)\}$ is characterized as the smallest set of those subsets of $\mathbf{P}_2$ which are *closed under taking remainders* and include $\{p \in \mathbf{P}_2 : \mathrm{DDC}^{(0)}(p)\}$, where closedness under taking remainders for subsets of $\mathbf{P}_2$ is defined as in Remark 2.

As stated in Definition 21, it is immediate that the downward closedness of $p \in \mathbf{P}_2$ follows from the fact that $\mathrm{DDC}(p)$. ∎

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

**Lemma 17**

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

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

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

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

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

**Proof.** These propositions are proved in a similar fashion to the proof of Lemma 6. Here we prove the essential part of **(3)**, i.e., that

$$\forall p_1, p_2 \in \mathbf{P}_2[\mathrm{DDC}(p_1) \wedge \mathrm{DDC}(p_2) \Rightarrow \mathrm{DDC}(p_1 \,\tilde{\|}\, p_2)]. \tag{32}$$

Let us show by induction on $n \in \omega$ that the following formula holds for every $n \in \omega$:

$$\forall p_1, p_2 \in \mathbf{P}_2[\mathrm{DDC}^{(n)}(p_1) \wedge \mathrm{DDC}^{(n)}(p_2) \Rightarrow \mathrm{DDC}^{(n)}(p_1 \,\tilde{\|}\, p_2)]]. \tag{33}$$

**Induction Base:** Let $p_1, p_2 \in \mathbf{P}_2$ such that $\mathrm{DDC}^{(0)}(p_1)$ and $\mathrm{DDC}^{(0)}(p_2)$, and fix $\sigma \in \Sigma$. By the definition of $\mathrm{DDC}^{(0)}(\cdot)$, there exists $\mathcal{R}_i \subseteq \wp(\mathrm{sort}[\mathrm{act}(p_i,\sigma)] \cap \mathbf{C})$ such that

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

Let $\mathcal{R} = \{R_1 \cup R_2 : R_1 \in \mathcal{R}_1 \wedge R_2 \in \mathcal{R}_2 \wedge R_1 \cap \overline{R_2} = \emptyset\}$. Then one has, by the definitions of $\tilde{\|}$ and $\#$, that

$$\forall\Gamma[\langle\sigma,\Gamma\rangle \in p_1 \,\tilde{\|}\, p_2 \Leftrightarrow \exists R \in \mathcal{R}[\Gamma \cap R = \emptyset]],$$

which implies that $\mathrm{DDC}^{(0)}(p_1 \,\tilde{\|}\, p_2)$.

**Induction Step:** For every $k \in \omega$, it is immediate by the definition of $\tilde{\|}$, that (33) with $n = k+1$ follows from (33) with $n = k$. ∎

## 4.4 Correctness of $\mathcal{D}_2$ with respect to $\mathcal{O}_2$

The correctness of $\mathcal{D}_2$ with respect to $\mathcal{O}_2$ is established as that of $\mathcal{D}_1$ with respect to $\mathcal{O}_1$, by means of an intermediate model $\tilde{\mathcal{O}}_2$.

### 4.4.1 Intermediate Model for $\mathcal{L}_2$ and Semantic Equivalence

First, the intermediate model $\tilde{\mathcal{O}}_2$, which is an alternative formulation of $\mathcal{D}_2$, is defined in terms of the transition $\longrightarrow_2$.

**Definition 24** *(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 $s \in \mathcal{L}_2$,

$$\begin{aligned}\tilde{\mathcal{O}}_2[\![s]\!] = \ & \bigcup_{\sigma\in\Sigma}[\bigcup\{(\langle\sigma,a,\sigma'\rangle)\cdot\tilde{\mathcal{O}}_2[\![s']\!] : \langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle\} \\ & \cup \mathrm{if}(\tau\notin\mathrm{act}(s,\sigma), \\ & \{(\langle\sigma,\Gamma\rangle) : \Gamma\cap\mathrm{sort}[\mathrm{act}(s,\sigma)] = \emptyset\},\emptyset)\ ].\ \blacksquare\end{aligned}$$

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

**Lemma 18** *(Distributivity of $\tilde{\|}$ in $\mathbf{P}_2$)*
*Let $k,l \geq 1$, $p_1,\cdots,p_k,p'_1,\cdots,p'_l \in \mathbf{P}_2^*$.*

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

**Proof.** See Appendix A.5. ■

By means of the above lemma, we establish the equivalence between $\mathcal{D}_2$ and $\tilde{\mathcal{O}}_2$ as we did in Lemma7.

**Lemma 19** *(Semantic Equivalence for $\mathcal{L}_2$)*

**(1)** *Let $F$ be an operation of $\mathcal{L}_2^*$ with arity $r$, and let $s_1,\cdots,s_r \in \mathcal{L}_2$. Then it holds that*
$\tilde{\mathcal{O}}_2[\![F(s_1,\cdots,s_r)]\!] = F^{\mathcal{I}_2}(\tilde{\mathcal{O}}_2[\![s_1]\!],\cdots,\tilde{\mathcal{O}}_2[\![s_r]\!])$.

**(2)** *For $s\in\mathcal{L}_2$, it holds that*
$\tilde{\mathcal{O}}_2[\![s]\!] = \mathcal{D}_2[\![s]\!]$. ■

**Proof.** **(1)** The proof is similar to that of Lemma 7. Here we prove the claim for the operation $\|$. For the other operations this is proved 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 $s_1, s_2 \in \mathcal{L}_2$,

$$F(s_1,s_2) = \tilde{\mathcal{O}}_2[\![s_1 \parallel s_2]\!],$$

$$G(s_1,s_2) = \tilde{\mathcal{O}}_2[\![s_1]\!]\ \tilde{\|}\ \tilde{\mathcal{O}}_2[\![s_2]\!].$$

Let $\mathcal{F} : \mathbf{H}_2 \to \mathbf{H}_2$ be defined as follows:

For $f\in\mathbf{H}_2$ and $s_1,s_2\in\mathcal{L}_2$,

$$\begin{aligned}\mathcal{F}(f)(s_1,s_2) = \ & \bigcup\{(\langle\sigma,a,\sigma'\rangle)\cdot f(s'_1,s_2) : \langle s_1,\sigma\rangle \xrightarrow{a}_2 \langle s'_1,\sigma'\rangle\} \\ & \cup\bigcup\{(\langle\sigma,a,\sigma'\rangle)\cdot f(s_1,s'_2) : \langle s_2,\sigma\rangle \xrightarrow{a}_2 \langle s'_2,\sigma'\rangle\} \\ & \cup\{(\langle\sigma,\tau,\sigma'\rangle)\cdot f(s'_1,s'_2) : \\ & \quad \exists c\in\mathrm{Chan}, \exists v\in\mathbf{V} \\ & \quad [\langle s_1,\sigma\rangle \xrightarrow{c!v}_2 \langle s'_1,\sigma\rangle \wedge \langle s_2,\sigma\rangle \xrightarrow{c?v}_2 \langle s'_2,\sigma'\rangle\ ]\} \\ & \cup\{(\langle\sigma,\tau,\sigma'\rangle)\cdot f(s'_1,s'_2) : \\ & \quad \exists c\in\mathrm{Chan}, \exists v\in\mathbf{V} \\ & \quad [\langle s_1,\sigma\rangle \xrightarrow{c?v}_2 \langle s'_1,\sigma\rangle \wedge \langle s_2,\sigma\rangle \xrightarrow{c!v}_2 \langle s'_2,\sigma'\rangle\ ]\} \\ & \cup\{(\langle\sigma,\Gamma\rangle) : \\ & \quad \tau\notin\mathrm{act}(s_1,\sigma)\wedge\tau\notin\mathrm{act}(s_1,\sigma) \\ & \quad \wedge\mathrm{sort}[\mathrm{act}(s_1,\sigma)]\cap\overline{\mathrm{sort}[\mathrm{act}(s_2,\sigma)]} = \emptyset\wedge \\ & \quad \Gamma\cap(\mathrm{sort}[\mathrm{act}(s_1,\sigma)]\cup\mathrm{sort}[\mathrm{act}(s_2,\sigma)]) = \emptyset\ \}.\end{aligned}$$

The mapping $\mathcal{F}$ is a contraction.

Let $s_1, s_2 \in \mathcal{L}_2$. By the definitions of $\tilde{\mathcal{O}}_2$ and $\rightarrow_2$, it holds that

$$F(s_1, s_2) = \mathcal{F}(F)(s_1, s_2).$$

That is, $F = \mathrm{fix}(\mathcal{F})$.

On the other hand,

$$\begin{aligned} G(s_1, s_2) = \ & (\tilde{\mathcal{O}}_2[\![s_1]\!] \,\lfloor\!\lfloor\, \tilde{\mathcal{O}}_2[\![s_2]\!]) \cup (\tilde{\mathcal{O}}_2[\![s_2]\!] \,\lfloor\!\lfloor\, \tilde{\mathcal{O}}_2[\![s_1]\!]) \\ & \cup (\tilde{\mathcal{O}}_2[\![s_1]\!] \triangleright \tilde{\mathcal{O}}_2[\![s_2]\!]) \cup (\tilde{\mathcal{O}}_2[\![s_2]\!] \triangleright \tilde{\mathcal{O}}_2[\![s_1]\!]) \cup (\tilde{\mathcal{O}}_2[\![s_1]\!] \,\#\, \tilde{\mathcal{O}}_2[\![s_2]\!]). \end{aligned}$$

First,

$$\begin{aligned} & \tilde{\mathcal{O}}_2[\![s_1]\!] \,\lfloor\!\lfloor\, \tilde{\mathcal{O}}_2[\![s_2]\!] \\ & = \bigcup\{(\langle\sigma, a, \sigma'\rangle) \cdot (\tilde{\mathcal{O}}_2[\![s_1]\!][\langle\sigma, a, \sigma'\rangle] \,\tilde{\|}\, \tilde{\mathcal{O}}_2[\![s_2]\!]) : \tilde{\mathcal{O}}_2[\![s_1]\!][\langle\sigma, a, \sigma'\rangle] \neq \emptyset\} \\ & = \bigcup\{(\langle\sigma, a, \sigma'\rangle) \cdot (\bigcup\{\tilde{\mathcal{O}}_2[\![s_1']\!] : \langle s_1, \sigma\rangle \xrightarrow{a}_2 \langle s_1', \sigma'\rangle\} \,\tilde{\|}\, \tilde{\mathcal{O}}_2[\![s_2]\!]) : \\ & \qquad \exists s_1'[\langle s_1, \sigma\rangle \xrightarrow{a}_2 \langle s_1', \sigma'\rangle]\} \\ & = \bigcup\{(\langle\sigma, a, \sigma'\rangle) \cdot \bigcup\{\tilde{\mathcal{O}}_2[\![s_1']\!] \,\tilde{\|}\, \tilde{\mathcal{O}}_2[\![s_2]\!] : \langle s_1, \sigma\rangle \xrightarrow{a}_2 \langle s_1', \sigma'\rangle\} : \\ & \qquad \exists s_1'[\langle s_1, \sigma\rangle \xrightarrow{a}_2 \langle s_1', \sigma'\rangle]\} \\ & \qquad \text{(by Lemma 18)} \\ & = \bigcup\{(\langle\sigma, a, \sigma'\rangle) \cdot (\tilde{\mathcal{O}}_2[\![s_1']\!] \,\tilde{\|}\, \tilde{\mathcal{O}}_2[\![s_2]\!]) : \langle s_1, \sigma\rangle \xrightarrow{a}_2 \langle s_1', \sigma'\rangle\}. \end{aligned}$$

Hence

$$\begin{aligned} & \tilde{\mathcal{O}}_2[\![s_1]\!] \,\lfloor\!\lfloor\, \tilde{\mathcal{O}}_2[\![s_2]\!] \\ & = \bigcup\{(\langle\sigma, a, \sigma'\rangle) \cdot G(s_1', s_2) : \langle s_1, \sigma\rangle \xrightarrow{a}_2 \langle s_1', \sigma'\rangle\}. \end{aligned} \tag{34}$$

Similarly

$$\begin{aligned} & \tilde{\mathcal{O}}_2[\![s_2]\!] \,\lfloor\!\lfloor\, \tilde{\mathcal{O}}_2[\![s_1]\!] \\ & = \bigcup\{(\langle\sigma, a, \sigma'\rangle) \cdot G(s_1, s_2') : \langle s_2, \sigma\rangle \xrightarrow{a}_2 \langle s_2', \sigma'\rangle\}. \end{aligned} \tag{35}$$

Next,

$$\begin{aligned} & \tilde{\mathcal{O}}_2[\![s_1]\!] \triangleright \tilde{\mathcal{O}}_2[\![s_2]\!] \\ & = \bigcup\{(\langle\sigma, \tau, \sigma'\rangle) \cdot (\tilde{\mathcal{O}}_2[\![s_1]\!][\langle\sigma, c!v, \sigma\rangle] \,\tilde{\|}\, \tilde{\mathcal{O}}_2[\![s_2]\!][\langle\sigma, c?v, \sigma'\rangle]) : \\ & \qquad \tilde{\mathcal{O}}_2[\![s_1]\!][\langle\sigma, c!v, \sigma\rangle] \neq \emptyset \wedge \tilde{\mathcal{O}}_2[\![s_2]\!][\langle\sigma, c?v, \sigma'\rangle] \neq \emptyset\} \\ & \qquad \text{(Taking closure is omitted, since} \\ & \qquad \mathrm{ASFin}^{(0)}(\mathcal{O}_2[\![s_i]\!]) \text{ and } \mathrm{OVFin}^{(0)}(\mathcal{O}_2[\![s_i]\!]) \ (i = 1, 2) \text{ by Lemma 14 (2), (3))} \\ & = \bigcup\{(\langle\sigma, \tau, \sigma'\rangle) \cdot (\bigcup\{\tilde{\mathcal{O}}_2[\![s_1']\!] : \langle s_1, \sigma\rangle \xrightarrow{c!v}_2 \langle s_1', \sigma\rangle\} \,\tilde{\|}\, \bigcup\{\tilde{\mathcal{O}}_2[\![s_2']\!] : \langle s_2, \sigma\rangle \xrightarrow{c?v}_2 \langle s_2', \sigma'\rangle\}) : \\ & \qquad \exists s_1'[\langle s_1, \sigma\rangle \xrightarrow{c!v}_2 \langle s_1', \sigma\rangle] \wedge \exists s_2'[\langle s_2, \sigma\rangle \xrightarrow{c?v}_2 \langle s_2', \sigma'\rangle]\} \\ & = \bigcup\{(\langle\sigma, \tau, \sigma'\rangle) \cdot (\bigcup\{\tilde{\mathcal{O}}_2[\![s_1']\!] \,\tilde{\|}\, \tilde{\mathcal{O}}_2[\![s_2']\!] : \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\}) : \\ & \qquad \exists s_1'[\langle s_1, \sigma\rangle \xrightarrow{c!v}_2 \langle s_1', \sigma\rangle] \wedge \exists s_2'[\langle s_2, \sigma\rangle \xrightarrow{c?v}_2 \langle s_2', \sigma'\rangle]\} \\ & \qquad \text{(by Lemma 18)} \\ & = \bigcup\{(\langle\sigma, \tau, \sigma'\rangle) \cdot (\tilde{\mathcal{O}}_2[\![s_1']\!] \,\tilde{\|}\, \tilde{\mathcal{O}}_2[\![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\ ]\}. \end{aligned}$$

Hence

$$\begin{aligned} & \tilde{\mathcal{O}}_2[\![s_1]\!] \triangleright \tilde{\mathcal{O}}_2[\![s_2]\!] \\ & = \bigcup\{(\langle\sigma, \tau, \sigma'\rangle) \cdot G(s_1', s_2') : \exists c, \exists v[\langle s_1, \sigma\rangle \xrightarrow{c!v}_2 \langle s_1', \sigma\rangle \wedge \langle s_2, \sigma\rangle \xrightarrow{c?v}_2 \langle s_2', \sigma'\rangle\ ]\}. \end{aligned} \tag{36}$$

Similarly

$$\begin{aligned} & \tilde{\mathcal{O}}_2[\![s_2]\!] \triangleright \tilde{\mathcal{O}}_2[\![s_1]\!] \\ & = \bigcup\{(\langle\sigma, \tau, \sigma'\rangle) \cdot G(s_1', s_2') : \exists c, \exists v[\langle s_1, \sigma\rangle \xrightarrow{c?v}_2 \langle s_1', \sigma'\rangle \wedge \langle s_2, \sigma\rangle \xrightarrow{c!v}_2 \langle s_2', \sigma\rangle\ ]\}. \end{aligned} \tag{37}$$

Then let us show that the following holds for every $\langle\sigma, \Gamma\rangle \in \Sigma \times \wp(\mathbf{C})$.

$$
\begin{aligned}
&\langle \sigma, \Gamma \rangle \in \tilde{\mathcal{O}}_2[\![s_1]\!] \# \tilde{\mathcal{O}}_2[\![s_2]\!] \\
&\quad \Leftrightarrow \langle \sigma, \Gamma \rangle \in \{(\langle \sigma, \Gamma \rangle) : \\
&\qquad \tau \notin \mathrm{act}(s_1, \sigma) \wedge \tau \notin \mathrm{act}(s_2, \sigma) \\
&\qquad \wedge \mathrm{sort}[\mathrm{act}(s_1, \sigma)] \cap \overline{\mathrm{sort}[\mathrm{act}(s_2, \sigma)]} = \emptyset \\
&\qquad \wedge \Gamma \cap (\mathrm{sort}[\mathrm{act}(s_1, \sigma)] \cup \mathrm{sort}[\mathrm{act}(s_2, \sigma)]) = \emptyset \ \}.
\end{aligned}
\tag{38}
$$

($\Rightarrow$) Suppose $\langle \sigma, \Gamma \rangle \in \tilde{\mathcal{O}}_2[\![s_1]\!] \# \tilde{\mathcal{O}}_2[\![s_2]\!]$. Then by the definition of $\#$ , one has

$$
\begin{aligned}
&\exists \langle \sigma, \Gamma_1 \rangle \in \tilde{\mathcal{O}}_2[\![s_1]\!], \exists \langle \sigma, \Gamma_2 \rangle \in \tilde{\mathcal{O}}_2[\![s_2]\!] \\
&[(\mathbf{C} \setminus \Gamma_1) \cap \overline{(\mathbf{C} \setminus \Gamma_2)} = \emptyset \wedge \Gamma \subseteq \Gamma_1 \cap \Gamma_2].
\end{aligned}
\tag{39}
$$

Fix such $\Gamma_1$ and $\Gamma_2$. By the definition of $\tilde{\mathcal{O}}_2$,

$$\tau \notin \mathrm{act}(s_1, \sigma). \tag{40}$$

Moreover,

$$\Gamma_1 \cap \mathrm{sort}[\mathrm{act}(s_1, \sigma)] = \emptyset,$$

and therefore,

$$\mathrm{sort}[\mathrm{act}(s_1, \sigma)] \subseteq \mathbf{C} \setminus \Gamma_1. \tag{41}$$

Similarly

$$\tau \notin \mathrm{act}(s_2, \sigma), \tag{42}$$

and

$$\mathrm{sort}[\mathrm{act}(s_2, \sigma)] \subseteq \mathbf{C} \setminus \Gamma_2,$$

that is,

$$\overline{\mathrm{sort}[\mathrm{act}(s_2, \sigma)]} \subseteq \overline{\mathbf{C} \setminus \Gamma_2}. \tag{43}$$

By (39), (41), and (43), one has

$$\mathrm{sort}[\mathrm{act}(s_1, \sigma)] \cap \overline{\mathrm{sort}[\mathrm{act}(s_2, \sigma)]} \subseteq \mathbf{C} \setminus \Gamma_1 \cap \overline{\mathbf{C} \setminus \Gamma_2} = \emptyset. \tag{44}$$

By (39),

$$\Gamma \subseteq \Gamma_1 \subseteq \mathbf{C} \setminus \mathrm{sort}[\mathrm{act}(s_1, \sigma)],$$

and therefore,

$$\Gamma \cap \mathrm{sort}[\mathrm{act}(s_1, \sigma)] = \emptyset. \tag{45}$$

Similarly

$$\Gamma \cap \mathrm{sort}[\mathrm{act}(s_2, \sigma)] = \emptyset. \tag{46}$$

By (40), (42), (44), (45), and (46), one has the right-hand side of (38).
($\Leftarrow$) This part is obtained by putting

$$\Gamma_1 = \mathbf{C} \setminus \mathrm{sort}[\mathrm{act}(s_1, \sigma)]$$

and

$$\Gamma_2 = \mathbf{C} \setminus \mathrm{sort}[\mathrm{act}(s_2, \sigma)].$$

Thus one has (38).
By (34), (35), (36), (37), and (38), one has

$G(s_1, s_2) = \mathcal{F}(G)(s_1, s_2)$.

That is, $G$ is also the fixed point of $\mathcal{F}$. Thus one has $F = G$. That is,

$$\forall s_1, s_2 \in \mathcal{L}_2[\tilde{\mathcal{O}}_2[\![s_1 \parallel s_2]\!] = \tilde{\mathcal{O}}_2[\![s_1]\!] \,\tilde{\parallel}\, \tilde{\mathcal{O}}_2[\![s_2]\!]].$$

**(2)** First, let us show, for $X \in$ RVar, that

$$\tilde{\mathcal{O}}_2[\![X]\!] = \mathcal{D}_2[\![X]\!] \tag{47}$$

Let $\langle X, g_X\rangle \in D$. Then,

$$\begin{aligned} &\tilde{\mathcal{O}}_2[\![X]\!] \\ &= \tilde{\mathcal{O}}_2[\![g_X]\!] \quad \text{(by the definition of } \tilde{\mathcal{O}}_2) \\ &= [\![g_X]\!]^{\mathcal{I}_2}[(\tilde{\mathcal{O}}_2[\![Y_1^X]\!], \cdots, \tilde{\mathcal{O}}_2[\![Y_{l(X)}^X]\!])/(Y_1^X, \cdots, Y_{l(X)}^X)] \quad \text{(by (1))}, \end{aligned} \tag{48}$$

where $\{Y_1^X, \cdots, Y_{l(X)}^X\}$ is the set of recursion variables contained in $g_X$.

Hence $(\lambda X \in \text{RVar} : \tilde{\mathcal{O}}_2[\![X]\!])$ is the fixed point of $\Pi_2$ defined in Definition 22. Thus by the definition of $\mathcal{D}_2[\![X]\!]$ one has (47). Next by induction on the structure of $s \in \mathcal{L}_2$ and (1), one has $\forall s \in \mathcal{L}_2[\tilde{\mathcal{O}}_2[\![s]\!] = \mathcal{D}_2[\![s]\!]]$. ∎

### 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*
$\alpha_2 : \mathbf{P}_2 \to (\Sigma \to \wp_{\text{nc}}((\mathbf{A} \times \Sigma)^{\leq\omega}))$
is defined as follows. As $\alpha_1$, this function is formulated in two ways, i.e., firstly as the fixed point of a higher-order mapping, and secondly as the set of histories.

**Definition 25** *(Abstraction Function $\alpha_2$ for $\mathcal{L}_2$)*
We have the unique mapping $\alpha_2 : \mathbf{P}_2 \to (\Sigma \to \wp_{\text{nc}}((\mathbf{A} \times \Sigma)^{\leq\omega}))$ satisfying the following equation; the existence and uniqueness of such a mapping are obtained as in Definition 12.

For every $p \in \mathbf{P}_2$, $\sigma \in \Sigma$,

$$\begin{aligned} \alpha_2(p)(\sigma) = \; & \bigcup\{(\langle a, \sigma'\rangle) \cdot \alpha_2(p[\langle \sigma, a, \sigma'\rangle])(\sigma') : \exists q \in \mathbf{Q}_2[(\langle \sigma, a, \sigma'\rangle) \cdot q \in p]\} \\ & \cup \text{if}(\exists \Gamma \in \mathbf{C}[(\langle \sigma, \Gamma\rangle) \in p], \{\epsilon\}, \emptyset). \end{aligned}$$ ∎

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

**Definition 26** *(Histories of Elements of $\mathbf{P}_2$)*
Let $q \in \mathbf{Q}_2 \cup (\Sigma \times \mathbf{A} \times \Sigma)^+$.

**(1)** The sequence $q$ is *executable*, notation: $\text{Exec}_2(q)$ iff

$$\begin{aligned} &\exists (\langle \sigma_i, a_i, \sigma_i'\rangle)_{i\in I}[q = (\langle \sigma_i, \sigma_i'\rangle)_{i\in I} \wedge \forall i \in I[i > 0 \Rightarrow \sigma_{i-1}' = \sigma_i]] \vee \\ &\exists k \in \omega, \exists (\langle \sigma_i, a_i, \sigma_i'\rangle)_{i\in \bar{k}}, \exists \sigma_{k+1}, \exists \Gamma[q = (\langle \sigma_i, \sigma_i'\rangle)_{i\in \bar{k}} \cdot (\langle \sigma_{k+1}, \Gamma\rangle) \wedge \forall i \in \bar{k}[\sigma_i' = \sigma_{i+1}]\;], \end{aligned}$$

where $I$ is $\omega$ or $\bar{n}$ for some $n \in \omega$.

**(2)** Let $q$ be executable. The *history* of $q$, notation: $\text{hist}_2(q)$, is given by

$$\text{hist}_2(q) = \begin{cases} (\langle a_i, \sigma_i'\rangle)_{i\in\omega} & \text{if } q = (\langle \sigma_i, a_i, \sigma_i'\rangle)_{i\in\omega}, \\ (\langle a_i, \sigma_i'\rangle)_{i\in\bar{k}} & \text{if } q = (\langle \sigma_i, a_i, \sigma_i'\rangle)_{i\in\bar{k}} \cdot (\langle \sigma_{k+1}, \Gamma\rangle). \end{cases}$$

**(3)** Let $q \in \mathbf{Q}_2$.

$$\text{istate}_2(q) = \begin{cases} \sigma & \text{if } q = (\langle \sigma, a, \sigma'\rangle) \cdot q', \\ \sigma & \text{if } \exists \Gamma[q = (\langle \sigma, \Gamma\rangle). \end{cases}$$ ∎

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

**Lemma 20** *(Another Formulation of Abstraction Function $\alpha_2$)*

**(1)** *For $p \in \mathbf{P}_2, \sigma \in \Sigma$, it holds that*
$\alpha_2(p)(\sigma) = \{\mathrm{hist}_2(q) : q \in p \wedge \mathrm{Exec}_2(q) \wedge \mathrm{istate}_2(q) = \sigma\}$.

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

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

**Lemma 21** *(Correctness of $\mathcal{D}_2$)*

**(1)** $\alpha_2 \circ \tilde{\mathcal{O}}_2 = \mathcal{O}_2$.

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

**Proof.** **(1)** Let us define $\mathcal{O}_2' : \mathcal{L}_2 \to (\Sigma \to \wp_{\mathrm{nc}}((\mathbf{A} \times \Sigma)^{\leq\omega}))$ by $\mathcal{O}_2' = \alpha_2 \circ \tilde{\mathcal{O}}_2$.
Fix $s \in \mathcal{L}_2$ and $\sigma \in \Sigma$. Let us show

$$\mathcal{O}_2'[\![s]\!](\sigma) = \Psi_2^{\mathcal{O}}(\mathcal{O}_2')(s)(\sigma), \tag{49}$$

where $\Psi_2^{\mathcal{O}}$ is the higher-order function introduced in Lemma 19.

$$\begin{aligned}
&\mathcal{O}_2'[\![s]\!](\sigma)\\
&= \bigcup\{(\langle a,\sigma'\rangle)\cdot\alpha_2(\tilde{\mathcal{O}}_2[\![s]\!][(\sigma,a,\sigma')])(\sigma') : \tilde{\mathcal{O}}_2[\![s]\!][(\sigma,a,\sigma')] \neq \emptyset\}\\
&\qquad \cup \mathrm{if}(\tau\notin\mathrm{act}(s,\sigma), \{(\langle\sigma,\Gamma\rangle) : \Gamma \cap \overline{\mathrm{sort}[\mathrm{act}(s,\sigma)]} = \emptyset\}, \emptyset)\\
&= \bigcup\{(\langle a,\sigma'\rangle)\cdot\alpha_2(\bigcup\{\tilde{\mathcal{O}}_2[\![s']\!] : \langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle\})(\sigma') :\\
&\qquad \exists s'[\langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle]\}\\
&\qquad \cup \mathrm{if}(\tau\notin\mathrm{act}(s,\sigma), \{(\langle\sigma,\Gamma\rangle) : \Gamma \cap \overline{\mathrm{sort}[\mathrm{act}(s,\sigma)]} = \emptyset\}, \emptyset)\\
&= \bigcup\{(\langle a,\sigma'\rangle)\cdot\bigcup\{\alpha_2(\tilde{\mathcal{O}}_2[\![s']\!])(\sigma') : \langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle\} :\\
&\qquad \exists s'[\langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle]\}\\
&\qquad \cup \mathrm{if}(\tau\notin\mathrm{act}(s,\sigma), \{(\langle\sigma,\Gamma\rangle) : \Gamma \cap \overline{\mathrm{sort}[\mathrm{act}(s,\sigma)]} = \emptyset\}, \emptyset)\\
&\qquad\ (\text{by Lemma 20 (2)})\\
&= \bigcup\{(\langle a,\sigma'\rangle)\cdot\alpha_2(\tilde{\mathcal{O}}_2[\![s']\!])(\sigma') : \langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle\}\\
&\qquad \cup \mathrm{if}(\tau\notin\mathrm{act}(s,\sigma), \{(\langle\sigma,\Gamma\rangle) : \Gamma \cap \overline{\mathrm{sort}[\mathrm{act}(s,\sigma)]} = \emptyset\}, \emptyset)\\
&= \bigcup\{(\langle a,\sigma'\rangle)\cdot\mathcal{O}_2'[\![s']\!](\sigma') : \langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle\}.\\
&\qquad \cup \mathrm{if}(\tau\notin\mathrm{act}(s,\sigma), \{(\langle\sigma,\Gamma\rangle) : \Gamma \cap \overline{\mathrm{sort}[\mathrm{act}(s,\sigma)]} = \emptyset\}, \emptyset)
\end{aligned}$$

Thus by the definition of $\Psi_2^{\mathcal{O}}$, (49) holds for every $s$ and $\sigma$. This implies that

$$\alpha_2 \circ \tilde{\mathcal{O}}_2 = \mathcal{O}_2' = \mathrm{fix}(\Psi_2^{\mathcal{O}}) = \mathcal{O}_2.$$

**(2)** This part follows from **(1)** and Lemma 19 (2). ∎

## 4.5 Full Abstractness of $\mathcal{D}_2$ with respect to $\mathcal{O}_2$

As for $\mathcal{L}_1$, we present the following lemma to establish the full abstractness of $\mathcal{D}_2$. Like Lemma 12, the claim of this lemma is stated roughly as follows:

> if $p_1 \neq p_2$, then $p_1$ and $p_2$ are *uniformly distinct*, i.e., then one has
> $\forall\sigma, \exists T[\alpha(p_1 \parallel \mathcal{D}_2[\![T]\!])(\sigma) \neq \alpha(p_2 \parallel \mathcal{D}_2[\![T]\!])(\sigma)]$,
> where $T$ is a tester.

**Lemma 22** *(Uniform Distinction Lemma for $\mathcal{L}_2$)*

**(1)** *Let* $r \in (\Sigma \times \Sigma)^+$, *and* $p_1, p_2 \in \mathbf{P}_2^*$.

*If* $p_1[r] \neq \emptyset$ *and* $p_2[r] = \emptyset$, *then it holds that*
$\forall \sigma \in \Sigma, \exists T \in \mathcal{L}_2[\alpha_2(p_1 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma) \setminus \alpha_2(p_2 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma) \neq \emptyset]$.

**(2)** *Let* $q \in (\Sigma \times \Sigma)^{<\omega} \times \Sigma$, *and* $p_1, p_2 \in \mathbf{P}_2^*$.

*If* $q \in p_1 \setminus p_2$, *then it holds that*
$\forall \sigma \in \Sigma, \exists T \in \mathcal{L}_2[\alpha_2(p_1 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma) \setminus \alpha_2(p_2 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma) \neq \emptyset]$. ∎

The proof of this lemma is given later. First, note that the full abstractness of $\mathcal{D}_2$ is derived from it in the same way as Theorem 1 was derived from Lemma 12.

**Theorem 2** *(Full Abstractness of* $\mathcal{D}_2$*)*
$\forall p_1, p_2 \in \mathcal{D}_2[\mathcal{L}_2][p_1 \neq p_2 \Rightarrow \exists p \in \mathcal{D}_2[\mathcal{L}_2][\alpha_2(p_1 \tilde{\|} p) \neq \alpha_2(p_2 \tilde{\|} p)]]$. ∎

We present the following lemma as a preliminary to the proof of Lemma 22. For its proof we assume that $\mathbf{V}$ is infinite. As in Section 3.5 we assume IVar $= \{x\}$ for simplicity. The *format of testers* $F(\cdot,\cdot,\cdot,\cdot,\cdot)$ is defined in $\mathcal{L}_2$ as in $\mathcal{L}_1$ (cf. Definition 15). We have the following lemma corresponding to Lemma 13.

**Lemma 23** *(Testing Lemma for* $\mathcal{L}_2$*)*
*For a process* $p \in \mathbf{P}_2^*$, *a step* $\langle\sigma', a, \sigma''\rangle \in (\Sigma \times \mathbf{A} \times \Sigma)$, *and a state* $\sigma_0 \in \Sigma$, *there are two finite sequences* $r_1, r_2 \in (\Sigma \times \mathbf{A} \times \Sigma)^{<\omega}$ *such that the following hold:*

**(i)** *The sequence* $r_1 \cdot \langle\sigma', \sigma''\rangle \cdot r_2$ *is executable and its initial state is* $\sigma_0$.

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

- **(a)** $\mathcal{D}_2[\![T]\!][r_1 \cdot r_2] = \mathcal{D}_2[\![T']\!]$,
- **(b)** *The process* $p$ *is* forced *to execute the step* $\langle\sigma', a, \sigma''\rangle$ *and forbidden to execute any other steps, when the parallel composition* $p \tilde{\|} \mathcal{D}_2[\![T]\!]$ *executes the sequence:* $r_1 \cdot \langle\sigma', a, \sigma''\rangle \cdot r_2$. *That is, for every* $q' \in \mathbf{Q}_2$, *the following holds:*

$$p[\langle\sigma', a, \sigma''\rangle] \neq \emptyset \wedge q' \in p[\langle\sigma', a, \sigma''\rangle] \tilde{\|} \mathcal{D}_2[\![T']\!] \Leftrightarrow$$
$$r_1 \cdot \langle\sigma', \sigma''\rangle \cdot r_2 \cdot q' \in p \tilde{\|} \mathcal{D}_2[\![T]\!]. \ ∎$$

**Proof.** The proof is formulated by supposing that IVar is reduced to one variable: IVar $= \{x\}$, as Lemma 13. However, the lemma still holds when IVar is composed of more than one variable, as Lemma 13.

Under this assumption $\Sigma$ is identified with $\mathbf{V}$ and the states $\sigma'$, $\sigma''$, and $\sigma_0$ are represented by $v', v'', v_0 \in \mathbf{V}$, respectively.

The proof is given by distinguishing two cases according to whether $v_0 = v'$.

**Case 1.** When $v_0 = v'$, we can easily construct two sequences $r_1$, $r_2$ satisfying **(i)**, **(ii)** in Lemma 23 as follows:

$$r_1 = \epsilon,$$
$$r_2 = \langle v'', \tau, v_1\rangle,$$

where $v_1$ is chosen such that

$$\begin{cases} \text{(i) } v_1 \neq v'', \\ \text{(ii) } v_1 \neq \{v \in \mathbf{V} : \langle v', a, v''\rangle \cdot \langle v'', \tau, v\rangle \in p^{[2]}\}. \end{cases} \tag{50}$$

Remark that the right-hand side of (50) (ii) is finite by Lemma 17 (4), and therefore there is $v_1$ satisfying (50).

It is immediate that **(i)** holds. It is shown that **(ii)** holds in a similar fashion to the corresponding part in the proof Lemma 13.

**Case 2.** When $v_0 \neq v'$, we can construct two sequences $r_1$, $r_2$, satisfying **(i)** and **(ii)** in Lemma 23 as follows:

$$r_1 = \langle v_0, \tau, v' \rangle,$$
$$r_2 = \langle v'', \tau, v_1 \rangle,$$

where $v_1$ is chosen such that

$$\begin{cases} \text{(i) } v_1 \neq \{v \in \mathbf{V} : \langle v_0, \tau, v'' \rangle \cdot \langle v', a, v'' \rangle \cdot \langle v'', \tau, v \rangle \in p^{[3]}\}, \\ \text{(ii) } v_1 \neq v', \\ \text{(iii) } v_1 \neq v'', \\ \text{(iv) } v_1 \neq \{v \in \mathbf{V} : \langle v', a, v'' \rangle \cdot \langle v'', \tau, v \rangle \in p^{[2]}\}. \end{cases} \tag{51}$$

Remark that the right-hand sides of (51) (i) and (iv) are finite by Lemma 17 (4), and therefore there is $v_1$ satisfying (51).

In this case also, it is immediate that **(i)** holds. Let us show **(ii)**. For every $T' \in \mathcal{L}_2$, let

$$T \equiv F(v', v_0, v_1, v_2, T'), \tag{52}$$

where $v_2$ is chosen such that

$$\begin{cases} \text{(i) } v_2 \neq v'', \\ \text{(ii) } v_2 \neq v_1. \end{cases} \tag{53}$$

It is shown that **(ii)** holds in a similar fashion to the corresponding part in the proof Lemma 13. ■

The following proposition follows immediately from Lemma 23 as Corollary 1 followed from Lemma 13; this corollary is to play a central role in the proof of Lemma 22.

**Corollary 2** *Let $p \in \mathbf{P}_2^*$, $\langle \sigma', a, \sigma'' \rangle \in (\Sigma \times \mathbf{A} \times \Sigma)$, and $\sigma_0 \in \Sigma$.*

*There are two finite sequences $\rho_1, \rho_2 \in (\mathbf{A} \times \Sigma)^{<\omega}$ such that for every tester $T' \in \mathcal{L}_2$ there exists another tester $T \in \mathcal{L}_2$ such that the following holds:*

*Let $\sigma_1$ be the last element of the finite sequence $\rho_1 \cdot \sigma'' \cdot \rho_2$. Then the following holds for every $\rho' \in (\mathbf{A} \times \Sigma)^{\leq\omega}$:*

$$p[\langle \sigma', \sigma'' \rangle] \neq \emptyset \wedge \rho' \in \alpha_2(p[\langle \sigma', \sigma'' \rangle] \tilde{\|} \mathcal{D}_2[\![T']\!])(\sigma_1) \Leftrightarrow$$
$$\rho_1 \cdot \sigma'' \cdot \rho_2 \cdot \rho' \in \alpha_2(p \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma_0). \quad ■$$

**Proof of Lemma 22.**

**(1)** The first part is proved by means of Corollary 2, as Lemma 12 (1) was proved by means of Lemma 13, by induction on the length of $r \in (\Sigma \times \mathbf{A} \times \Sigma)^+$.

**(2)** Let us establish the second part. For $q \in \mathbf{Q}_2$, let

$$\Phi_2(q) \Leftrightarrow$$
$$\forall p_1, p_2 \in \mathbf{P}_2^*[q \in p_1 \setminus p_2 \Rightarrow$$
$$\forall \sigma \in \Sigma, \exists T \in \mathcal{L}_2[\alpha_2(p_1 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma) \setminus \alpha_2(p_2 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma) \neq \emptyset]\ ].$$

Let us prove by induction on the length of $q \in \mathbf{Q}_2$, that $\Phi_2(q)$ holds for every $q \in \mathbf{Q}_2$.

We establish the following induction base by means of the method of [BKO 88] with some adaptation to the present setting; the induction step is established by means of the testing method (Corollary 2).

**Induction Base:** Let $\mathrm{lgt}(q) = 1$ and $q = (\langle \sigma', \Gamma' \rangle)$, and suppose

$q \in p_1$ and $q \notin p_2$.

Since $p_1$ satisfies the disjointness deadlock condition by the definition of $\mathbf{P}_2^*$, there exists $\mathcal{R}_1 \subseteq \wp(\mathrm{sort}[\mathrm{act}(p_1, \sigma)] \cap \mathbf{C})$ such that

$$\forall \Gamma \in \wp(\mathbf{C})[(\langle \sigma', \Gamma \rangle) \in p_1 \Leftrightarrow \exists R \in \mathcal{R}_1[\Gamma \cap R = \emptyset]].$$

Fix such $\mathcal{R}_1$. We will distinguish two cases according to whether $\exists \Gamma \in \wp(\mathbf{C})[(\langle \sigma', \Gamma \rangle) \in p_2]$.

**Case 1.** Suppose

$$\exists \Gamma \in \wp(\mathbf{C})[(\langle \sigma', \Gamma \rangle) \in p_2]. \tag{54}$$

Then, we will construct a tester $T$ such that

$$(\langle\tau,\sigma'\rangle) \in \alpha_2(p_1 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma) \setminus \alpha_2(p_2 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma). \tag{55}$$

By (54) and the fact that $p_2$ satisfies the disjointness deadlock condition, there exists

$$\mathcal{R}_2 \subseteq \wp(\mathrm{sort}[\mathrm{act}(p_2,\sigma)] \cap \mathbf{C}) \tag{56}$$

such that

$$\mathcal{R}_2 \neq \emptyset \wedge \forall\Gamma \in \wp(\mathbf{C})[(\langle\sigma',\Gamma\rangle) \in p_2 \Leftrightarrow \exists R \in \mathcal{R}_2[\Gamma \cap R = \emptyset]]. \tag{57}$$

Fix such $\mathcal{R}_2$, and let

$$\Gamma'' = \mathrm{sort}[\mathrm{act}(p_2,\sigma)] \cap \Gamma'. \tag{58}$$

By (57) and the fact that $q \notin p_2$, one has
$\forall R' \in \mathcal{R}_2[\Gamma' \cap R' \neq \emptyset]$.
Thus, by (58), $\Gamma''$ is nonempty. Moreover $\mathrm{sort}[\mathrm{act}(p_2,\sigma)]$ is finite since $\mathrm{ASFin}(p_2)$, which implies that $\Gamma''$ is finite. Let
$\Gamma'' = \{\gamma_1,\cdots,\gamma_n\}$,
and let

$$\begin{cases} T \equiv (x := \sigma'); T', \\ T' \equiv \phi(\overline{\gamma_1}) + \cdots + \phi(\overline{\gamma_n}), \end{cases}$$

where

$$\phi(\gamma) = \begin{cases} (c!\ v_0); \mathrm{Stop} & \text{if } \gamma = c!, \\ (c?\ x); \mathrm{Stop} & \text{if } \gamma = c?. \end{cases}$$

With this $T$, we will show that

$$(\langle\tau,\sigma'\rangle) \in \alpha_2(p_1 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma) \setminus \alpha_2(p_2 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma).$$

First, let us show

$$(\langle\tau,\sigma'\rangle) \in \alpha_2(p_1 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma).$$

Since $(\langle\sigma',\Gamma'\rangle) \in p_1$, one has

$$\exists R \in \mathcal{R}_1[\Gamma' \cap R = \emptyset]. \tag{59}$$

Fix such $R$. Then

$$\begin{cases} (\langle\sigma',\mathbf{C} \setminus R\rangle) \in p_1. \\ (\langle\sigma',\mathbf{C} \setminus \overline{\Gamma''}\rangle) \in \mathcal{D}_2[\![\phi(\overline{\gamma_1}) + \cdots + \phi(\overline{\gamma_n})]\!]. \end{cases} \tag{60}$$

Moreover

$$\begin{aligned} &(\mathbf{C} \setminus (\mathbf{C} \setminus R)) \cap \overline{(\mathbf{C} \setminus (\mathbf{C} \setminus \overline{\Gamma''}))} \\ &= R \cap \Gamma'' \\ &= R \cap \mathrm{sort}[\mathrm{act}(p_2,\sigma)] \cap \Gamma' \quad \text{(by (58))} \\ &\subseteq R \cap \Gamma' \\ &= \emptyset \quad \text{(by (59))}. \end{aligned}$$

By this, (60), and the definitions of $\tilde{\|}$ and $\#$, one has

$$(\langle\sigma,\tau,\sigma'\rangle, \langle\sigma',\emptyset\rangle) \in p_1 \tilde{\|} \mathcal{D}_2[\![T]\!],$$

that is,

$$(\langle\tau,\sigma'\rangle) \in \alpha_2(p_1 \tilde{\|} \mathcal{D}_2[\![T]\!])(\sigma). \tag{61}$$

Next let us show, by contradiction, that

$$(\langle\tau,\sigma'\rangle)\notin\alpha_2(p_2 \mathbin{\tilde{\|}} \mathcal{D}_2[\![T]\!])(\sigma).$$

Assume

$$(\langle\tau,\sigma'\rangle)\in\alpha_2(p_2 \mathbin{\tilde{\|}} \mathcal{D}_2[\![T]\!])(\sigma). \tag{62}$$

Then, by the definition of $\alpha_2$, one has

$$(\langle\sigma,\tau,\sigma'\rangle,\langle\sigma',\emptyset\rangle)\in p_2 \mathbin{\tilde{\|}} \mathcal{D}_2[\![T]\!]. \tag{63}$$

Hence either

$$(\langle\sigma',\emptyset\rangle)\in p_2[\langle\sigma,\tau,\sigma'\rangle] \mathbin{\tilde{\|}} \mathcal{D}_2[\![T]\!]$$

or

$$(\langle\sigma',\emptyset\rangle)\in p_2 \mathbin{\tilde{\|}} \mathcal{D}_2[\![T]\!][\langle\sigma,\tau,\sigma'\rangle].$$

The first case is impossible, since

$$\neg\exists\Gamma[(\langle\sigma',\Gamma'\rangle)\in\mathcal{D}_2[\![T]\!]].$$

Hence

$$(\langle\sigma',\emptyset\rangle)\in p_2 \mathbin{\tilde{\|}} \mathcal{D}_2[\![T]\!][\langle\sigma,\tau,\sigma'\rangle].$$

By this and the definitions of $\tilde{\|}$ and $\#$, there exist $\Gamma_1,\Gamma_2\in\wp(\mathbf{C})$ such that

$$(\langle\sigma',\Gamma_1\rangle)\in p_2\wedge(\langle\sigma',\Gamma_2\rangle)\in\mathcal{D}_2[\![T']\!]\wedge(\mathbf{C}\setminus\Gamma_1)\cap\overline{(\mathbf{C}\setminus\Gamma_2)}=\emptyset. \tag{64}$$

Moreover, there exists $R'\in\mathcal{R}_2$ such that

$$\Gamma_1\cap R'=\emptyset.$$

Fix such $R'$. Then

$$\mathbf{C}\setminus\Gamma_1\supseteq R'. \tag{65}$$

By the fact that $(\langle\sigma',\Gamma'\rangle)\notin p_2$, one has

$$\Gamma'\cap R'\neq\emptyset. \tag{66}$$

By the condition $(\langle\sigma',\Gamma_2\rangle)\in\mathcal{D}_2[\![T']\!]$ in (64), one has

$$\Gamma_2\cap\overline{\Gamma''}=\emptyset,$$

that is,

$$\mathbf{C}\setminus\Gamma_2\supseteq\overline{\Gamma''},$$

and therefore,

$$\overline{\mathbf{C}\setminus\Gamma_2}\supseteq\Gamma''. \tag{67}$$

Thus

$$\begin{aligned}
&(\mathbf{C}\setminus\Gamma_1)\cap\overline{\mathbf{C}\setminus\Gamma_2}\\
&\supseteq R'\cap\Gamma'' \quad \text{(by (65) and (67))}\\
&= R'\cap(\mathrm{sort}[\mathrm{act}(p_2,\sigma)]\cap\Gamma') \quad \text{(by (58))}\\
&= R'\cap\Gamma' \quad (\text{since } R'\subseteq\mathrm{sort}[\mathrm{act}(p_2,\sigma)] \text{ by (56)})\\
&\neq\emptyset \quad \text{(by (66))}.
\end{aligned}$$

This contradicts (64). Hence (62) is false, and therefore, one has

$$(\langle\tau,\sigma'\rangle)\notin\alpha_2(p_2 \mathbin{\tilde{\|}} \mathcal{D}_2[\![T]\!])(\sigma).$$

By this and (61), one has

$$(\langle\tau,\sigma'\rangle)\in\alpha_2(p_1 \mathbin{\tilde{\|}} \mathcal{D}_2[\![T]\!])(\sigma)\setminus\alpha_2(p_2 \mathbin{\tilde{\|}} \mathcal{D}_2[\![T]\!])(\sigma).$$

**Case 2.** Suppose $\neg\exists\Gamma\in\wp(\mathbf{C})[(\langle\sigma',\Gamma\rangle)\in p_2]$.
Then, one has the same result by putting

$$T\equiv(x:=\sigma');\mathrm{Stop}.$$

**Induction Step:** By means of Corollary 2, it is shown for every $k\in\omega$ that

$$\forall q[\mathrm{lgt}(q)=k\Rightarrow\Phi_2(q)]\Rightarrow\forall q[\mathrm{lgt}(q)=k+1\Rightarrow\Phi_2(q)],$$

as in the induction step of the proof of Lemma 12 (1). ■

# 5 Concluding Remarks

Table 1: Results on Fully Abstract Models for Communicating Processes

| | | | |
|---|---|---|---|
| Liner Time | Strong | Uniform | [BKO 88]: Characterization of a fully abstract compositional model.[*1]<br>[Rut 89]: Construction of a fully abstract denotational model.[*2] |
| | | Nonuniform | This paper: Construction of a fully abstract denotational model with respect to an operational model *with states*.[*3]<br>?: With respect to an operational model *without states*.[*4] |
| | Weak | Uniform | ? [*5] |
| | | Nonuniform | ? [*6] |
| Branching Time | Strong | Uniform | [Mil 80] and [Mil 85]: Characterization of a fully abstract compositional model for CCS.[*7]<br>[GV 88]: Characterization of a fully abstract compositional model in general.[*8]<br>[Rut 90]: Construction of a fully abstract denotational model.[*9] |
| | | Nonuniform | ? |
| | Weak | Uniform | [Mil 80] and [Mil 85]: Characterization of a fully abstract compositional model.[*10] |
| | | Nonuniform | ? |

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 $\mathcal{L}_3$ in [BR 90].

The other is to investigate fully abstract denotational models for the same languages $\mathcal{L}_1$ (or $\mathcal{L}_2$) with respect to other operational models.

For instance, it might be possible to construct a fully abstract denotational model for an operational model Op for $\mathcal{L}_1$ defined as follows:

For every statement $s$ and state $\sigma$,

$$
\begin{aligned}
&\mathrm{Op}[\![s]\!](\sigma) = \\
&\{\sigma' : \exists\langle s_1,\sigma_1\rangle,\cdots,\langle s_n,\sigma_n\rangle[\sigma_n = \sigma' \wedge \langle s,\sigma\rangle \rightarrow_1 \langle s_1,\sigma_1\rangle \rightarrow_1 \cdots \rightarrow_1 \langle s_n,\sigma_n\rangle] \\
&\qquad \wedge\neg\exists\langle s',\sigma'\rangle[\langle s_n,\sigma_n\rangle \rightarrow_1 \langle s',\sigma'\rangle]\ \} \\
&\cup \mathrm{if}(\exists(\langle s_n,\sigma_n\rangle)_{n\in\omega}[s_0 = s \wedge \sigma_0 = \sigma \wedge \forall n \in \omega[\langle s_n,\sigma_n\rangle \rightarrow_1 \langle s_{n+1},\sigma_{n+1}\rangle]], \\
&\qquad \{\perp\},\emptyset).
\end{aligned}
$$

It was shown in [AP 86] that there is no fully abstract denotational model with respect to Op if the language has *countable nondeterminism* and *random assignment*. However it is still to be investigated whether there is a fully abstract denotational model with respect to Op, since the language $\mathcal{L}_1$ does not have random assignment. It seems that $\mathcal{D}_1$ is not fully abstract with respect to Op; at least, we cannot establish the full abstractness with respect to Op as we have done with respect to $\mathcal{O}_1$, since there are $s_1, s_2 \in \mathcal{L}_1$ such that $\mathcal{D}_1[\![s_1]\!] \neq \mathcal{D}_1[\![s_2]\!]$ but $\forall T \in \mathcal{L}_1[\mathrm{Op}[\![s_1 \parallel T]\!] = \mathrm{Op}[\![s_2 \parallel T]\!]$. This is easily verified by putting $s_1 \equiv \mathrm{Stop}$ and $s_2 \equiv (x := x); \mathrm{Stop}$.

For $\mathcal{L}_2$, a language for communicating concurrent systems, there are several possible operational models besides $\mathcal{O}_2$ defined in Section 4.

There are several dimensions for classifying operational models for such a language; such a classification and comparative study of those models were presented in [Gla 90]. 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 *branching time*. Another dimension is the dichotomy *weak* versus *strong*: a model is called *weak*, if it identifies processes differing only in their internal or silent actions (denoted by $\tau$ 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 to construct a fully abstract denotational model, or to characterize somehow the fully abstract compositional model. The obtained results on these problems so far are summarized in Table 1.

As described in the introduction, fully abstract models for uniform languages with respect to *strong* operational models of the *linear time* variety were investigated in [BKO 88] and [Rut 89] (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 with respect to 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 $\sigma$,

$$\mathcal{O}_2^*[\![s]\!](\sigma) = \begin{cases} \bigcup\{(a)\cdot\mathcal{O}_2^*[\![s']\!](\sigma') : \langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle\} & \text{if } \exists a, \exists\langle s',\sigma'\rangle[\langle s,\sigma\rangle \xrightarrow{a}_2 \langle s',\sigma'\rangle], \\ \{\epsilon\} & \text{otherwise.} \end{cases}$$

It is to be investigated whether $\mathcal{D}_2$ is fully abstract with respect to $\mathcal{O}_2^*$ (cf. $*4$ in Table 1).

It seems more difficult to construct fully abstract denotational models with respect to *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 $\sigma$,
$\mathcal{O}_2^{**}[\![s]\!](\sigma) = \{\rho\setminus\tau : \rho\in\mathcal{O}_2^*[\![s]\!](\sigma)\}$,
where $\rho\setminus\tau$ is the result of ignoring $\tau$'s in $\rho\in(\mathbf{C}\cup\{\tau\})^{\leq\omega}$.

This problem seems difficult not only in the nonuniform setting but also in the uniform setting (cf. $*5, *6$ in Table 1). A related discussion is found in the last section of [BKO 88].

In [Mil 80] and [Mil 85]), 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 [GV 88] 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 [Rut 90]; the denotational models are fully abstract with respect to the operational models by definition (cf. $*9$ in Table 1).

In [Mil 80] and [Mil 85]), Milner characterized a fully abstract compositional model for CCS with respect to *weak bisimulation* $\approx$, which coincides with *observation equivalence* under certain conditions (cf. $*10$ in Table 1). This relation $\approx$ is a weak operational equivalence relation of the *branching time* variety. This characterization was established by showing that *observation congruence* $\approx^c$, which is the coarsest congruent relation included in $\approx$, coincides with the equivalence relation $\approx^+$ defined as follows:

For every two statements $s_1$ and $s_2$,
$s_1 \approx^+ s_2$ iff $\forall s[s_1 + s \approx s_2 + s]$.

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 abstractness 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 abstractness problem for a concurrent language in this setting see [HP 79]. In [Hen 88], which is based on [DH 83], [Hen 83], and [Hen 85], Hennessy showed in detail the full abstractness of a denotational model consisting of *acceptance trees* equipped with a complete partial order, with respect to *testing equivalence*. In [Mu 85], the question of semantic equivalence and full abstractness, mainly for typed lambda-calculi, was tackled with the help of so-called inclusive predicates, again in an order-theoretic framework.

Fully abstract models can be defined *syntactically* for some languages as Milner showed for typed lambda-calculi in [Mil 77], while it may be unknown whether these models are denotational in the sense explained in the introduction.

For a survey of the full abstractness problem for sequential languages, see [BCL 85]. In [St 86], the general question concerning the existence of fully abstract models was treated in an algebraic context.

# Acknowledgements

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

# A Appendices

## A.1 Proof of Lemma 1

(1) Let $A$, $B$, $\mathbf{Q}$, $\pi_n$, $\mathbf{P}$, and $\tilde{\pi}_n$ be as in Definition 2 $(n \in \omega)$. For $n \in \omega$, let

$$\Phi(n) \Leftrightarrow \forall q_1, q_2[(\pi_{n+1}(q_1) \neq \pi_{n+1}(q_2) \Rightarrow d_Q(q_1,q_2) \geq (1/2)^n) \\ \wedge(\pi_n(q_1) = \pi_n(q_2) \Rightarrow d_Q(q_1,q_2) \leq (1/2)^n)\ ].$$

It is sufficient to show $\forall n \in \omega[\Phi(n)]$, which will be established by induction.

**Induction Base:** Fix $q_1, q_2 \in \mathbf{Q}$.

First, let us show that

$$\pi_1(q_1) \neq \pi_1(q_2) \Rightarrow d_Q(q_1,q_2) = (1/2)^0.$$

Suppose

$$\pi_1(q_1) \neq \pi_1(q_2). \tag{68}$$

**Case 1.** Suppose $q_1, q_2 \in B$. Then

$$\pi_1(q_1) = q_1 \wedge \pi_1(q_2) = q_2.$$

By this and (68), one has

$d_Q(q_1,q_2) = 1 = (1/2)^0$.

**Case 2.** Suppose $q_1, q_2 \in A \times \mathbf{Q}$. Then, by (68), one has

$$\exists a_1, a_2 \in A, \exists q_1', q_2' \in \mathbf{Q}[q_1 = \langle a_1, q_1'\rangle \wedge q_2 = \langle a_2, q_2'\rangle \wedge a_1 \neq a_2].$$

Hence $d_Q(q_1,q_2) = 1 = (1/2)^0$.

**Case 3.** Otherwise, by the definition of $d_Q$, one has

$d_Q(q_1,q_2) = 1 = (1/2)^0$.

Thus one has

$$\pi_1(q_1) \neq \pi_1(q_2) \Rightarrow d_Q(q_1,q_2) = (1/2)^0.$$

Moreover by the definition of $d_Q$, it holds that $d_Q(q_1,q_2) \leq (1/2)^0$.

Thus one has $\Phi(0)$.

**Induction Step:** Fix $n \in \omega$ and assume $\Phi(n)$. Then fix $q_1, q_2 \in \mathbf{Q}$.

First, let us show that

$$\pi_{(n+1)+1}(q_1) \neq \pi_{(n+1)+1}(q_2) \Rightarrow d_Q(q_1, q_2') \geq (1/2)^{n+1}.$$

Suppose

$$\pi_{(n+1)+1}(q_1) \neq \pi_{(n+1)+1}(q_2). \tag{69}$$

**Case 1.** Suppose $q_1, q_2 \in A \times \mathbf{Q}$. Then

$$\exists a_1, a_2 \in A, \exists q_1', q_2' \in \mathbf{Q}[q_1 = \langle a_1, q_1' \rangle \wedge q_2 = \langle a_2, q_2' \rangle].$$

If $a_1 \neq a_2$, then

$$d_Q(q_1, q_2) = 1 \geq (1/2)^{n+1}.$$

Otherwise, (69) implies that

$$\pi_{n+1}(q_1') \neq \pi_{n+1}(q_2').$$

By this and the induction hypothesis, one has

$$d_Q(q_1', q_2') \geq (1/2)^n,$$

and therefore

$$d_Q(q_1, q_2) = (1/2) \cdot d_Q(q_1', q_2') \geq (1/2)^{n+1}.$$

**Case 2.** Otherwise as in the induction base, one has

$$d_Q(q_1, q_2') = 1 \geq (1/2)^{n+1}.$$

Thus one has

$$\pi_{(n+1)+1}(q_1) \neq \pi_{(n+1)+1}(q_2) \Rightarrow d_Q(q_1, q_2) = (1/2)^{n+1}.$$

Next let us show that

$$\pi_{n+1}(q_1) = \pi_{n+1}(q_2) \Rightarrow d_Q(q_1, q_2) \leq (1/2)^{n+1}.$$

Suppose

$$\pi_{n+1}(q_1) = \pi_{n+1}(q_2). \tag{70}$$

Then, if $q_1, q_2 \in \mathbf{B}$, one has $q_1 = q_2$, and therefore, $d_Q(q_1, q_2) = 0 \leq (1/2)^{n+1}$.
Otherwise, by the definition of $\pi_{n+1}$, one has

$$\exists a \in A, \exists q_1', q_2' \in \mathbf{Q}[q_1 = \langle a, q_1' \rangle \wedge q_2 = \langle a, q_2' \rangle].$$

By (70), one has

$$\pi_n(q_1') = \pi_n(q_2').$$

By this and the induction hypothesis, one has

$$d_Q(q_1', q_2') \leq (1/2)^n,$$

and therefore

$$d_Q(q_1, q_2) = (1/2) \cdot d_Q(q_1', q_2') \leq (1/2)^{n+1}.$$

Thus one has

$$\pi_{n+1}(q_1) = \pi_{n+1}(q_2) \Rightarrow d_Q(q_1', q_2') \leq (1/2)^{n+1}.$$

Summing up, one has $\Phi(n+1)$.

**(2)** By **(1)** and the definitions of $\tilde{\pi}_n$ ($n \in \omega$) and $d_P$, one has

$$\begin{aligned} &\forall n \in \omega, \forall p_1, p_2 \in \mathbf{P} \\ &[(\tilde{\pi}_n(p_1) = \tilde{\pi}_n(p_2) \Rightarrow d(p_1, p_2) \leq (1/2)^n) \\ &\wedge(\tilde{\pi}_{n+1}(p_1) \neq \tilde{\pi}_{n+1}(p_2) \Rightarrow d(p_1, p_2) \geq (1/2)^n)], \end{aligned}$$

which implies the claim of this part.

**(3)** This part follows immediately from **(2)**. ∎

## A.2 Proof of Lemma 8

By Lemma 5 (2), it is sufficient to show

$$\forall k \geq 1, \forall p_1, \cdots, p_k, p' \in \mathbf{P}_1[\bigcup_{i \in \bar{k}}[p_i] \,\tilde{\|}\, p' = \bigcup_{i \in \bar{k}}[p_i \,\tilde{\|}\, p']]. \quad (71)$$

Let $\mathbf{M}_1^{\text{dis}} = ((\wp_+(\bar{k}) \to \mathbf{P}_1) \times \mathbf{P}_1 \to \mathbf{P}_1)$. Let $F, G \in \mathbf{M}_1^{\text{dis}}$ be as follows:

For $I \in \wp_+(\bar{k})$, $(p_i)_{i \in I} \in (I \to \mathbf{P}_1)$, and $p' \in \mathbf{P}_1$,

$$F((p_i)_{i \in I}, p') = \bigcup_{i \in I}[p_i] \,\tilde{\|}\, p',$$
$$G((p_i)_{i \in I}, p') = \bigcup_{i \in I}[p_i \,\tilde{\|}\, p'].$$

By the definition of $\tilde{\|}$, one has

$$F((p_i)_{i \in I}, p') = \bigcup_{i \in I}[p_i] \,\lfloor\!\lfloor\, p' \cup p' \,\lfloor\!\lfloor\, \bigcup_{i \in I}[p_i] \cup (\bigcup_{i \in I}[p_i] \,\#\, (p')). \quad (72)$$

For $J \subseteq I$, let

$$A(J) = \{\langle \sigma, \sigma' \rangle : \forall i \in J[p_i[\langle \sigma, \sigma' \rangle] \neq \emptyset] \wedge \forall i \in (I \setminus J)[p_i[\langle \sigma, \sigma' \rangle] = \emptyset]\}.$$

Then

$$\begin{aligned}
&\bigcup_{i \in I}[p_i] \,\lfloor\!\lfloor\, p' \\
&= \bigcup\{\langle \sigma, \sigma' \rangle \cdot (\bigcup_{i \in I}[p_i][\langle \sigma, \sigma' \rangle] \,\lfloor\!\lfloor\, p') : \bigcup_{i \in I}[p_i][\langle \sigma, \sigma' \rangle] \neq \emptyset\} \\
&= \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, \sigma' \rangle \in A(J)}[\langle \sigma, \sigma' \rangle \cdot (\bigcup_{i \in I}[p_i][\langle \sigma, \sigma' \rangle] \,\lfloor\!\lfloor\, p')]] \\
&= \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, \sigma' \rangle \in A(J)}[\langle \sigma, \sigma' \rangle \cdot F((p_i[\langle \sigma, \sigma' \rangle])_{i \in J}, p')]].
\end{aligned}$$

Let $\mathcal{F} : \mathbf{M}_1^{\text{dis}} \to \mathbf{M}_1^{\text{dis}}$ be defined as follows:

For $f \in \mathbf{M}_1^{\text{dis}}$,

$$\begin{aligned}
&\mathcal{F}(f)((p_i)_{i \in I}, p') \\
&= \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, \sigma' \rangle \in A(J)}[\langle \sigma, \sigma' \rangle \cdot f((p_i[\langle \sigma, \sigma' \rangle])_{i \in J}, p')]] \\
&\quad \cup \bigcup\{\langle \sigma, \sigma' \rangle \cdot f((p_i)_{i \in I}, p'[\langle \sigma, \sigma' \rangle]) : p'[\langle \sigma, \sigma' \rangle] \neq \emptyset\} \\
&\quad \cup (\bigcup_{i \in I}[p_i] \,\#\, p').
\end{aligned} \quad (73)$$

Then $\mathcal{F}$ is a contraction and $F = \mathcal{F}(F)$, i.e., $F = \text{fix}(\mathcal{F})$.

Next, let us show that $G = \mathcal{F}(G)$.

$$\begin{aligned}
&G((p_i)_{i \in I}, p') \\
&= \bigcup_{i \in I}[(p_i \,\lfloor\!\lfloor\, p') \cup (p' \,\lfloor\!\lfloor\, p_i) \cup (p_i \,\#\, p')] \\
&= \bigcup_{i \in I}[p_i \,\lfloor\!\lfloor\, p'] \cup \bigcup_{i \in I}[p' \,\lfloor\!\lfloor\, p_i] \cup \bigcup_{i \in I}[p_i \,\#\, p'] \\
&= \bigcup_{i \in I}[p_i \,\lfloor\!\lfloor\, p'] \\
&\quad \cup \bigcup\{\langle \sigma, \sigma' \rangle \cdot G((p_i)_{i \in I}, p'[\langle \sigma, \sigma' \rangle]) : p'[\langle \sigma, \sigma' \rangle] \neq \emptyset\} \\
&\quad \cup (\bigcup_{i \in I}[p_i] \,\#\, p').
\end{aligned} \quad (74)$$

Thus it is sufficient to show

$$\bigcup_{i \in I}[p_i \,\lfloor\!\lfloor\, p'] = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, \sigma' \rangle \in A(J)}[\langle \sigma, \sigma' \rangle \cdot G((p_i[\langle \sigma, \sigma' \rangle])_{i \in J}, p')]].$$

For $H : I \times \wp_+(I) \to \mathbf{P}_1$, it is immediate that

$$\begin{aligned}
&\bigcup_{i \in I}[\bigcup\{H(i, J) : J \in \wp_+(I) \wedge i \in J\}] \\
&= \bigcup_{J \in \wp_+(I)}[\bigcup_{i \in J}[H(i, J)]].
\end{aligned} \quad (75)$$

Hence

$$
\begin{aligned}
&\bigcup_{i\in I}[p_i \,\lfloor\!\lfloor\, p'] \\
&= \bigcup_{i\in I}[\bigcup\{\langle\sigma,\sigma'\rangle \cdot (p_i[\langle\sigma,\sigma'\rangle] \,\tilde{\|}\, p') : p_i[\langle\sigma,\sigma'\rangle] \neq \emptyset\}] \\
&= \bigcup_{i\in I}[\bigcup\{\bigcup_{\langle\sigma,\sigma'\rangle\in A(J)}[\langle\sigma,\sigma'\rangle \cdot (p_i[\langle\sigma,\sigma'\rangle] \,\tilde{\|}\, p')] : J \in \wp_+(I) \wedge i \in J\}] \\
&= \bigcup_{J\in\wp_+(I)}[\bigcup_{i\in J}[\bigcup_{\langle\sigma,\sigma'\rangle\in A(J)}[\langle\sigma,\sigma'\rangle \cdot (p_i[\langle\sigma,\sigma'\rangle] \,\tilde{\|}\, p')]]] \\
&\qquad \text{(by (75) with } H(i,J) = \bigcup_{\langle\sigma,\sigma'\rangle\in A(J)}[\langle\sigma,\sigma'\rangle \cdot (p_i[\langle\sigma,\sigma'\rangle] \,\tilde{\|}\, p')]) \\
&= \bigcup_{J\in\wp_+(I)}[\bigcup_{\langle\sigma,\sigma'\rangle\in A(J)}[\bigcup_{i\in J}[\langle\sigma,\sigma'\rangle \cdot (p_i[\langle\sigma,\sigma'\rangle] \,\tilde{\|}\, p')]]] \\
&= \bigcup_{J\in\wp_+(I)}[\bigcup_{\langle\sigma,\sigma'\rangle\in A(J)}[\langle\sigma,\sigma'\rangle \cdot \bigcup_{i\in J}[p_i[\langle\sigma,\sigma'\rangle] \,\tilde{\|}\, p']]] \\
&= \bigcup_{J\in\wp_+(I)}[\bigcup_{\langle\sigma,\sigma'\rangle\in A(J)}[\langle\sigma,\sigma'\rangle \cdot G((p_i[\langle\sigma,\sigma'\rangle])_{i\in J}, p')]].
\end{aligned}
$$

By this and (74), one has

$$G((p_i)_{i\in I}, p') = \mathcal{F}(G)((p_i)_{i\in I}, p').$$

Hence $G = \mathrm{fix}(\mathcal{F}) = F$. ■

## A.3 Proof of Lemma 9

**(1)** Let $\alpha_1' : \mathbf{P}_1 \to (\Sigma \to \wp_{\mathrm{nc}}(\Sigma^{\leq\omega}))$ be defined as follows:

For $p \in \mathbf{P}_1$ and $\sigma \in \Sigma$,

$$\alpha_1'(p)(\sigma) = \{\mathrm{hist}_1(q) : q \in p \wedge \mathrm{istate}_1(q) = \sigma \wedge \mathrm{Exec}_1(q)\}. \tag{76}$$

The set $\alpha_1'(p)(\sigma)$ is closed, because $p$ is closed. Moreover $\alpha_1'(p)(\sigma)$ is nonempty, because $p$ is uniformly nonempty. Thus $\alpha_1'(p)(\sigma) \in \wp_{\mathrm{nc}}(\Sigma^{\leq\omega})$.

Let $\mathbf{M}_1^{\alpha}$ and $\Delta_1$ as in Definition 12. By Definition 12, it is sufficient to show

$$\alpha_1' = \Delta_1 \circ \alpha_1'. \tag{77}$$

Let $p \in \mathbf{P}_1$ and $\sigma \in \Sigma$. By the definition of $\Delta_1$, one has

$$
\begin{aligned}
&(\Delta_1 \circ \alpha_1')(p)(\sigma) \\
&= \bigcup\{\sigma' \cdot \{\mathrm{hist}_1(q') : q' \in p[\langle\sigma,\sigma'\rangle] \wedge \mathrm{istate}_1(q') = \sigma' \wedge \mathrm{Exec}_1(q')\} : p[\langle\sigma,\sigma'\rangle] \neq \emptyset\ \} \\
&\quad \cup \mathrm{if}(\langle\sigma\rangle \in p, \{\epsilon\}, \emptyset).
\end{aligned} \tag{78}
$$

First, by (76), (78), it is straightforward that

$$\epsilon \in \alpha_1'(p)(\sigma) \Leftrightarrow \epsilon \in (\Delta_1 \circ \alpha_1')(p)(\sigma).$$

Next, let us show, for $\sigma' \in \Sigma$ and $\rho' \in \Sigma^{\leq\omega}$, that

$$\sigma' \cdot \rho' \in \alpha_1'(p)(\sigma) \Leftrightarrow \sigma' \cdot \rho' \in (\Delta_1 \circ \alpha_1')(p)(\sigma). \tag{79}$$

($\Rightarrow$) Suppose $\sigma' \cdot \rho' \in \alpha_1'(p)(\sigma)$. Then there exists $q \in p$ such that

$$\sigma' \cdot \rho' = \mathrm{hist}_1(q) \wedge \mathrm{istate}_1(q) = \sigma \wedge \mathrm{Exec}_1(q).$$

For such $q$, there exists $q'$ such that

$$q = \langle\sigma,\sigma'\rangle \cdot q' \wedge \rho' = \mathrm{hist}_1(q') \wedge \mathrm{istate}_1(q') = \sigma' \wedge \mathrm{Exec}_1(q').$$

For such $q'$, $q' \in p[\langle\sigma,\sigma'\rangle]$, and therefore, $p[\langle\sigma,\sigma'\rangle] \neq \emptyset$. Hence

$$
\begin{aligned}
&\sigma' \cdot \rho' \\
&\in \bigcup\{\sigma'' \cdot \{\mathrm{hist}_1(q'') : q'' \in p[\langle\sigma,\sigma''\rangle] \wedge \mathrm{istate}_1(q'') = \sigma'' \wedge \mathrm{Exec}_1(q'')\} : p[\langle\sigma,\sigma''\rangle] \neq \emptyset\ \} \\
&\subseteq (\Delta_1 \circ \alpha_1')(p)(\sigma).
\end{aligned}
$$

($\Leftarrow$) Suppose

$$\sigma' \cdot \rho' \in (\Delta_1 \circ \alpha_1')(p)(\sigma).$$

Then

$$\sigma' \cdot \rho' \\ \in \bigcup\{\sigma'' \cdot \{\mathrm{hist}_1(q'') : q'' \in p[\langle\sigma,\sigma''\rangle] \wedge \mathrm{istate}_1(q'') = \sigma'' \wedge \mathrm{Exec}_1(q'')\} : p[\langle\sigma,\sigma''\rangle] \neq \emptyset\ \}.$$

Hence $p[\langle\sigma,\sigma'\rangle] \neq \emptyset$ and there exists $q' \in p[\langle\sigma,\sigma'\rangle]$ such that

$$\rho' = \mathrm{hist}_1(q') \wedge \mathrm{istate}_1(q') = \sigma' \wedge \mathrm{Exec}_1(q').$$

For such $q'$, let $q = \langle\sigma,\sigma'\rangle \cdot q'$. Then

$$q \in p \wedge \mathrm{istate}_1(q) = \sigma \wedge \mathrm{Exec}_1(q) \wedge \mathrm{hist}_1(q) = \sigma' \cdot \mathrm{hist}_1(q') = \sigma' \cdot \rho'.$$

Hence

$$\begin{aligned}\sigma' \cdot \rho' &\in \{\mathrm{hist}_1(q) : q \in p \wedge \mathrm{istate}_1(q) = \sigma \wedge \mathrm{Exec}_1(q)\} \\ &= \alpha_1'(p).\end{aligned}$$

Thus one has (79).
**(2)** This part follows immediately from **(1)**. ■

## A.4 Proof of Lemma 13 with $\sharp\mathrm{IVar} \geq 2$

We assumed that IVar is finite, which allows us to put $\mathrm{IVar} = \{x_k : k \in \bar{n}\}$ with $n \geq 1$, and identify $\Sigma$ with $\mathbf{V}^n$. Roughly, the proof of Lemma 13 with $\sharp\mathrm{IVar} \geq 2$ is given by $n$-times iteration of the method introduced in Subsection 3.5 for the proof with $\sharp\mathrm{IVar} = 1$.

First, we need some preliminary definitions.

**Notation 5**

**(1)** For $i, j \in \omega$, let $[i..j] = \{k \in \omega : i \leq k \leq j\}$.

**(2)** Let $k \geq 1$, and $\mathbf{v} \in \mathbf{V}^k$. For $i \in \bar{k}$, $\mathbf{v}(i)$ is the $i$-th component of $\mathbf{v}$. For $i \in [0..k]$, let $\mathbf{v}[i] = (\mathbf{v}(1), \cdots, \mathbf{v}(i))$. Remark that $\mathbf{v}[0] = \epsilon$.

**(3)** For $k \in \bar{n}$, $\mathbf{x}[k] = (x_1, \cdots, x_k)$. ■

**Definition 27** *(Format for Testers with $\sharp\mathrm{IVar} \geq 2$)*
For $k \in \bar{n}$, $\mathbf{v}, \mathbf{v}', \mathbf{v}_2 \in \mathbf{V}^k$, $v_1 \in \mathbf{V}$, $\mathbf{v}_3 \in \mathbf{V}^{k-1}$, and $T' \in \mathcal{L}_1$, the *tester* $F_k^{(n)}(\mathbf{v}, \mathbf{v}', v_1, \mathbf{v}_2, \mathbf{v}_3, T')$ is defined by induction $k$ as follows:

**(i)** $F_1^{(n)}(\mathbf{v}, \mathbf{v}', v_1, \mathbf{v}_2, \epsilon, T')$

$$= \begin{cases} (x_1 := v_1); T' & \text{if } \mathbf{v}'(1) = \mathbf{v}(1), \\ \mathrm{If}(x_1 = \mathbf{v}(1), & \\ \quad (x_1 := \mathbf{v}'(1)); (x_1 := v_1); T', & \\ \quad (x_1 := \mathbf{v}_2(1)); \mathrm{Stop}\ ) & \text{otherwise.} \end{cases}$$

**(ii)** For $k \in \overline{n-1}$,

$$F_{k+1}^{(n)}(\mathbf{v}, \mathbf{v}', v_1, \mathbf{v}_2, \mathbf{v}_3, T')$$
$$= \begin{cases} (x_k := \mathbf{v}_3(k)); & \\ \quad F_k^{(n)}(\mathbf{v}[\mathbf{v}_3(k)/x_k][k], \mathbf{v}'[k], v_1, \mathbf{v}_2[k], \mathbf{v}_3[k-1], T') & \text{if } \mathbf{v}'(k+1) = \mathbf{v}(k+1), \\ \mathrm{If}(x_{k+1} = \mathbf{v}(k+1), & \\ \quad (x_{k+1} := \mathbf{v}'(k+1)); (x_k := \mathbf{v}_3(k)); & \\ \quad\quad F_k^{(n)}(\mathbf{v}[\mathbf{v}_3(k)/x_k][k], \mathbf{v}'[k], v_1, \mathbf{v}_2[k], \mathbf{v}_3[k-1], T'), & \\ \quad (x_{k+1} := \mathbf{v}_2(k+1)); \mathrm{Stop}\ ) & \text{otherwise.} \end{cases}$$

■

**Definition 28** *(Testing Sequence with $\sharp$IVar $\geq 2$)*
Let $k \in \bar{n}$, $\mathbf{v} \in \mathbf{V}^n$, $\mathbf{v}' \in \mathbf{V}^k$, and $\mathbf{v}_3 \in \mathbf{V}^{k-1}$. The *testing sequence* $r_k^{(n)}(\mathbf{v}, \mathbf{v}', \mathbf{v}_3)$ is defined by induction on $k$ as follows:

**(i)** $r_1^{(n)}(\mathbf{v}, \mathbf{v}', \epsilon)$
$$= \begin{cases} \epsilon & \text{if } \mathbf{v}(1) = \mathbf{v}'(1), \\ \langle \mathbf{v}, \mathbf{v}[\mathbf{v}'(1)/x_1] \rangle & \text{otherwise.} \end{cases}$$

**(ii)** For $k \in \overline{n-1}$,

$$r_{k+1}^{(n)}(\mathbf{v}, \mathbf{v}', \mathbf{v}_3)$$
$$= \begin{cases} \langle \mathbf{v}, \mathbf{v}[\mathbf{v}_3(k)/x_k] \rangle \cdot r_k^{(n)}(\mathbf{v}[\mathbf{v}_3(k)/x_k], \mathbf{v}'[k], \mathbf{v}_3[k-1]) & \text{if } \mathbf{v}'(k+1) = \mathbf{v}(k+1), \\ \langle \mathbf{v}, \mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}] \rangle & \\ \quad \cdot \langle \mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}], \mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}][\mathbf{v}_3(k)/x_k] \rangle & \\ \quad \cdot r_k^{(n)}(\mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}][\mathbf{v}_3(k)/x_k], \mathbf{v}'[k], \mathbf{v}_3[k-1]) & \text{otherwise.} \end{cases} \blacksquare$$

By means of the format for testers and the testng sequences, we can establish the following lemma, from which the Testing Lemma with $\sharp$IVar $\geq 2$ follows immediately.

**Lemma 24** *Let $p \in \mathbf{P}_1^*$, and $\mathbf{v}', \mathbf{v}'', \mathbf{v}_0 \in \mathbf{V}^n$.*

**(1)** *For every $v_1 \in \mathbf{V}$, $\mathbf{v}_2 \in \mathbf{V}^n$, and $\mathbf{v}_3 \in \mathbf{V}^{n-1}$, the following hold:*

- **(i)** *The sequence $r_n^{(n)}(\mathbf{v}_0, \mathbf{v}', \mathbf{v}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle$ is executable and its initial state is $\mathbf{v}_0$.*
- **(ii)** *For every $T' \in \mathcal{L}_1$, the following hold:*
  - **(a)** $\mathcal{D}_1[\![F_n^{(n)}(\mathbf{v}_0, \mathbf{v}', v_1, \mathbf{v}_2, \mathbf{v}_3, T')]\!][r_n^{(n)}(\mathbf{v}_0, \mathbf{v}', \mathbf{v}_3) \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle] = \mathcal{D}_1[\![T']\!]$.
  - **(b)** $\forall q' \in \mathbf{Q}_1$
    $[p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!]$
    $\Rightarrow r_n^{(n)}(\mathbf{v}_0, \mathbf{v}', \mathbf{v}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q' \in p \,\tilde{\|}\, \mathcal{D}_1[\![F_n^{(n)}(\mathbf{v}_0, \mathbf{v}', v_1, \mathbf{v}_2, \mathbf{v}_3, T')]\!]\ ]$.

**(2)** *There exist $v_1 \in \mathbf{V}$, $\mathbf{v}_2 \in \mathbf{V}^n$, and $\mathbf{v}_3 \in \mathbf{V}^{n-1}$ such that for every $T' \in \mathcal{L}_1$, the following holds:*
$\forall q' \in \mathbf{Q}_1$
$[r_n^{(n)}(\mathbf{v}_0, \mathbf{v}', \mathbf{v}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q' \in p \,\tilde{\|}\, \mathcal{D}_1[\![F_n^{(n)}(\mathbf{v}_0, \mathbf{v}', v_1, \mathbf{v}_2, \mathbf{v}_3, T')]\!]$
$\Rightarrow p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!]\ ]$. $\blacksquare$

**Proof.** **(1)** Let $v_1 \in \mathbf{V}$, $\mathbf{v}_2 \in \mathbf{V}^n$, and $\mathbf{v}_3 \in \mathbf{V}^{n-1}$.
First, let us show **(i)**. It is shown immediately by induction on $k \in \bar{n}$ that for every $k \in \bar{n}$, the following holds:

$$\begin{aligned} &\forall \mathbf{v} \in \mathbf{V}^n \\ &[\forall i \in (\bar{n} \setminus \bar{k})[\mathbf{v}(i) = \mathbf{v}'(i)] \Rightarrow \\ &\quad \mathrm{Exec}_1(r_k^{(n)}(\mathbf{v}, \mathbf{v}'[k], \mathbf{v}_3[k-1]) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle) \\ &\quad \wedge \mathrm{istate}_1(r_k^{(n)}(\mathbf{v}, \mathbf{v}'[k], \mathbf{v}_3[k-1]) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle) = \mathbf{v}]. \end{aligned}$$

Putting $k = n$ and $\mathbf{v} = \mathbf{v}_0$, one has the desired result.
Next let us show **(ii)**. Fix $T' \in \mathcal{L}_1$.
Part **(a)**: It is shown immediately by induction on $k \in \bar{n}$ that for $k \in \bar{n}$ the following holds:

$$\begin{aligned} &\forall \mathbf{v} \in \mathbf{V} \\ &[\mathcal{D}_1[\![F_k^{(n)}(\mathbf{v}[k], \mathbf{v}'[k], v_1, \mathbf{v}_2[k], \mathbf{v}_3[k-1], T')]\!][r_k^{(n)}(\mathbf{v}, \mathbf{v}'[k], \mathbf{v}_3[k-3]) \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle] \\ &\quad = \mathcal{D}_1[\![T']\!]\ ]. \end{aligned}$$

Putting $k = n$ and $\mathbf{v} = \mathbf{v}_0$, one has the desired reslut.
Part **(b)**: It is shown immediately by induction on $k \in \bar{n}$ that for $k \in \bar{n}$ the following holds:

$$\forall \mathbf{v} \in \mathbf{V}^n, \forall q' \in \mathbf{Q}_1$$
$$[p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!] \Rightarrow$$
$$r_k^{(n)}(\mathbf{v}, \mathbf{v}'[k], \mathbf{v}_3[k-1]) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q'$$
$$\in p \,\tilde{\|}\, \mathcal{D}_1[\![F_k^{(n)}(\mathbf{v}[k], \mathbf{v}'[k], v_1, \mathbf{v}_2[k], \mathbf{v}_3[k-1], T')]\!]\ ].$$

Putting $k = n$ and $\mathbf{v} = \mathbf{v}_0$, one has the desired reslut.
**(2)** Let us show by induction on $i \in \bar{n}$ that for every $i \in \bar{n}$ the following holds:

$$\forall \mathbf{v} \in \mathbf{V}^n, \forall q' \in \mathbf{Q}_1$$
$$[\exists v_1 \in \mathbf{V}, \exists \mathbf{v}_2 \in \mathbf{V}^i, \exists \mathbf{v}_3 \in \mathbf{V}^{i-1}$$
$$[r_i^{(n)}(\mathbf{v}, \mathbf{v}'[i], \mathbf{v}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q' \tag{80}$$
$$\in p \,\tilde{\|}\, \mathcal{D}_1[\![F_i^{(n)}(\mathbf{v}[i], \mathbf{v}'[i], v_1, \mathbf{v}_2, \mathbf{v}_3, T')]\!]$$
$$\Rightarrow p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!]\ ]].$$

**Induction Base:** We can show (80) with $i = 1$ in a similar fashion to the proof of Lemma 13 in Subsection 3.5. Let $\mathbf{v} \in \mathbf{V}^n$ and $q' \in \mathbf{Q}_1$. We distinguish two cases according to whether $\mathbf{v}(1) = \mathbf{v}'(1)$.
**Case 1.** Suppose $\mathbf{v}(1) = \mathbf{v}'(1)$. Then, by the defintions of $r_1^{(n)}(\cdots)$ and $F_1^{(n)}(\cdots)$, it is sufficient to show that

$$\exists v_1 \in \mathbf{V}[\ \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q' \in p \,\tilde{\|}\, \mathcal{D}_1[\![(x_1 := v_1); T']\!] \tag{81}$$
$$\Rightarrow p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \wedge q' \in p \,\tilde{\|}\, \mathcal{D}_1[\![T']\!]\ ].$$

We can establish this by choosing $v_1$ such that

$$v_1 \neq \mathbf{v}''(1),$$
$$v_1 \notin \{v \in \mathbf{V} : \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v/x_1] \rangle \in p^{[2]}\}.$$

**Case 2.** Suppose $\mathbf{v}(1) \neq \mathbf{v}'(1)$. Then, by the defintions of $r_1^{(n)}(\cdots)$ and $F_1^{(n)}(\cdots)$, it is sufficient to show that

$$\exists v_1 \in \mathbf{V}$$
$$[\langle \mathbf{v}, \mathbf{v}[\mathbf{v}'(1)/x_1] \rangle \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q' \tag{82}$$
$$\in p \,\tilde{\|}\, \mathcal{D}_1[\![\mathrm{If}(x_1 = \mathbf{v}(1), (x_1 := \mathbf{v}'(1)); (x_1 := v_1); T', (x_1 := \mathbf{v}_2(1)); \mathrm{Stop})]\!]$$
$$\Rightarrow p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \wedge q' \in p \,\tilde{\|}\, \mathcal{D}_1[\![T']\!]\ ].$$

We can establish this by choosing $v_1$ and $\mathbf{v}_2 = (v_2)$ such that

$$\begin{cases} \text{(i) } v_1 \notin \{v \in \mathbf{V} : \langle v_0, v'' \rangle \cdot \langle v', v'' \rangle \cdot \langle v'', v \rangle \in p^{[3]}\}, \\ \text{(ii) } v_1 \neq v', \\ \text{(iii) } v_1 \neq v'', \\ \text{(iv) } v_1 \notin \{v \in \mathbf{V} : \langle v', v'' \rangle \cdot \langle v'', v \rangle \in p^{[2]}\}, \end{cases}$$

and

$$\begin{cases} \text{(i) } v_2 \neq v'', \\ \text{(ii) } v_2 \neq v_1. \end{cases}$$

**Induction Step:** Assume that (80) with $i = k$ holds for some $k \in \overline{n-1}$. We will show that (80) with i $= k+1$ holds. Let $\mathbf{v} \in \mathbf{V}^n$ and $q' \in \mathbf{Q}_1$. We distinguish two cases according to whether $\mathbf{v}'(k+1) = \mathbf{v}(k+1)$.
**Case 1.** Suppose $\mathbf{v}'(k+1) = \mathbf{v}(k+1)$. Let us choose $v_3 \in \mathbf{V}$ such that

$$v_3 \notin \{v \in \mathbf{V} : \langle \mathbf{v}, \mathbf{v}[v_3/x_k] \rangle \in p^{[1]}\}, \tag{83}$$

and let $v_2$ be an arbitrary element of $\mathbf{V}$.
By the induction hypothesis, there exist $v_1 \in \mathbf{V}$, $\mathbf{v}_2 \in \mathbf{V}^k$, and $\mathbf{v}_3 \in \mathbf{V}^{k-1}$ such that

$$\begin{aligned}&\forall q' \in \mathbf{Q}_1\\&[\exists v_1 \in \mathbf{V}, \exists \mathbf{v}_2 \in \mathbf{V}^k, \exists \mathbf{v}_3 \in \mathbf{V}^{k-1}\\&\ [r_k^{(n)}(\mathbf{v}[v_3/x_k], \mathbf{v}'[k], \mathbf{v}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q'\\&\quad \in p \,\tilde{\|}\, \mathcal{D}_1[\![ F_k^{(n)}(\mathbf{v}[v_3/x_k][k], \mathbf{v}'[k], v_1, \mathbf{v}_2, \mathbf{v}_3, T') ]\!]\\&\quad \Rightarrow p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!] \ ]].\end{aligned} \tag{84}$$

Put $\overline{\mathbf{v}}_3 = \mathbf{v}_3 \cup \{\langle k, v_3 \rangle\}$ and $\overline{\mathbf{v}}_2 = \mathbf{v}_2 \cup \{\langle k+1, v_2 \rangle\}$. We will show

$$\begin{aligned}&r_{k+1}^{(n)}(\mathbf{v}, \mathbf{v}'[k+1], \overline{\mathbf{v}}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q'\\&\quad \in p \,\tilde{\|}\, \mathcal{D}_1[\![ F_{k+1}^{(n)}(\mathbf{v}[k+1], \mathbf{v}'[k+1], v_1, \overline{\mathbf{v}}_2, \overline{\mathbf{v}}_3, T') ]\!]\\&\quad \Rightarrow p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!].\end{aligned} \tag{85}$$

Suppose

$$\begin{aligned}&r_{k+1}^{(n)}(\mathbf{v}, \mathbf{v}'[k+1], \overline{\mathbf{v}}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q'\\&\quad \in p \,\tilde{\|}\, \mathcal{D}_1[\![ F_{k+1}^{(n)}(\mathbf{v}[k+1], \mathbf{v}'[k+1], v_1, \overline{\mathbf{v}}_2, \overline{\mathbf{v}}_3, T') ]\!].\end{aligned}$$

Then, by (83), the first step of $r_{k+1}^{(n)}(\mathbf{v}, \mathbf{v}'[k+1], \overline{\mathbf{v}}_3)$ cannot stem from $p$, and therefore, it must stem from $\mathcal{D}_1[\![ F_{k+1}^{(n)}(\mathbf{v}[k+1], \mathbf{v}'[k+1], v_1, \overline{\mathbf{v}}_2, \overline{\mathbf{v}}_3, T') ]\!]$. Thus one has

$$\begin{aligned}&r_k^{(n)}(\mathbf{v}[v_3/x_k], \mathbf{v}'[k], \mathbf{v}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q'\\&\quad \in p \,\tilde{\|}\, \mathcal{D}_1[\![ F_k^{(n)}(\mathbf{v}[v_3/x_k][k], \mathbf{v}'[k], v_1, \mathbf{v}_2, \mathbf{v}_3, T') ]\!].\end{aligned}$$

By this and (84), one has

$$p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!].$$

Thus one has (85).

**Case 2.** Suppose $\mathbf{v}'(k+1) \neq \mathbf{v}(k+1)$. Let us choose $v_2 \in \mathbf{V}$ such that

$$v_2 \neq \mathbf{v}'(k+1). \tag{86}$$

Also choose $v_3 \in \mathbf{V}$ such that

$$\begin{cases}\text{(i)} \ \ v_3 \notin \{v \in \mathbf{V} : \langle \mathbf{v}, \mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}] \rangle \\ \qquad \cdot \langle \mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}], \mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}][v_3/x_k] \rangle \in p^{[2]}\},\\ \text{(ii)} \ \ v_3 \notin \{v \in \mathbf{V} : \langle \mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}], \mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}][v/x_k] \rangle \in p^{[1]}\}.\end{cases} \tag{87}$$

By the induction hypothesis, there exist $v_1 \in \mathbf{V}$, $\mathbf{v}_2 \in \mathbf{V}^k$, and $\mathbf{v}_3 \in \mathbf{V}^{k-1}$ such that

$$\begin{aligned}&\forall q' \in \mathbf{Q}_1\\&[\exists v_1 \in \mathbf{V}, \exists \mathbf{v}_2 \in \mathbf{V}^k, \exists \mathbf{v}_3 \in \mathbf{V}^{k-1}\\&\ [r_k^{(n)}(\mathbf{v}[\mathbf{v}'(k+1)/x_{k+1}][v_3/x_k], \mathbf{v}'[k], \mathbf{v}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q'\\&\quad \in p \,\tilde{\|}\, \mathcal{D}_1[\![ F_k^{(n)}(\mathbf{v}[v_3/x_k][k], \mathbf{v}'[k], v_1, \mathbf{v}_2, \mathbf{v}_3, T') ]\!]\\&\quad \Rightarrow p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!] \ ]].\end{aligned} \tag{88}$$

Let $\overline{\mathbf{v}}_3 = \mathbf{v}_3 \cup \{\langle k, v_3 \rangle\}$ and $\overline{\mathbf{v}}_2 = \mathbf{v}_2 \cup \{\langle k+1, v_2 \rangle\}$. We will show

$$\begin{aligned}&r_{k+1}^{(n)}(\mathbf{v}, \mathbf{v}'[k+1], \overline{\mathbf{v}}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q'\\&\quad \in p \,\tilde{\|}\, \mathcal{D}_1[\![ F_{k+1}^{(n)}(\mathbf{v}[k+1], \mathbf{v}'[k+1], v_1, \overline{\mathbf{v}}_2, \overline{\mathbf{v}}_3, T') ]\!]\\&\quad \Rightarrow p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!].\end{aligned} \tag{89}$$

Suppose

$$\begin{aligned}&r_{k+1}^{(n)}(\mathbf{v}, \mathbf{v}'[k+1], \overline{\mathbf{v}}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q'\\&\quad \in p \,\tilde{\|}\, \mathcal{D}_1[\![ F_{k+1}^{(n)}(\mathbf{v}[k+1], \mathbf{v}'[k+1], v_1, \overline{\mathbf{v}}_2, \overline{\mathbf{v}}_3, T') ]\!].\end{aligned}$$

Let $t = \mathcal{D}_1[\![F^{(n)}_{k+1}(\mathbf{v}[k+1], \mathbf{v}'[k+1], v_1, \overline{\mathbf{v}}_2, \overline{\mathbf{v}}_3, T')]\!]$.

By (87) (i), it is impossible that both the first two steps of $r^{(n)}_{k+1}(\mathbf{v}, \mathbf{v}'[k+1], \overline{\mathbf{v}}_3)$ stem from $p$.

Also, by (86), it is impossible that the first two steps stem form $p$ and $t$, respectively.

Thus the first step must stem form $t$. Moreover, by (87) (ii), the second step cannot stem from $p$. Hence both the first two steps must stem from $t$. Thus one has

$$r^{(n)}_k(\mathbf{v}[v_3/x_k], \mathbf{v}'[k], \mathbf{v}_3) \cdot \langle \mathbf{v}', \mathbf{v}'' \rangle \cdot \langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle \cdot q' \\ \in p \,\tilde{\|}\, \mathcal{D}_1[\![F^{(n)}_k(\mathbf{v}[v_3/x_k][k], \mathbf{v}'[k], v_1, \mathbf{v}_2, \mathbf{v}_3, T')]\!].$$

By this and (84), one has

$$p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \neq \emptyset \wedge q' \in p[\langle \mathbf{v}', \mathbf{v}'' \rangle] \,\tilde{\|}\, \mathcal{D}_1[\![T']\!]\,]].$$

Thus one has (85). ■

**Proof of Lemma 13 with $\sharp$IVar $\geq 2$.**

Take $v_1 \in \mathbf{V}$, $\mathbf{v}_2 \in \mathbf{V}^n$, and $\mathbf{v}_3 \in \mathbf{V}^{n-1}$ as in (2) of the above lemma, and let $r_1 = r^{(n)}_n(\mathbf{v}_0, \mathbf{v}', \mathbf{v}_3)$ and $r_2 = (\langle \mathbf{v}'', \mathbf{v}''[v_1/x_1] \rangle)$. Then by the above lemma, one has the desired relsut. ■

## A.5 Proof of Lemma 18

The proof is similar to that of Lemma 8. By Lemma 16, it is sufficient to show

$$\forall k \geq 1, \forall p_1, \cdots, p_k, p' \in \mathbf{P}^*_2[\bigcup_{i \in \bar{k}}[p_i] \,\tilde{\|}\, p' = \bigcup_{i \in \bar{k}}[p_i \,\tilde{\|}\, p']]. \tag{90}$$

Let $\mathbf{M}^{\mathrm{dis}}_2 = ((\wp_+(\bar{k}) \to \mathbf{P}^*_2) \times \mathbf{P}^*_2 \to \mathbf{P}^*_2)$, and let $F, G \in \mathbf{M}^{\mathrm{dis}}_2$ be defined as follows:

For $I \in \wp_+(\bar{k})$, $(p_i)_{i \in I} \in (I \to \mathbf{P}^*_2)$, and $p' \in \mathbf{P}^*_2$,

$$F((p_i)_{i \in I}, p') = \bigcup_{i \in I}[p_i] \,\tilde{\|}\, p', \\ G((p_i)_{i \in I}, p') = \bigcup_{i \in I}[p_i \,\tilde{\|}\, p'].$$

By the definition of $\tilde{\|}$, one has

$$F((p_i)_{i \in I}, p') = (\bigcup_{i \in I}[p_i] \,\|\!\_\, p') \cup (p' \,\|\!\_\, \bigcup_{i \in I}[p_i]) \\ \cup(\bigcup_{i \in I}[p_i] \rhd p') \cup (p' \rhd \bigcup_{i \in I}[p_i]) \cup (\bigcup_{i \in I}[p_i] \,\#\, p'). \tag{91}$$

For $J \subseteq I$, let

$$A(J) = \{\langle \sigma, a, \sigma' \rangle : \forall i \in J[p_i[\langle \sigma, a, \sigma' \rangle] \neq \emptyset] \wedge \forall i \in (I \setminus J)[p_i[\langle \sigma, a, \sigma' \rangle] = \emptyset]\}.$$

Then as in Lemma 8, one has

$$\bigcup_{i \in I}[p_i] \,\|\!\_\, p' \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, a, \sigma' \rangle \in A(J)}[\langle \sigma, a, \sigma' \rangle \cdot F((p_i[\langle \sigma, a, \sigma' \rangle])_{i \in J}, p')]]. \tag{92}$$

Let

$$C(J) = \{\langle \sigma, c, v, \sigma' \rangle : \ \forall i \in J[p_i[\langle \sigma, c!v, \sigma \rangle] \neq \emptyset] \\ \wedge \forall i \in (I \setminus J)[p_i[\langle \sigma, c!v, \sigma \rangle] = \emptyset] \\ \wedge p'[\langle \sigma, c?v, \sigma' \rangle] \neq \emptyset \ \}.$$

Then

$$\bigcup_{i \in I}[p_i] \rhd p' \\ = \bigcup\{ \ \langle \sigma, \tau, \sigma' \rangle \cdot (\bigcup_{i \in I}[p_i][\langle \sigma, c!v, \sigma \rangle] \,\tilde{\|}\, p'[\langle \sigma, c?v, \sigma' \rangle]) : \\ \bigcup_{i \in I}[p_i][\langle \sigma, c!v, \sigma \rangle] \neq \emptyset \wedge p'[\langle \sigma, c?v, \sigma' \rangle] \neq \emptyset \ \} \\ \text{(Taking closure is omitted, since} \\ \text{ASFin}(\bigcup_{i \in I}[p_i]) \wedge \text{OVFin}(\bigcup_{i \in I}[p_i]) \wedge \text{ASFin}(p') \text{ by Lemma 17)} \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot (\bigcup_{i \in I}[p_i][\langle \sigma, c!v, \sigma \rangle] \,\tilde{\|}\, p'[\langle \sigma, c?v, \sigma' \rangle])]] \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot F((p_i[\langle \sigma, c!v, \sigma \rangle])_{i \in J}, p'[\langle \sigma, c?v, \sigma' \rangle])]]. \tag{93}$$

Let

$$C'(J) = \{\langle \sigma, c, v, \sigma' \rangle : \begin{array}{l} \forall i \in J[p_i[\langle \sigma, c?v, \sigma' \rangle] \neq \emptyset] \\ \wedge \forall i \in (I \setminus J)[p_i[\langle \sigma, c?v, \sigma' \rangle] = \emptyset] \\ \wedge p'[\langle \sigma, c!v, \sigma \rangle] \neq \emptyset \ \}. \end{array}$$

Then as (93), one has

$$\begin{array}{l} p' \triangleright \bigcup_{i \in I}[p_i] \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C'(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot F((p_i[\langle \sigma, c?v, \sigma' \rangle])_{i \in J}, p'[\langle \sigma, c!v, \sigma \rangle])]]. \end{array} \quad (94)$$

Let $\mathcal{F} : \mathbf{M}_2^{\text{dis}} \to \mathbf{M}_2^{\text{dis}}$ be defined as follows:

For $f \in \mathbf{M}_2^{\text{dis}}$,

$$\begin{array}{l} \mathcal{F}(f)((p_i)_{i \in I}, p') \\ = \ \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, a, \sigma' \rangle \in A(J)}[\langle \sigma, a, \sigma' \rangle \cdot f((p_i[\langle \sigma, a, \sigma' \rangle])_{i \in J}, p')]] \\ \quad \cup \bigcup\{\langle \sigma, a, \sigma' \rangle \cdot f((p_i)_{i \in I}, p'[\langle \sigma, a, \sigma' \rangle]) : p'[\langle \sigma, a, \sigma' \rangle] \neq \emptyset\} \\ \quad \cup \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot f((p_i[\langle \sigma, c!v, \sigma \rangle])_{i \in J}, p'[\langle \sigma, c?v, \sigma' \rangle])]] \\ \quad \cup \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C'(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot f((p_i[\langle \sigma, c?v, \sigma' \rangle])_{i \in J}, p'[\langle \sigma, c!v, \sigma \rangle])]] \\ \quad \cup (\bigcup_{i \in I}[p_i] \# p'). \end{array} \quad (95)$$

Then $\mathcal{F}$ is a contraction; by (92), (93), and (94), one has $F = \mathcal{F}(F)$, i.e., $F = \text{fix}(\mathcal{F})$.
Next, let us show that $G = \mathcal{F}(G)$.

$$\begin{array}{l} G((p_i)_{i \in I}, p') \\ = \bigcup_{i \in I}[(p_i \mathbin{⌊\!⌊} p') \cup (p' \mathbin{⌊\!⌊} p_i) \cup (p_i \triangleright p') \cup (p' \triangleright p_i) \cup (p_i \# p')] \\ = \bigcup_{i \in I}[p_i \mathbin{⌊\!⌊} p'] \cup \bigcup_{i \in I}[p' \mathbin{⌊\!⌊} p_i] \cup \bigcup_{i \in I}[p_i \triangleright p'] \cup \bigcup_{i \in I}[p' \triangleright p_i] \cup \bigcup_{i \in I}[p_i \# p']. \end{array} \quad (96)$$

As in the proof of Lemma 8, one has

$$\begin{array}{l} \bigcup_{i \in I}[p_i \mathbin{⌊\!⌊} p'] \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, a, \sigma' \rangle \in A(J)}[\langle \sigma, a, \sigma' \rangle \cdot G((p_i[\langle \sigma, a, \sigma' \rangle])_{i \in J}, p')]]. \end{array} \quad (97)$$

Moreover

$$\begin{array}{l} \bigcup_{i \in I}[p_i \triangleright p'] \\ = \bigcup_{i \in I}[\bigcup\{\langle \sigma, \tau, \sigma' \rangle \cdot (p_i[\langle \sigma, c!v, \sigma' \rangle] \mathbin{\tilde{\|}} p'[\langle \sigma, c?v, \sigma' \rangle]) : p_i[\langle \sigma, c!v, \sigma' \rangle] \neq \emptyset \wedge p'[\langle \sigma, c?v, \sigma' \rangle] \ \}] \\ = \bigcup_{i \in I}[\bigcup\{\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot (p_i[\langle \sigma, c!v, \sigma \rangle] \mathbin{\tilde{\|}} p'[\langle \sigma, c?v, \sigma' \rangle])] : J \in \wp_+(I) \wedge i \in J \ \}] \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{i \in J}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot (p_i[\langle \sigma, c!v, \sigma \rangle] \mathbin{\tilde{\|}} p'[\langle \sigma, c?v, \sigma' \rangle])]]] \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C(J)}[\bigcup_{i \in J}[\langle \sigma, \tau, \sigma' \rangle \cdot (p_i[\langle \sigma, c!v, \sigma \rangle] \mathbin{\tilde{\|}} p'[\langle \sigma, c?v, \sigma' \rangle])]]] \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot \bigcup_{i \in J}[(p_i[\langle \sigma, c!v, \sigma \rangle] \mathbin{\tilde{\|}} p'[\langle \sigma, c?v, \sigma' \rangle])]]]. \end{array}$$

Thus

$$\begin{array}{l} \bigcup_{i \in I}[p_i \triangleright p'] \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot G((p_i[\langle \sigma, c!v, \sigma \rangle])_{i \in J}, p'[\langle \sigma, c?v, \sigma' \rangle])]]. \end{array} \quad (98)$$

Similarly to the above equation, one has

$$\begin{array}{l} \bigcup_{i \in I}[p' \triangleright p_i] \\ = \bigcup_{J \in \wp_+(I)}[\bigcup_{\langle \sigma, c, v, \sigma' \rangle \in C'(J)}[\langle \sigma, \tau, \sigma' \rangle \cdot G((p_i[\langle \sigma, c?v, \sigma' \rangle])_{i \in J}, p'[\langle \sigma, c!v, \sigma \rangle])]]. \end{array} \quad (99)$$

By the definition of $\mathbin{⌊\!⌊}$, one has

$$\begin{array}{rl} \bigcup_{i \in I}[p' \mathbin{⌊\!⌊} p_i] & = \bigcup_{i \in I}[\bigcup\{\langle \sigma, a, \sigma' \rangle \cdot (p'[\langle \sigma, a, \sigma' \rangle] \parallel p_i) : p'[\langle \sigma, a, \sigma' \rangle] \neq\}] \\ & = \bigcup\{\langle \sigma, a, \sigma' \rangle \cdot \bigcup_{i \in I}[p'[\langle \sigma, a, \sigma'] \parallel p_i] : p'[\langle \sigma, a, \sigma' \rangle] \neq\}] \\ & = \bigcup\{\langle \sigma, a, \sigma' \rangle \cdot G((p_i)_{i \in I}, p'[\langle \sigma, a, \sigma']) : p'[\langle \sigma, a, \sigma' \rangle] \neq\}. \end{array} \quad (100)$$

Also by the definition of $\#$,

$$\bigcup_{i\in I}[p' \# p_i] = p' \# \bigcup_{i\in I}[p_i]. \tag{101}$$

Thus by (96), (97), (98), (99), (100), and (101), one has

$$G((p_i)_{i\in I}, p') = \mathcal{F}(G)((p_i)_{i\in I}, p').$$

Hence $G = \text{fix}(\mathcal{F}) = F$. ■

# References


[AP 86] K. Apt, G. Plotkin (1986). Countable nondeterminism and random assignment, *Journal of the Association for Computer Machinery, Vol.33, No.4*, pp. 724-767.

[AR 89] P. America, J.J.M.M. Rutten (1988). Solving reflexive domain equations in a category of complete metric spaces, *Journal of Computer and System Sciences, Vol.39, No.3*, pp. 343–375.

[BCL 85] G. Berry, P.L. Curien, J. Levy (1985). Full abstraction for sequential languages: the state of art, in *Algebraic Methods in Semantics* (M. Nivat, J.C. Reynolds, eds), pp. 90–132, Cambridge University Press.

[BHR 84] S.D. Brookes, C.A.R. Hoare, A.W. Roscoe (1984). A theory of communicating sequential processes, *Journal of the Association for Computer Machinery, Vol.31*, pp. 560–599.

[BKO 88] J.A. Bergstra, J.W. Klop, E.-R. Olderog (1988). Readies and failures in the algebra of communicating processes, *SIAM J. of Computing Vol.17, No.6*, pp. 1134–1177.

[BR 90] J.W. de Bakker, J.J.M.M. Rutten (1990). Concurrency semantics based on metric domain equations, in *Domain Theory and Computation* (G.M. Reed, A.W. Roscoe, R.F. Wachter, eds.), Oxford University Press, to appear in 1990.

[BZ 82] J.W. de Bakker, J.I. Zucker (1982). Processes and the denotational semantics of concurrency, *Information and Control Vol.54*, pp. 70–120.

[DH 83] R. De Nicola, M. Hennessy (1983). Testing equivalence and processes, *Theoretical Computer Science, Vol.34*, pp. 83–133.

[Dug 66] J. Dugundji (1966). *Topology*, Allyn and Bacon, Boston.

[Eng 77] R. Engelking (1977). *General topology*, Polish Scientific Publishers.

[Gla 90] R.J. van Glabbeek (1990). *Comparative concurrency semantics and refinement of actions*, Ph.D. Thesis, Free University of Amsterdam.

[GV 88] J.F. Groote, F. Vaandrager (1988). *Structured operational semantics and bisimulation as a congruence*, Technical Report CS-R8845, Centre for Mathematics and Computer Science, Amsterdam. (To appear in Information and Computation. Extended abstract in *Proceedings 16th ICALP, Stresa, Lecture Notes in Computer Science, Vol.372*, pp. 423–438, Springer.)

[Hen 83] M. Hennessy (1983). Synchronous and asynchronous experiments on processes, *Information and Control, Vol.59*, pp. 36–83.

[Hen 85] M. Hennessy (1985). Acceptance trees, *Journal of the Association for Computer Machinery, Vol.32*, pp. 896–928.

[Hen 88] M. HENNESSY (1988). *Algebraic theory of processes*, MIT Press.

[HP 79] M. HENNESSY, G.D. PLOTKIN (1979). Full abstraction for a simple parallel programming language, in *Proceedings 8th MFCS* (J. BEČVÁŘ ed.), *Lecture Notes in Computer Science, Vol. 74*, pp. 108–120, Springer.

[KR 88] J.N. KOK, J.J.M.M. RUTTEN (1988). Contractions in comparing concurrency semantics, in *Proc. 15th ICALP* (T. LEPISTO, A. SALOMAA, eds.), *Lecture Notes in Computer Science, Vol.317*, pp. 317–332, Springer.

[Mil 73] R. MILNER (1973). Processes: a mathematical model of computing agents, in *Proceedings of Logic Colloquium 73* (H.E. ROSE, J.C. SHEPHERDSON, eds.), pp. 157–173, North-Holland.

[Mil 77] R. MILNER (1977). Fully abstract models of typed lambda-calculi, *Theoretical Computer Science, Vol.4*, pp. 1–22.

[Mil 80] R. MILNER (1980). *A Calculus of communicating systems*, *Lecture Notes in Computer Science Vol.92*, Springer.

[Mil 85] R. MILNER (1985). Lectures on a calculus for communicating systems, in *Seminar on Concurrency* (S.D. BROOKES, A.W. ROSCOE, AND G. WINSKEL, eds.), *Lecture Notes in Computer Science Vol.197*, pp. 197–220, Springer.

[Mu 85] K. MULMULEY (1985). *Full abstraction and semantic equivalence*, Ph.D. Thesis, Report CMU-CS-85-148, Computer Science Department, Carnegie-Mellon University.

[Plo 81] G.D. PLOTKIN (1981). *A structured approach to operational semantics*, Report DAIMI FN-19, Computer Science Department, Aarhus University.

[Rut 89] J.J.M.M. RUTTEN (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.), *Lecture Notes in Computer Science Vol.354*, pp. 628–658, Springer.

[Rut 90] J.J.M.M. RUTTEN (1990). Deriving Denotational models for bisimulation from Structured Operational Semantics, in *Proceedings of the 4th Conference on Programming Concepts and Methods, April 2–5, 1990, Sea Gallilee, Israel*, pp. 148–170, IFIP.

[St 86] A. STOUGHTON, *Fully abstract models of programming languages*, Ph.D. Thesis, Report CST-40-86, Department of Computer Science, University of Edinburgh.