高階双模倣性に対する完全抽象的な距離モデル (Extended Abstract)

堀田 英一 (horita@slab.ntt.co.jp)
NTT ソフトウェア研究所

あらまし: プロセスのみではなく，前もって定められた基本型の通信チャネルや値も送受できるように Thomsen の Calculus of Higher-Order Communicating Systems (CHOCS) を拡張した言語 $\mathcal{L}$ を提案する．この言語に対して，距離空間意味論の方法を用いて表示的意味モデル $\mathcal{D}$ を構築し，$\mathcal{D}$ が高階双模倣性に対して完全抽象的であること，即ち $\mathcal{D}$ が高階双模倣性を忠実に表現することを示す．

A Fully Abstract Metric Model for Higher-Order Bisimilarity (Extended Abstract)

Eiichi Horita (horita@slab.ntt.co.jp)
NTT Software Laboratories

Abstract: Thomsen's Calculus of Higher-Order Communicating Systems (CHOCS) is extended to a language $\mathcal{L}$ that supports the transmission of channels and data values of a predefined ground type as well as the transmission of processes in the sense of CHOCS. A denotational model $\mathcal{D}$ for $\mathcal{L}$ is constructed by using the methodology of metric semantics, and it is shown that $\mathcal{D}$ is fully abstract with respect to the higher-order bisimilarity.

1 Introduction

We construct a denotational model $\mathcal{D}$ for an extension of the Calculus of Higher-Order Communicating Systems (CHOCS) of Thomsen [14], and show that $\mathcal{D}$ is fully abstract with respect to the higher-order bisimilarity $\sim$, which is the operational criterion employed in [14] for identifying higher-order communicating processes.

The need for higher-order process calculi has remarkably increased recently: for example, new type of languages for network programming such as Java and Telescript involve code passing, and thus require higher-order process calculi as their theoretical basis.

For understanding semantics of a programming language or calculus, it is important to develop its denotational semantics in addition to its operational and axiomatic semantics, because these three semantics give complementary views of the language supporting one another (cf. [15, the Preface]). The operational and axiomatic semantics of CHOCS has been well developed by Thomsen in [12, 13, 14], leaving its denotational semantics less developed: Thomsen proposed a denotational model for his higher-order calculus. The model, however, has two limitations (see the remark made just below Theorem 4.4.7 of [13]): (i) this model is shown to be fully abstract with respect to the finitary approximation $\sim_\omega$ of the higher-order bisimilarity $\sim$, but is not shown to be fully abstract with respect to $\sim$ itself; (ii) even the above full abstraction result is guaranteed only when the set of channels is finite.

The result of this paper overcomes these limitations, except that we need to impose the guardedness condition on the syntax of CHOCS.[1] By overcoming restriction (ii) above, we can easily extend CHOCS and its denotational model so that processes can pass communication channels to other processes as in [2], without imposing any restriction on the cardinality of the set of channels. In this paper, we extend CHOCS to our language $\mathcal{L}$ that supports the transmission of channels and data values of a predefined ground type (such as integers and strings) as well as the transmission of processes in the sense of CHOCS (we identify channels and data values, thus implementing a simple variant of *parametric channels* [2]). To construct our denotational model, we employ the metric methodology due to de Bakker and his colleagues [5, 4, 3, 8, 6].

Hennessy has proposed a denotational model for higher-order processes and proved the model is fully abstract with respect to two observational criteria [7]. One of his observational criteria is based on the ability of processes to perform an action in all contexts, and the other is based on a certain kind of *may testing*. The two criteria are equivalent and they are substantially coarser than the higher-order bisimilarity, the observational criterion treated in the present paper.

The rest of this paper is organized as follows. In Sec. 2, we define our higher-order process language $\mathcal{L}$, which extends CHOCS so as to support the transmission of channels and data values of a predefined ground type as well as the transmission of processes. In Sec. 3, we give the operational semantics of $\mathcal{L}$ in terms of a labeled transition system, and introduce the concept of the *higher-order*

[1] The guardedness condition, which is presented in Sec. 2, corresponds to the usual Greibach condition in formal language theory.

bisimilarity. Next, in Sec. 4, we define the denotational model $\mathcal{D}$ using the methodology of metric semantics. Finally, in Sec. 5, we establish the full abstraction of $\mathcal{D}$ with respect to the higher-order bisimilarity. Full proofs are given in the full paper [9]; this extended abstract gives only brief sketches of them.

As a preliminary to what follows, we introduce several symbols used in this paper: we denote by ω the set of natural numbers $0, 1, \cdots$. The syntactical identity is denoted by $\equiv$. The powerset of a set X is denoted by $\wp(X)$. We write "let $(x \in) X \cdots$" to introduce a set X with variable x ranging over X.

2 Language for Higher-Order Processes

Our higher-order process language $\mathcal{L}$ is based on Thomsen's calculus of higher order communicating systems (CHOCS) and extends it so as to support the transmission of channels and data values of a predefined ground type (such as integers and strings) as well as the transmission of processes in the sense of CHOCS. Below, we formally define $\mathcal{L}$ using the S-expression notation of Lisp for the convenience in focusing on abstract structures of programs.

Definition 1 Let $(v \in) \mathbf{V}$ be the set of *values* that are passed between processes and also used as (identifiers of) *communication channels*; let $(x \in) \mathcal{V}_{\mathrm{val}}$ be the set of *value variables*. We assume that a set $(E \in) \mathcal{E}$ of *value expressions* is given. We do not need to specify the syntax of $\mathcal{E}$ here, but it is assumed that elements of $\mathcal{E}$ are constructed from value variables and some operators on values and that $\mathbf{V} \subseteq \mathcal{E}$. Let $(e \in) \mathcal{E}^{\emptyset}$ be the set of value expressions that contain no value variables.

Let $(X \in) \mathcal{V}_{\mathrm{proc}}$ be the set of *process variables* and let $(\xi \in) \mathcal{V} = \mathcal{V}_{\mathrm{val}} \cup \mathcal{V}_{\mathrm{proc}}$. We define the set $(S \in) \tilde{\mathcal{L}}$ of *process expressions* by the following grammar:[2]

$$S ::= X \mid \delta \mid (!\ E\ E_1\ S_2) \mid (!\ E\ S_1\ S_2) \mid (?\ E\ x\ S) \mid (?\ E\ X\ S) \mid (\|\ S_1\ S_2) \mid (\partial\ E\ S) \mid (\mu\ X\ S). \quad (1)$$

The construct $(?\ E\ x\ \cdots)$ binds the variable x in $\cdots$, and the ones $(?\ E\ X\ \cdots)$ and $(\mu\ X\ \cdots)$ bind the variable X in $\cdots$, as $\lambda\ X$ in the λ-calculus. Thus, we can define *free variables* and *bound variables* of a process expression as in the λ-calculus. For $S \in \mathcal{L}$, let $fv(S)$ be the set of free variables in S.

We define $\mathcal{L}$ $(\subseteq \tilde{\mathcal{L}})$ to be the set of those elements of $\tilde{\mathcal{L}}$ which satisfy the following *guardedness condition*:

> If $(\mu\ X\ S')$ is a subexpression of S, then all free occurrences of X in S' are in a subexpression of S' having the form $(!\ E\ \cdots)$ or $(?\ E\ \cdots)$. (2)

For $\mathcal{X} \subseteq \mathcal{V}_{\mathrm{proc}}$, we set

$$\mathcal{L}^{\mathcal{X}} = \{S \in \mathcal{L} \mid fv(S) \subseteq \mathcal{X}\}. \quad (3)$$

Thus, $\mathcal{L}^{\emptyset}$ is the set of *closed* process expressions; elements of $\mathcal{L}^{\emptyset}$ are called *programs* and we use s as a variable ranging over $\mathcal{L}^{\emptyset}$. ∎

[2] We omit the choice operator and the renaming operator of CHOCS only for simplicity; they can be easily incorporated in our $\mathcal{L}$ and be given denotational interpretations along the lines of this paper.

3 Operational Semantics

In this section, we define a labeled transition system between elements of $\mathcal{L}^{\emptyset}$. In terms of the transition system, we define the *higher-order bisimilarity* $\sim$ (the operational criterion employed in [14] for identifying higher-order communicating processes). We introduce another relation $\simeq$ and establishes the equality between $\sim$ and $\simeq$. The second relation $\simeq$ will be conveniently used in connecting $\sim$ and the denotational model defined in Sec. 4.

The labeled transition system is defined by several rules for transitions:

Definition 2 As a preliminary to the definition of the transition system, we need a few auxiliary definitions. First, let $\mathbf{V}! = \{v! \mid v \in \mathbf{V}\}$, $\mathbf{V}? = \{v? \mid v \in \mathbf{V}\}$, and $(\gamma \in) \tilde{\mathbf{V}} = \mathbf{V}! \cup \mathbf{V}?$. We define the set $(a \in) \mathbf{A}$ of *ground labels* by $\mathbf{A} = (\tilde{\mathbf{V}} \times \mathbf{V}) \cup \{\tau\}$. By using $\mathbf{A}$, the set $(\Gamma \in) \mathbf{G}$ of *transition labels* is defined by $\mathbf{G} = \mathbf{A} \cup (\tilde{\mathbf{V}} \times \mathcal{L}^{\emptyset})$. Elements of $\tilde{\mathbf{V}} \times \mathcal{L}^{\emptyset}$ are called *first-order labels*. We use u as a variable ranging over $\mathbf{V} \cup \mathcal{L}^{\emptyset}$. For $\Gamma \in \mathbf{G} \setminus \{\tau\}$, we define $\overline{\Gamma}$ as follows:

$$\overline{\Gamma} = \begin{cases} (v?, u) & \text{if } \Gamma = (v!, u), \\ (v!, u) & \text{if } \Gamma = (v?, u). \end{cases}$$

An *evaluation function* $[\![\cdot]\!] : \mathcal{E}^{\emptyset} \to \mathbf{V}$ is assumed to be given. For $S \in \mathcal{L}$, a variable $\xi \in \mathcal{V}$ and a closed expression $t \in \mathcal{E}^{\emptyset} \cup \mathcal{L}^{\emptyset}$, we denote by $S[t/\xi]$ the result of substituting t for all free occurrences of ξ in S.

The labeled transition system $\langle \xrightarrow{\Gamma} \mid \Gamma \in \mathbf{G} \rangle$ is defined as the family of the smallest relations satisfying the following rules OUT_1 through REC.

$\mathbf{OUT_1}$: $(!\ e\ e_1\ s) \xrightarrow{([\![e]\!]!, [\![e_1]\!])} s$.

$\mathbf{OUT_2}$: $(!\ e\ s_1\ s) \xrightarrow{([\![e]\!]!, s_1)} s$.

$\mathbf{IN_1}$: For every $v \in \mathbf{V}$, $(?\ e\ x\ S) \xrightarrow{([\![e]\!]?, v)} S[v/x]$.

$\mathbf{IN_2}$: For every $s \in \mathcal{L}^{\emptyset}$,

$$(?\ e\ X\ S) \xrightarrow{([\![e]\!]?, s)} S[s/X].$$

$$\mathbf{PAR_1} : \frac{s_1 \xrightarrow{\Gamma} s_1'}{(\|\ s_1\ s_2) \xrightarrow{\Gamma} (\|\ s_1'\ s_2)}.$$

$$\mathbf{PAR}_2: \frac{s_2 \xrightarrow{\Gamma} s_2'}{(\|\ s_1\ s_2) \xrightarrow{\Gamma} (\|\ s_1\ s_2')}.$$

$$\mathbf{PAR}_3: \frac{s_1 \xrightarrow{\Gamma} s_1',\ \ s_2 \xrightarrow{\overline{\Gamma}} s_2'}{(\|\ s_1\ s_2) \xrightarrow{\tau} (\|\ s_1'\ s_2')}.$$

RES: If neither

$$\exists u[\ \Gamma = ([\![e]\!]!, u)\ ] \text{ nor } \exists u[\ \Gamma = ([\![e]\!]?, u)\ ],$$

$$\text{then } \frac{s \xrightarrow{\Gamma} s'}{(\partial\ e\ s) \xrightarrow{\Gamma} (\partial\ e\ s')}.$$

$$\mathbf{REC}: \frac{S[(\mu\ X\ S)/X] \xrightarrow{\Gamma} s'}{(\mu\ X\ S) \xrightarrow{\Gamma} s'}. \blacksquare$$

Definition 3 For a binary relation R on $\mathcal{L}^\emptyset$, let

$$\begin{aligned} R^\bullet = \{\, (\Gamma_1, \Gamma_2) \in \mathbf{G} \times \mathbf{G} \mid\ & \Gamma_1 = \Gamma_2 \in \mathbf{A} \\ & \vee \exists v, s_1, s_2[\ \Gamma_1 = (v!, s_1) \wedge \Gamma_2 = (v!, s_2) \\ & \qquad \wedge s_1\ R\ s_2\ ] \\ & \vee \exists v, s_1, s_2[\ \Gamma_1 = (v?, s_1) \wedge \Gamma_2 = (v?, s_2) \\ & \qquad \wedge s_1\ R\ s_2\ ] \}. \end{aligned}$$

A *higher-order bisimulation* is a binary relation R on $\mathcal{L}^\emptyset$ such that whenever $s_1\ R\ s_2$ then the following two propositions (4) and (5) hold:

$$\begin{aligned} & \forall \Gamma_1, \forall s_1'[\ s_1 \xrightarrow{\Gamma_1} s_1' \Rightarrow \\ & \exists \Gamma_2, \exists s_2'[\ s_2 \xrightarrow{\Gamma_2} s_2' \wedge \Gamma_1\ R^\bullet\ \Gamma_2 \wedge s_1'\ R\ s_2'\ ]]. \end{aligned} \tag{4}$$

$$\begin{aligned} & \forall \Gamma_2, \forall s_2'[\ s_2 \xrightarrow{\Gamma_2} s_2' \Rightarrow \\ & \exists \Gamma_1, \exists s_1'[\ s_1 \xrightarrow{\Gamma_1} s_1' \wedge \Gamma_1\ R^\bullet\ \Gamma_2 \wedge s_1'\ R\ s_2'\ ]]. \end{aligned} \tag{5}$$

Let $\sim$ be the union of all higher-order bisimulations. It is easy to check that $\sim$ itself is a higher-order bisimulation [10, Sect. 4.6]. Thus, $\sim$ is the largest higher-order bisimulation. We call $\sim$ the *higher-order bisimilarity*. $\blacksquare$

For the use in the next section, we inductively define another binary relation $\simeq$, which turns out to be equal to $\sim$ but is more convenient.

Definition 4 For a binary relation R on $\mathcal{L}^\emptyset$, let

$$\begin{aligned} R^\circ = \{\, (\Gamma_1, \Gamma_2) \in \mathbf{G} \times \mathbf{G} \mid\ & \\ & \Gamma_1 = \Gamma_2 \in \mathbf{A} \cup (\mathbf{V}? \times \mathcal{L}^\emptyset) \\ & \vee \exists v, s_1, s_2[\ \Gamma_1 = (v!, s_1) \wedge \Gamma_2 = (v!, s_2) \\ & \qquad \wedge s_1\ R\ s_2\ ] \}. \end{aligned}$$

Note the difference between $R^\bullet$ and R°: we have

$$(v?, s)\ \mathcal{R}^\circ\ \Gamma_2 \Rightarrow \Gamma_2 = (v?, s),$$

which does not hold when $\mathcal{R}^\circ$ is replaced by $\mathcal{R}^\bullet$.

We define a *strict higher-order bisimulation* as in Definition 3 except that we use R° instead of $R^\bullet$.

We inductively define $(\simeq_n)_{n\in\omega}$ as follows:

(i) $\simeq_0 = \mathcal{L}^\emptyset \times \mathcal{L}^\emptyset$.

(ii) For each $n \in \omega$, we define $\simeq_{n+1}$ in terms of $\simeq_n$ as follows: $s_1 \simeq_{n+1} s_2$ iff the following two propositions (6) and (7) hold:

$$\begin{aligned} & \forall \Gamma_1, s_1'[\ s_1 \xrightarrow{\Gamma_1} s_1' \Rightarrow \\ & \exists \Gamma_2, s_2'[\ s_2 \xrightarrow{\Gamma_2} s_2' \wedge \\ & \qquad \Gamma_1 \simeq_n^\circ \Gamma_2 \wedge s_1' \simeq_n s_2'\ ]]. \end{aligned} \tag{6}$$

$$\begin{aligned} & \forall \Gamma_2, s_2'[\ s_2 \xrightarrow{\Gamma_2} s_2' \Rightarrow \\ & \exists \Gamma_1, s_1'[\ s_1 \xrightarrow{\Gamma_1} s_1' \\ & \qquad \wedge \Gamma_1 \simeq_n^\circ \Gamma_2 \wedge s_1' \simeq_n s_2'\ ]]. \end{aligned} \tag{7}$$

We define $\simeq = \bigcap_{n\in\omega} [\simeq_n]$. $\blacksquare$

It is easy to check that the sequence $(\simeq_n)_{n\in\omega}$ is shrinking in that $\forall n \in \omega[\ \simeq_n \supseteq \simeq_{n+1}\ ]$. From this, we immediately obtain

$$\forall n \in \omega[\ \simeq_n^\circ \supseteq \simeq_{n+1}^\circ\ ]. \tag{8}$$

It is also easy to check that $\simeq_n$ is an equivalence relation for every $n \in \omega$, and therefore so is $\simeq$.

It turns out that $\simeq$ coincides with $\sim$:

Lemma 1 $\forall s_1, s_2 \in \mathcal{L}^\emptyset[\ s_1 \sim s_2 \Leftrightarrow s_1 \simeq s_2\ ]$. $\blacksquare$

Proof. We can prove that $\simeq\ \subseteq\ \sim$ by using the fact that $\simeq$ is a strict higher-order bisimulation and therefore is a higher-order bisimulation. The reverse relation that $\sim\ \subseteq\ \simeq$ can be established by proving that $\forall n \in \omega[\ \sim\ \subseteq \simeq_n\ ]$ by induction on n. $\blacksquare$

4 Denotational Model

We use various complete metric spaces (cms's) for constructing our denotational model $\mathcal{D}$. For a complete metric space M, every contraction F on M has a unique fixed-point by Banach's fixed-point theorem. We denote by $\mathit{fix}(F)$ the unique fixed-point of F. In this section, we define many semantic objects—such as semantic interpretations of language constructs and denotational meaning functions—as unique fixed-points of contractions on cms's of various kinds.

As a preliminary to the definition of the denotational semantic domain, we introduce basic notions and operations on cms's (we give their definitions only to be self-contained with fixed notation; refer to [3] for more explanation of these and other standard notions concerning metric topology).

Definition 5 Let A be a set, and let (M, d), (M_1, d_1), (M_2, d_2) be cms's.

(1) The set A is equipped with a metric d_A, called the *discrete metric* defined as follows: for $a_1, a_2 \in A$, let $d_A(a_1, a_2) = 0$ if $a_1 = a_2$; otherwise let $d_A(a_1, a_2) = 1$. It is easy to check that (A, d_A) is a cms.

(2) For a real number $\kappa \geq 0$, we define a cms $id_\kappa((M, d))$ as follows: the carrier of $id_\kappa((M, d))$ is M, and the distance between $x_1, x_2 \in M$ in $id_\kappa((M, d))$ is defined to be $\kappa \cdot d(x_1, x_2)$.

(3) The *disjoint sum* $(M_1, d_1) \uplus (M_2, d_2)$ of (M_1, d_1) and (M_2, d_2) is defined as follows: the carrier of $(M_1, d_1) \uplus (M_2, d_2)$ is $(\{0\} \times M_1) \cup (\{1\} \times M_2)$, and for $(i, z), (j, z') \in (\{0\} \times M_1) \cup (\{1\} \times M_2)$ we define the distance $d_+((i, z), (j, z'))$ by

$$d_+((i,z),(j,z')) = \begin{cases} 0 & \text{if } i \neq j, \\ d_1(z,z') & \text{if } i = j = 0, \\ d_2(z,z') & \text{if } i = j = 1. \end{cases}$$

(4) The *product* $(M_1, d_1) \times (M_2, d_2)$ of (M_1, d_1) and (M_2, d_2) is defined as follows: the carrier of $(M_1, d_1) \times (M_2, d_2)$ is the Cartesian product $M_1 \times M_2$. For $(x_1, y_1), (x_2, y_2) \in M_1 \times M_2$, the distance $d_*((x_1, y_1), (x_2, y_2))$ is defined to be $\max\{d_1(x_1, x_2), d_2(y_1, y_2)\}$.
Let $\times$ have higher precedence than $\uplus$.

(5) The function space $(A \to M)$ is equipped with a metric d_f defined as follows: for $f_1, f_2 \in (A \to M)$, $d_f(f_1, f_2) = \sup\{d(f_1(a), f_2(a)) |\ a \in A\}$. Suppose (A, d_A) is also a metric space and let α be a nonnegative real. A function $f : A \to M$ is called *α-Lipschitz* iff $\forall x, y \in A[\ d(f(x), f(y)) \leq \alpha \cdot d_A(x, y)\ ]$. We set $(A \to^\alpha \mathbf{P}) = \{f \in (A \to \mathbf{P}) |\ f \text{ is } \alpha\text{-Lipschitz}\}$.

(6) The *Hausdorff distance* d_H on $\wp(M)$ is defined as follows: for $X, Y \in \wp(M)$,

$$d_H(X_1, X_2) = \max\{\sup_{x \in X} \inf_{y \in Y}[\, d(x,y)\,], \sup_{y \in X} \inf_{x \in Y}[\, d(x,y)\,]\ \}.$$

Let $\wp_{cl}(M)$ be the set of closed subsets of M. Then, $(\wp_{cl}(M), d_H)$ is a cms (this fact is known as Hahn's theorem; see [5] and its errata in [4] for a proof of this fact). ∎

In the rest of this paper, we abuse notation by denoting any metric function simply by d. It will be easily inferred from the context which metric is referred to by d.

4.1 Denotational Semantic Domain

In terms of above defined operations on cms's, we define our denotational semantic domain by:

Definition 6 The following equation has a unique solution in the category cms's with their metric functions bounded by 1:

$$\mathbf{P} \cong \wp_{cl}(\mathbf{A} \times id_{\frac{1}{2}}(\mathbf{P}) \uplus \tilde{\mathbf{V}} \times id_{\frac{1}{2}}(\mathbf{P} \times \mathbf{P})), \quad (9)$$

where $\cong$ denotes that there exists an isometry from the left-hand side onto the right-hand side. (For the existence and uniqueness of the solution, refer to [5] (or [3, Chap. 10]) and [1], respectively.) We denote by $(\mathbf{P}, d)$ the unique solution of (9), and use $(\mathbf{P}, d)$ as our denotational semantic domain. ∎

Remark 1 Elements of $\mathbf{P}$ can be viewed as labeled trees with an additional structure pairing some nodes. We call such trees *higher-order communication trees*. Fig. 1 illustrates such a tree t. ∎

Let (M, d) be an arbitrary metric space. For $X \in \wp(M)$, we denote by X^{cls} the topological closure of X.

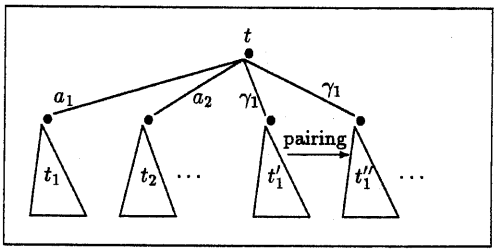

Figure 1: Higher-order communication tree.

4.2 Intermediate Model $\mathcal{O}$

In this subsection, we define an intermediate model $\mathcal{O} : \mathcal{L}^\emptyset \to \mathbf{P}$ so that the following holds for every $s \in \mathcal{L}^\emptyset$.[3]

$$\begin{aligned} &\mathcal{O}[\![s]\!] = \\ &\{(a, \mathcal{O}[\![s']\!]) |\ a \in \mathbf{A} \wedge s \xrightarrow{a} s'\} \cup \\ &\{(\gamma, (\mathcal{O}[\![\tilde{s}]\!], \mathcal{O}[\![s']\!])) |\ \gamma \in \tilde{\mathbf{V}} \wedge s \xrightarrow{(\gamma, \tilde{s})} s'\}^{cls}. \end{aligned} \quad (10)$$

Formally, we define $\mathcal{O}$ as the unique fixed-point of an higher-order mapping $\mathcal{H}$ defined as follows.

Definition 7 We define a higher-order mapping $\mathcal{H} : (\mathcal{L}^\emptyset \to \mathbf{P}) \to (\mathcal{L}^\emptyset \to \mathbf{P})$ as follows: for every $F \in (\mathcal{L}^\emptyset \to \mathbf{P})$ and $s \in \mathcal{L}^\emptyset$,

$$\begin{aligned} &\mathcal{H}(F)[\![s]\!] = \\ &\{(a, F[\![s']\!]) |\ a \in \mathbf{A} \wedge s \xrightarrow{a} s'\} \cup \\ &\{(\gamma, (F[\![\tilde{s}]\!], F[\![s']\!])) |\ \gamma \in \tilde{\mathbf{V}} \wedge s \xrightarrow{(\gamma, \tilde{s})} s'\}^{cls}. \end{aligned} \quad (11)$$ ∎

Since $\mathbf{P}$ is a cms, the space $(\mathcal{L}^\emptyset \to \mathbf{P})$ is a cms as well. It is easy to check that

$$\forall F_1, F_2 \in (\mathcal{L}^\emptyset \to \mathbf{P})[\ \tilde{d}(\mathcal{H}(F_1), \mathcal{H}(F_2)) \leq \tfrac{1}{2} \cdot \tilde{d}(F_1, F_2)\ ],$$

where $\tilde{d}$ is the metric on $(\mathcal{L}^\emptyset \to \mathbf{P})$. Thus, $\mathcal{H}$ is a contraction from $(\mathcal{L}^\emptyset \to \mathbf{P})$ to itself, and therefore it has a unique fixed-point by Banach's fixed-point theorem. We define $\mathcal{O}$ to be $fix(\mathcal{H})$. Then, $\mathcal{O}$ satisfies (10) by its definition.

As the next lemma states, the identifying power of $\mathcal{O}$ is the same as that of $\simeq$.

Lemma 2

$$\forall s_1, s_2[\ s_1 \simeq s_2 \Leftrightarrow \mathcal{O}[\![s_1]\!] = \mathcal{O}[\![s_2]\!]\ ]. \quad (12)$$ ∎

Proof. We can prove, by induction, that

$$\forall n \in \omega, \forall s_1, s_2[\ s_1 \simeq_n s_2 \Leftrightarrow d(\mathcal{O}[\![s_1]\!], \mathcal{O}[\![s_2]\!]) \leq (\tfrac{1}{2})^n\ ]. \quad (13)$$

From this, (12) immediately follows. ∎

[3] We mean by a (semantic) *model* a meaning function which maps elements of a language to elements of a semantic domain. (Note that some authors use different terminology and refers to semantic domains as models.) Here we mean by an *intermediate model* an operationally defined semantic model having $\mathbf{P}$—the denotational semantic domain—as its codomain.

4.3 Interpretations of Operators

In this subsection, we give denotational interpretations for the operators of $\mathcal{L}$; these interpretations are referred to as *semantic operations*. To define these, we slightly restrict the domain $\mathbf{P}$ to obtain a subdomain $\bar{\mathbf{P}}$ on which the semantic operations are defined.

Definition 8 **(1)** Let $\hat{\mathbf{P}} = \{\mathcal{O}[\![s]\!] \mid s \in \mathcal{L}^{\emptyset}\}$ and $\bar{\mathbf{P}} = \hat{\mathbf{P}}^{\mathrm{cls}}$. For $\mathbf{P}' \subseteq \mathbf{P}$, let

$$G(\mathbf{P}') = \{p \in \wp_{\mathrm{cl}}(\mathbf{A} \times id_{\frac{1}{2}}(\mathbf{P}') \uplus \check{\mathbf{V}} \times id_{\frac{1}{2}}(\tilde{\mathbf{P}} \times \mathbf{P}')) \mid \forall \gamma \in \mathbf{V}?, \tilde{p} \in \tilde{\mathbf{P}}, p' \in \mathbf{P}'[(\gamma, (\tilde{p}, p')) \in p \Rightarrow \forall \tilde{p}' \in \tilde{\mathbf{P}}, \exists p'' \in \mathbf{P}'[\ (c?, (\tilde{p}', p'')) \in p \wedge d(p', p'') \leq d(\tilde{p}, \tilde{p}')\ ]]\}.$$

(2) We inductively define $\langle \bar{\mathbf{P}}_n \mid n \in \omega \rangle$ as follows: **(i)** $\bar{\mathbf{P}}_0 = \mathbf{P}$; **(ii)** $\bar{\mathbf{P}}_{n+1} = G(\bar{\mathbf{P}}_n)$. Finally, we set $\bar{\mathbf{P}} = \bigcap_{n \in \omega}[\bar{\mathbf{P}}_n]$. ∎

It is easy to check that $\langle \bar{\mathbf{P}}_n \mid n \in \omega \rangle$ is shrinking and that for every n, $\bar{\mathbf{P}}_n$ is closed in $\mathbf{P}$. By using these, we can show that

$$\textbf{(i)}\ G(\bar{\mathbf{P}}) = \bar{\mathbf{P}}, \quad \textbf{(ii)}\ \bar{\mathbf{P}} \subseteq \bar{\mathbf{P}}. \tag{14}$$

The interpretation $\tilde{\delta}$ of the constant δ is defined by $\tilde{\delta} = \emptyset$. The interpretation of the output operator !, which has the most simple interpretations of all operators, is given by:

Definition 9 The operator ! is polymorphic; to interpret this operator, we define two semantic operations $\tilde{!}_v$ and $\tilde{!}'_v$ for each $v \in \mathbf{V}$.

(1) The first interpretation $\tilde{!}_v \in (\mathbf{V} \times \bar{\mathbf{P}} \to^{\frac{1}{2}} \bar{\mathbf{P}})$ is defined as follows: for each $v' \in \mathbf{V}$ and $p \in \bar{\mathbf{P}}$,

$$\tilde{!}_v(v', p) = \{((v!, v'), p)\}. \tag{15}$$

(2) The second interpretation $\tilde{!}'_v \in (\bar{\mathbf{P}} \times \bar{\mathbf{P}} \to^{\frac{1}{2}} \bar{\mathbf{P}})$ is defined as follows: for each $p, p' \in \bar{\mathbf{P}}$,

$$\tilde{!}'_v(p, p') = \{(v!, (p', p))\}. \blacksquare \tag{16}$$

Next, the interpretation of the input operator ? is given by:

Definition 10 The operator ? is polymorphic; to interpret this operator, we define two semantic operations $\tilde{?}_v$ and $\tilde{?}'_v$ for each $v \in \mathbf{V}$.

(1) The first interpretation $\tilde{?}_v \in ((\mathbf{V} \to \bar{\mathbf{P}}) \to^{\frac{1}{2}} \bar{\mathbf{P}})$ is defined as follows: for each $f \in (\mathbf{V} \to \bar{\mathbf{P}})$,

$$\tilde{?}_v(f) = \{((v?, v'), f(v')) \mid v' \in \mathbf{V}\}. \blacksquare \tag{17}$$

(2) The second interpretation $\tilde{?}'_v \in ((\bar{\mathbf{P}} \to^1 \mathbf{P}) \to^{\frac{1}{2}} \bar{\mathbf{P}})$ is defined as follows: for each $\pi \in (\bar{\mathbf{P}} \to^1 \bar{\mathbf{P}})$,

$$\tilde{?}'_v(\pi) = \{(v?, (p, \pi(p))) \mid p \in \bar{\mathbf{P}}\}. \blacksquare \tag{18}$$

The interpretations of the remaining two operators $\|$ and ∂ are defined as fixed-points of certain higher-order mappings.

First, the interpretation $\tilde{\|}$ of $\|$ is defined so that the following holds for every $p_1, p_2 \in \mathbf{P}$.

$$\tilde{\|}(p_1, p_2) = \tilde{[\![}(p_1, p_2) \cup \tilde{[\![}(p_2, p_1) \cup \tilde{\lfloor}(p_1, p_2) \cup \tilde{\lfloor}(p_2, p_1), \tag{19}$$

where we define the auxiliary operators $\tilde{[\![}$ and $\tilde{\lfloor}$ as follows: For each $p_1, p_2 \in \mathbf{P}$,

$$\tilde{[\![}(p_1, p_2) = \{(a, \tilde{\|}(p_1', p_2)) \mid a \in \mathbf{A} \wedge (a, p_1') \in p_1\}^{\mathrm{cls}} \cup \{(\gamma, (\tilde{p}_1, \tilde{\|}(p_1', p_2))) \mid (\gamma, (\tilde{p}_1, p_1')) \in p_1\}^{\mathrm{cls}}. \tag{20}$$

Also for each $p_1, p_2 \in \mathbf{P}$,

$$\tilde{\lfloor}(p_1, p_2) = \{(\tau, \tilde{\|}(p_1', p_2')) \mid \exists v, v' \in \mathbf{V}[\ ((v!, \tilde{v}), p_1') \in p_1 \wedge ((v?, \tilde{v}), p_2') \in p_2\ ]\}^{\mathrm{cls}} \cup \{(\tau, \tilde{\|}(p_1', p_2')) \mid \exists v \in \mathbf{V}, \exists \tilde{p} \in \mathbf{P}[\ (v!, (\tilde{p}, p_1')) \in p_1 \wedge (v?, (\tilde{p}, p_2')) \in p_2\ ]\}^{\mathrm{cls}}. \tag{21}$$

To be formal, we define the interpretation $\tilde{\|} \in (\bar{\mathbf{P}} \times \bar{\mathbf{P}} \to^1 \bar{\mathbf{P}})$, to be the fixed-point of a certain higher-order mapping as we defined $\mathcal{O}$ in Sec. 3 (see [9] for the formal definition). The reason for using $\bar{\mathbf{P}}$ instead of $\mathbf{P}$ is to ensure the nonexpansiveness of $\tilde{\|}$.

To interpret the construct $(\partial\ \cdots)$, we define the operation $\tilde{\partial}_v \in (\bar{\mathbf{P}} \to^1 \bar{\mathbf{P}})$ for each $v \in \mathbf{V}$; we omit the definition here.

4.4 Denotational Model $\mathcal{D}$

By means of the above defined interpretations of the operators, we define the denotational model $\mathcal{D}$ by:

Definition 11 Let $(\rho \in)\ \mathbf{R} = \{\rho \in (\mathcal{V} \to \mathbf{V} \cup \bar{\mathbf{P}}) \mid \forall x \in \mathcal{V}_{\mathrm{val}}[\ \rho(x) \in \mathbf{V}\}$; elements of $\mathbf{R}$ are called *semantic environments*. We define $\mathcal{D} : \mathcal{L} \to (\mathbf{R} \to \bar{\mathbf{P}})$ by structural induction on $S \in \mathcal{L}$ using the following clauses (0)–(8). In the clauses, let ρ range over $\mathbf{R}$.

(0) $\mathcal{D}[\![x]\!](\rho) = \rho(x)$. **(1)** $\mathcal{D}[\![\delta]\!](\rho) = \tilde{\delta}$.

(2) $\mathcal{D}[\![(!\ E\ E_1\ S_2)]\!](\rho) = \tilde{!}_{[\![E]\!](\rho)}([\![E_1]\!](\rho), \mathcal{D}[\![S_2]\!](\rho))$.

(3) $\mathcal{D}[\![(!\ E\ S_1\ S_2)]\!](\rho) = \tilde{!}'_{[\![E]\!](\rho)}(\mathcal{D}[\![S_1]\!](\rho), \mathcal{D}[\![S_2]\!](\rho))$.

(4) $\mathcal{D}[\![(?\ E\ x\ S)]\!](\rho) = \tilde{?}_{[\![E]\!](\rho)}(\lambda\ v \in \mathbf{V}.\ \mathcal{D}[\![S]\!](\rho[v/x]))$.

(5) $\mathcal{D}[\![(?\ E\ X\ S)]\!](\rho) = \tilde{?}'_{[\![E]\!](\rho)}(\lambda\ p \in \bar{\mathbf{P}}.\ \mathcal{D}[\![S]\!](\rho[p/X]))$.

(6) $\mathcal{D}[\![(\|\ S_1\ S_2)]\!](\rho) = \tilde{\|}(\mathcal{D}[\![S_1]\!](\rho), \mathcal{D}[\![S_2]\!](\rho))$.

(7) $\mathcal{D}[\![(\partial\ E\ S)]\!](\rho) = \tilde{\partial}_{[\![E]\!](\rho)}(\mathcal{D}[\![S]\!](\rho))$.

(8) $\mathcal{D}[\![(\mu\ X\ S)]\!](\rho) = fix(\lambda\ p \in \mathbf{P}.\ \mathcal{D}[\![S]\!](\rho[p/X]))$, when the function $(\lambda\ p \in \mathbf{P}.\ \mathcal{D}[\![S]\!](\rho[p/X]))$ is a contraction; otherwise, we set $\mathcal{D}[\![(\mu\ X\ S)]\!](\rho) = \emptyset$. As a matter of fact, the contractivity of $(\lambda\ p \in \mathbf{P}.\ \mathcal{D}[\![S]\!](\rho[p/X]))$ is guaranteed by the guardedness condition (2). ∎

5 Full Abstraction

To connect our two models $\mathcal{O}$ and $\mathcal{D}$, we first show that the intermediate model $\mathcal{O}$ is a homomorphism in the following sense.

Lemma 3 **(1)** $\mathcal{O}[\![\delta]\!] = \tilde{\delta}$.

(2) $\mathcal{O}[\![(!\ e\ e_1\ s_2)]\!] = \tilde{!}_{[\![e]\!]}(\mathcal{O}[\![e_1]\!], \mathcal{O}[\![s_2]\!])$.

(3) $\mathcal{O}[\![(!\ e\ s_1\ s_2)]\!] = \tilde{!}'_{[\![e]\!]}(\mathcal{O}[\![s_1]\!], \mathcal{O}[\![s_2]\!])$.

(4) $\mathcal{O}[\![(?\ e\ x\ S)]\!] = \tilde{?}_{[\![e]\!]}(\lambda\ v \in \mathbf{V}.\ \mathcal{O}[\![S[v/x]]\!])$.

(5) $\mathcal{O}[\![(?\ e\ X\ S)]\!] = \tilde{?}'_{[\![e]\!]}((\lambda\ p \in \hat{\mathbf{P}}.\ \mathcal{O}[\![S[s_p/X]]\!])^{\mathrm{cls}})$ *where* $s_p \in \mathcal{L}^{\emptyset}$ *is chosen so that* $\mathcal{O}[\![s_p]\!] = p$ *for each* $p \in \hat{\mathbf{P}}$. *We can show that* $(\lambda\ p \in \hat{\mathbf{P}}.\ \mathcal{O}[\![S[s_p/X]]\!])^{\mathrm{cls}} \in (\tilde{\mathbf{P}} \to^1 \tilde{\mathbf{P}})$ *by an analysis of the transition system.*

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

(7) $\mathcal{O}[\![(\partial\ v\ s)]\!] = \tilde{\partial}_v(\mathcal{O}[\![s]\!])$. ∎

Proof. Here we consider part (6); the other parts can be established similarly. We put

$$\kappa = \sup\{\, d(\mathcal{O}[\![(\|\ s_1\ s_2)]\!], \tilde{\|}(\mathcal{O}[\![s_1]\!], \mathcal{O}[\![s_1]\!])) \mid s_1, s_2 \in \mathcal{L}^{\emptyset}\}.$$

Then, by using (10), (19), (20) and (21), we can show that $0 \leq \kappa \leq \frac{1}{2} \cdot \kappa$. Thus, we have $\kappa = 0$. ∎

Let $(\eta \in)\ \mathbf{H} = \{\eta \in (\mathcal{V} \to \mathcal{E}^{\emptyset} \cup \mathcal{L}^{\emptyset}) \mid \forall x \in \mathcal{V}_{\mathrm{val}}[\ \eta(x) \in \mathcal{E}^{\emptyset}\ ]\}$. We call elements of $\mathbf{H}$ *syntactic environments*. For $S \in \mathcal{L}$ and $\eta \in \mathbf{H}$, let $S[\eta]$ be the result of replacing each free occurrences of each y in $fv(S)$ by $\eta(y)$. For a type-preserving partial function $\hat{\eta}$ from $\mathcal{V}$ to $\mathcal{E}^{\emptyset} \cup \mathcal{L}^{\emptyset}$, we define $S[\hat{\eta}]$ in a similar fashion. Clearly, we have $S[\eta] \in \mathcal{L}^{\emptyset}$. For $\eta \in \mathbf{H}$, let $\mathcal{O} \bullet \eta = (\lambda\ X \in \mathcal{V}_{\mathrm{proc}}.\ \mathcal{O}[\![\eta(x)]\!]) \cup (\lambda\ x \in \mathcal{V}_{\mathrm{val}}.\ [\![\eta(x)]\!])$. Clearly, we have $\mathcal{O} \bullet \eta \in \mathbf{R}$.

By means of syntactical environments, we can connect $\mathcal{O}$ and $\mathcal{D}$ as follows.

Lemma 4 *For every* $S \in \mathcal{L}$, *one has*

$$\forall \eta \in \mathbf{H}[\ \mathcal{O}[\![S[\eta]]\!] = \mathcal{D}[\![S]\!](\mathcal{O} \bullet \eta)\ ].\ \blacksquare \qquad (22)$$

Proof. This lemma can be proved by induction on the structural of $S \in \mathcal{L}$ utilizing Lemma 3. ∎

From Lemma 4, the next Corollary immediately follows.

Corollary 1 $\forall s \in \mathcal{L}^{\emptyset}[\ \mathcal{O}[\![s]\!] = \mathcal{D}[\![s]\!]\ ].$ ∎

From Lemma 1, Lemma 2 and Corollary 1, we immediately obtain the next theorem.

Theorem 1 *For every* $s_1, s_2 \in \mathcal{L}^{\emptyset}$, *one has*

$$s_1 \sim s_2 \Leftrightarrow \mathcal{D}[\![s_1]\!] = \mathcal{D}[\![s_2]\!].\ \blacksquare \qquad (23)$$

6 Concluding Remarks

We conclude this paper with several remarks about future work.

First, it remains for future research to construct models for more expressive languages. Sangiorgi's higher-order π-calculus [11] is a target for extending our model.

Another topic for future research is to construct a denotational model that is fully abstract with respect to the *weak bisimilarity* instead of strong bisimilarity treated in this paper.

References

[1] P. America and J.J.M.M. Rutten: Solving reflexive domain equations in a category of complete metric spaces, *Journal of Computer and System Sciences*, Vol. 39, No. 3, pp. 343–375, 1989.

[2] E. Astesiano and E. Zucca: Parametric channels via label expressions in CCS, in *Theoretical Computer Science, Vol. 33*, pp. 45–63, 1984.

[3] J. de Bakker and E.P. de Vink: *Control Flow Semantics*, The MIT Press, 1996.

[4] J. de Bakker and J.J.M.M. Rutten, eds.: *Ten Years of Concurrency Semantics*, World Scientific Publishing, Singapore, 1992.

[5] J. de Bakker and J.I. Zucker: Processes and the denotational semantics of concurrency, *Information and Control*, Vol. 54, pp. 70–120, 1982.

[6] F. van Breugel: *Topological Models in Comparative Semantics*, Ph.D. thesis, the Free University of Amsterdam, 1994.

[7] M. Hennessy: A fully abstract denotational model for higher-order processes, *Information and Computation*, Vol. 112, pp. 55-95, 1994.

[8] E. Horita: *Fully Abstract Models for Concurrent Languages*, Ph.D. thesis, the Free University of Amsterdam, 1993.

[9] E. Horita: *A Fully Abstract Metric Model for Higher-Order Bisimilarity*, to appear as an ECL Technical Report, NTT Software Laboratories, 1997.

[10] R. Milner: *Communication and Concurrency*, Prentice Hall International, 1989.

[11] D. Sangiorgi: From π-calculus to Higher-Order π-calculus—and back, in *Lecture Notes in Computer Science, Vol. 668*, pp. 151–166, 1993.

[12] B. Thomsen: A calculus of higher order communicating systems, in *Proceedings of POPL '89*, pp. 143–154, 1989.

[13] B. Thomsen: *Calculi for Higher Order Communicating Systems*, Ph.D. thesis, Imperial College, London University, 1990.

[14] B. Thomsen: A calculus of higher order communicating systems, Information and Computation, Vol. 116, pp. 38–57, 1995.

[15] G. Winskel: *The Formal Semantics of Programming Languages: an Introduction*, The MIT Press, 1993.