Metric Semantics for Second Order Communication

Jaco de Bakker

CWI
Department of Software Technology
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Vrije Universiteit
Department of Mathematics and Computer Science
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Franck van Breugel*

McGill University
School of Computer Science
3480 University Street, Montreal H3A 2A7, Canada

Abstract

An operational and a denotational semantics are presented for a simple imperative language. The main feature of the language is second order communication: sending and receiving of statements rather than values. The operational semantics is based on a transition system. A complete 1-bounded ultrametric space is used in the denotational semantics. In establishing the connection between the two semantics fruitful use is made of Banach's fixed point theorem, Rutten's processes as terms technique, and Van Breugel's metric transition systems.

Introduction

In recent years the study of higher order programming notions has become a central topic in the field of semantics. Seminal in this development have been two schools of research, viz that of typed λ-calculus in the area of functional programming (see, e.g., Barendregt's survey [Bar92]), and that of higher order processes in the theory of concurrency (see, e.g., the theses by Sangiorgi [San92] and Thomson [Tho90]). The aim of the present paper is to provide another perspective on this problem area by studying higher order notions embedded in the traditional setting of imperative languages.

The higher order notion we study is *second order communication*[1]. Recall that ordinary—what is also called first order—communication as, e.g., in Milner's CCS [Mil80] is expressed by the two actions $c\,!\,e$ and $c?v$, for c a channel, e some expression, and v a variable, occurring in two parallel components.[2] Synchronised execution of these actions results in the transmission of the current value of e to v. A second order variant of this is the pair of communication constructs $c\,!\,s$ and $c\,?\,x$, for c a channel, s some statement, and x a statement variable. Now the statement s, a higher order value, is passed at the moment of synchronised execution. Similar higher order notions one encounters in, e.g., Boudol's γ-calculus [Bou89], Sangiorgi's HOπ [San92]—a higher order variant of Milner, Parrow, and Walker's π-calculus [MPW92]—and Thomson's CHOCS [Tho90].

Though this higher order notion is, we hope, conceptually quite simple, a not so simple arsenal of semantic tools is necessary to model this notion operationally and denotationally, and to obtain a full picture of the relationship between the operational and denotational semantics.

*Supported by the Netherlands Organization for Scientific Research.

[1] For a discussion why it is called second order we refer the reader to Section 7.4 of Milner's [Mil91].

[2] In [Mil80] the notation $\bar{c}e$ and cv is used.

The definition of the *operational semantics* follows the customary pattern in that it is derived from some *transition system*, as advocated by Plotkin [Plo81]. In the configurations of the transition system we encounter *syntactic stores*, a second order variant of ordinary syntactic states. A syntactic state assigns to each variable its value. A syntactic store assigns to each statement variable a statement. Synchronous execution of $c\,!\,s$ and $c\,?\,x$ amounts to passing s and storing it in x. The latter is operationally modelled by assigning s to x in the syntactic store. The transition system is *finitely branching*: each configuration has only finitely many outgoing transitions. We exploit this fact in relating the operational and denotational semantics.

The *denotational semantics* employs a *complete 1-bounded ultrametric space*. This space is defined as the solution of a recursive equation. In the equation we use *semantic stores*. A semantic store assigns to each statement variable the denotation of a statement. Denotationally the synchronous execution of $c\,!\,s$ and $c\,?\,x$ is modelled by passing the denotation of s and storing it in x in the semantic store. For a large variety of languages denotational semantics based on ultrametric spaces have been developed (numerous examples are provided by De Bakker and De Vink in [BV95]). Higher order notions have been modelled denotationally by, e.g., Hennessy [Hen94], Jagadeesan and Panangaden [JP90], and Thomson [Tho90].

To link the operational and denotational semantics we introduce an intermediate semantics. Like the operational semantics the intermediate semantics is defined by means of a transition system. In the configurations of the transition system we use the (denotational) semantic stores rather than the (operational) syntactic stores. As we will see this induces the appearance of denotations of statements in the configurations. This phenomenon is known as *processes as terms* (see Rutten's [Rut92]). The presence of denotations of statements in the configurations causes that the transition system is not finitely branching—a property which is usually exploited in relating different semantics. By providing a metric on the configurations we obtain a *metric transition system* (see Van Breugel's thesis [Bre94]). This metric transition system is *compactly branching*: each configuration has a compact set of outgoing transitions. This fact is used in relating the operational and denotational semantics via the intermediate semantics.

Banach's fixed point theorem [Ban22] plays a crucial role in the present paper. It is used to define the metric space employed in the denotational semantics, various operators on this space, and the intermediate semantics. Furthermore, the operational, intermediate, and denotational semantics are related by means of Banach's theorem.

In Section 1 we introduce a simple imperative language the second order communication is couched in. The operational and denotational semantics are presented in Section 2 and 3 and are related in Section 4. In the concluding section we discuss some related issues including bisimulation and full abstractness. Appendix A contains some notions from metric topology and Banach's theorem.

Acknowledgements

The authors have benefitted from discussions with Marcello Bonsangue, Michele Boreale, Uffe Engberg, Furio Honsell, Marina Lenisa, Vincent van Oostrom, Prakash Panangaden, Jan Rutten, and Davide Sangiorgi.

1 Language definition

We present a simple imperative language with second order communication as main construction. Besides second order communication the language contains assignment statements, sequential composition, conditional statements, parallel composition, and statement calls. The syntax of the language is given in BNF. The basic components are

- a set $(v \in)$ *Var*[3] of *variables*,
- a set $(e \in)$ *Exp* of *expressions*,
- a set $(b \in)$ *BExp* of *Boolean expressions*,

[3] Throughout this paper we use the notation $(x \in) X$ for the introduction of a set or space X with typical elements x, x', x_1, ...

- a set $(c \in)$ *Chan* of *channels*, and
- a set of $(x \in)$ *SVar* of *statement variables*.

We assume a simple syntax for the sets of expressions and Boolean expressions.

Definition 1.1 The set $(s \in)$ *Stat* of *statements* is defined by

$$s ::= v := e \mid s\,;s \mid \textbf{\textit{if}}\ b\ \textbf{\textit{then}}\ s\ \textbf{\textit{else}}\ s\ \textbf{\textit{fi}} \mid s \parallel s \mid c\,!\,s \mid c\,?\,x \mid x.$$

For this language we present an operational and a denotational semantics in the following two sections.

2 Operational semantics

The operational semantics is defined by means of a transition system. In the configurations of the transition system we encounter

- statements,
- the *empty statement* E indicating successful termination,
- syntactic states[4],
- syntactic stores[4], and
- syntactic communications[4].

Definition 2.1 The set $(\bar{s} \in)$ $Stat_{\mathrm{E}}$ is defined by

$$\bar{s} ::= \mathrm{E} \mid s.$$

A syntactic state is used to store and retrieve the values of the variables. It assigns to each variable its value. Let $(\alpha \in)$ *Val* be a set of *values*.

Definition 2.2 The set $(\sigma \in)$ *SynState* of *syntactic states* is defined by

$$SynState = Var \to Val.$$

Storing the value α for the variable v in the syntactic state σ gives rise to the syntactic state $\sigma\,\{\alpha/v\}$ defined by

$$\sigma\,\{\alpha/v\}\,(w) = \begin{cases} \alpha & \text{if } v = w \\ \sigma\,(w) & \text{if } v \neq w \end{cases}$$

We use a syntactic store to administrate which statements the statement variables are assigned to.

Definition 2.3 The set $(\theta \in)$ *SynStore* of *syntactic stores* is defined by

$$SynStore = SVar \to Stat.$$

Assigning the statement s to the statement variable x in the syntactic store θ gives rise to the syntactic store $\theta\,\{s/x\}$ defined by

$$\theta\,\{s/x\}\,(y) = \begin{cases} s & \text{if } x = y \\ \theta\,(y) & \text{if } x \neq y \end{cases}$$

A syntactic communication $c\,!\,s$ is used to model the willingness to send the statement s along the channel c. To model the willingness to receive a statement along the channel c and store it in the statement variable x we use the syntactic communication $c\,?\,x$.

Definition 2.4 The set $(\pi \in)$ *SynCom* of *syntactic communications* is defined by

$$\pi ::= c\,!\,s \mid c\,?\,x.$$

In the configurations of the transition system a statement is accompanied by a syntactic action: a syntactic state and a syntactic store, or a syntactic communication[5].

[4] In Section 3 we introduce semantic states, semantic stores, and semantic communications to define the denotational semantics.

[5] In this case there is no need to keep the syntactic state and the syntactic store (see rule (6) of Definition 2.6).

Definition 2.5 The set $(\rho \in)\, SynAct$ of *syntactic actions* is defined by

$$\rho ::= (\sigma, \theta) \mid \pi.$$

We assume that the evaluation of an expression and a Boolean expression always terminates and delivers a value and *true* or *false*, respectively. This is expressed by means of the functions

$$\mathcal{E} : Exp \to SynState \to Val$$

and

$$\mathcal{B} : BExp \to SynState \to \{true, false\}.$$

With $\mathcal{E}[\![e]\!](\sigma)$ we denote the value of the expression e in the syntactic state σ. The value of the Boolean expression b in the syntactic state σ is denoted by $\mathcal{B}[\![b]\!](\sigma)$.

The transition relation is defined by means of a collection of axioms and rules.

Definition 2.6 The transition relation $\to_1$ is defined as the smallest subset of

$$(Stat_{\mathrm{E}} \times SynAct) \times (Stat_{\mathrm{E}} \times SynAct)$$

satisfying the following axioms and rules.

(1) $[v := e,\, \sigma,\, \theta] \to_1 [\mathrm{E},\, \sigma\{\alpha/v\},\, \theta]$, where $\alpha = \mathcal{E}[\![e]\!](\sigma)$

(2) $$\frac{[s_1,\, \rho] \to_1 [\bar{s}_1,\, \rho']}{[s_1 \,;\, s_2,\, \rho] \to_1 [\bar{s}_1 \,;_{\mathrm{E}}\, s_2,\, \rho']}$$, where $\bar{s}_1 \,;_{\mathrm{E}}\, \bar{s}_2 = \begin{cases} \bar{s}_2 & \text{if } \bar{s}_1 = \mathrm{E} \\ \bar{s}_1 \,;\, \bar{s}_2 & \text{if } \bar{s}_1 \neq \mathrm{E} \end{cases}$

(3) $$\frac{[s_1,\, \sigma,\, \theta] \to_1 [\bar{s}_1,\, \rho]}{[\textbf{\textit{if}}\ b\ \textbf{\textit{then}}\ s_1\ \textbf{\textit{else}}\ s_2\ \textbf{\textit{fi}},\, \sigma,\, \theta] \to_1 [\bar{s}_1,\, \rho]}$$, if $\mathcal{B}[\![b]\!](\sigma) = true$

(4) $$\frac{[s_2,\, \sigma,\, \theta] \to_1 [\bar{s}_2,\, \rho]}{[\textbf{\textit{if}}\ b\ \textbf{\textit{then}}\ s_1\ \textbf{\textit{else}}\ s_2\ \textbf{\textit{fi}},\, \sigma,\, \theta] \to_1 [\bar{s}_2,\, \rho]}$$, if $\mathcal{B}[\![b]\!](\sigma) = false$

(5) $$\frac{[s_1,\, \rho] \to_1 [\bar{s}_1,\, \rho']}{\begin{array}{l}[s_1 \parallel s_2,\, \rho] \to_1 [\bar{s}_1 \parallel_{\mathrm{E}} s_2,\, \rho'] \\ [s_2 \parallel s_1,\, \rho] \to_1 [s_2 \parallel_{\mathrm{E}} \bar{s}_1,\, \rho']\end{array}}$$, where $\bar{s}_1 \parallel_{\mathrm{E}} \bar{s}_2 = \begin{cases} \mathrm{E} & \text{if } \bar{s}_1 = \mathrm{E} \text{ and } \bar{s}_2 = \mathrm{E} \\ \bar{s}_2 & \text{if } \bar{s}_1 = \mathrm{E} \text{ and } \bar{s}_2 \neq \mathrm{E} \\ \bar{s}_1 & \text{if } \bar{s}_1 \neq \mathrm{E} \text{ and } \bar{s}_2 = \mathrm{E} \\ \bar{s}_1 \parallel \bar{s}_2 & \text{if } \bar{s}_1 \neq \mathrm{E} \text{ and } \bar{s}_2 \neq \mathrm{E} \end{cases}$

(6) $$\frac{[s_1,\, \sigma,\, \theta] \to_1 [\bar{s}_1,\, c\,!\,s] \qquad [s_2,\, \sigma,\, \theta] \to_1 [\bar{s}_2,\, c\,?\,x]}{\begin{array}{l}[s_1 \parallel s_2,\, \sigma,\, \theta] \to_1 [\bar{s}_1 \parallel_{\mathrm{E}} \bar{s}_2,\, \sigma,\, \theta\{s/x\}] \\ [s_2 \parallel s_1,\, \sigma,\, \theta] \to_1 [\bar{s}_2 \parallel_{\mathrm{E}} \bar{s}_1,\, \sigma,\, \theta\{s/x\}]\end{array}}$$

(7) $[c\,!\,s,\, \sigma,\, \theta] \to_1 [\mathrm{E},\, c\,!\,s]$

(8) $[c\,?\,x,\, \sigma,\, \theta] \to_1 [\mathrm{E},\, c\,?\,x]$

(9) $[x,\, \sigma,\, \theta] \to_1 [\theta(x),\, \sigma,\, \theta]$

A transition

$$[s,\, \sigma,\, \theta] \to_1 [\bar{s},\, \sigma',\, \theta']$$

denotes that the statement s in the (current) syntactic state σ and syntactic store θ can perform a computation step resulting in the statement $\bar{s}$ and the (possibly updated) syntactic state σ' and syntactic store θ'. A transition

$$[s_1,\, \sigma,\, \theta] \to_1 [\bar{s}_1,\, c\,!\,s]$$

denotes that the statement s_1 in the syntactic state σ and syntactic store θ is willing to send the statement s along the channel c. Since communication is synchronous, the statement s can only been sent if there is a

statement s_2 in parallel with s_1 in the same syntactic state and syntactic store willing to receive a statement along the channel c (and storing it in some statement variable x) denoted by

$$[s_2,\, \sigma,\, \theta] \rightarrow_1 [\bar{s}_2,\, c\, ?\, x].$$

The transition system is *finitely branching*: every configuration has only finitely many outgoing transitions, i.e., for all $\bar{s} \in Stat_{\mathrm{E}}$ and $\rho \in SynAct$, the set

$$\mathcal{FB}([\bar{s},\, \rho]) = \{\, [\bar{s}',\, \rho'] \mid [\bar{s},\, \rho] \rightarrow_1 [\bar{s}',\, \rho'] \,\}$$

is finite. This property is one of the ingredients of the proof relating the operational and denotational semantics (see Property 4.18 and 4.20). An alternative formulation[6] is presented in

Property 2.7 *The function* $\mathcal{FB} : Stat_{\mathrm{E}} \times SynAct \rightarrow \mathcal{P}\,(Stat_{\mathrm{E}} \times SynAct)$ *defined by*

$$\mathcal{FB}([\bar{s},\, \rho]) = \{\, [\bar{s}',\, \rho'] \mid [\bar{s},\, \rho] \rightarrow_1 [\bar{s}',\, \rho'] \,\}$$

is an element of $Stat_{\mathrm{E}} \times SynAct \rightarrow \mathcal{P}_f\,(Stat_{\mathrm{E}} \times SynAct)$.

Proof By structural induction on $\bar{s}$. □

In the operational semantics we collect successive transitions, i.e. sequences of configurations connected by transitions. We do not consider configurations modelling (unsuccessful) communication attempts, i.e. configurations of the form $[\bar{s},\, c\, !\, s]$ or $[\bar{s},\, c\, ?\, x]$. We extract the syntactic states and syntactic stores from the configurations. In this way we obtain sequences of pairs, each pair consisting of a syntactic state and a syntactic store. We distinguish three types of sequences:

- finite sequences modelling successfully terminating computations (the final configuration is of the form $[\mathrm{E},\, \sigma,\, \theta]$),
- finite sequences followed by a δ modelling deadlocking computations (the final configuration is of the form $[s,\, \sigma,\, \theta]$ and can only make unsuccessful communication attempts), and
- infinite sequences modelling nonterminating computations.

Definition 2.8 The set $(w \in)\,(SynState \times SynStore)^{\infty}_{\delta}$ is defined by

$$\begin{aligned}&(SynState \times SynStore)^{\infty}_{\delta}\\ &\quad = \quad (SynState \times SynStore)^{*} \cup (SynState \times SynStore)^{*} \cdot \{\delta\} \cup (SynState \times SynStore)^{\omega}.\end{aligned}$$

Given a statement s and an (initial) syntactic state σ and syntactic store θ, the operational semantics $\mathcal{O}$ gives us the set $\mathcal{O}\,[\![s]\!](\sigma, \theta)$ of sequences modelling all possible computations of s started in σ and θ.

Definition 2.9 The *operational semantics*

$$\mathcal{O} : Stat \rightarrow (SynState \times SynStore) \rightarrow \mathcal{P}\,((SynState \times SynStore)^{\infty}_{\delta})$$

is defined by

$$\begin{aligned}&\mathcal{O}\,[\![s]\!](\sigma,\theta)\\ &= \{\, (\sigma_1,\theta_1)(\sigma_2,\theta_2)\ldots(\sigma_n,\theta_n) \;\mid [s,\, \sigma,\, \theta] \rightarrow_1 [s_1,\, \sigma_1,\, \theta_1] \rightarrow_1 \cdots \rightarrow_1 [\mathrm{E},\, \sigma_n,\, \theta_n] \,\} \cup\\ &\quad \{\, (\sigma_1,\theta_1)(\sigma_2,\theta_2)\ldots(\sigma_n,\theta_n)\delta \mid [s,\, \sigma,\, \theta] \rightarrow_1 [s_1,\, \sigma_1,\, \theta_1] \rightarrow_1 \cdots \rightarrow_1 [s_n,\, \sigma_n,\, \theta_n] \text{ deadlocks} \,\} \cup\\ &\quad \{\, (\sigma_1,\theta_1)(\sigma_2,\theta_2)\ldots \;\mid [s,\, \sigma,\, \theta] \rightarrow_1 [s_1,\, \sigma_1,\, \theta_1] \rightarrow_1 \cdots \,\},\end{aligned}$$

where $[s,\, \sigma,\, \theta]$ *deadlocks* if $[s,\, \sigma,\, \theta] \rightarrow_1 [\bar{s},\, \sigma',\, \theta']$ for no $\bar{s} \in Stat_{\mathrm{E}}$, $\sigma' \in SynState$, and $\theta' \in SynStore$.

The operational semantics is not compositional with respect to parallel composition as is shown in the following examples (cf. [Mil93]). The operational semantics is compositional with respect to all the other operators.

[6] We present this alternative formulation as it can be generalized more conveniently (see Property 4.7).

Example 2.10 We have that

$$\mathcal{O}[\![v := 1 \,;\, v := 2]\!] = \mathcal{O}[\![v := 1 \,;\, v := v + 1]\!]$$

but

$$\mathcal{O}[\![(v := 1 \,;\, v := 2) \parallel v := 3]\!] \neq \mathcal{O}[\![(v := 1 \,;\, v := v + 1) \parallel v := 3]\!].$$

Also

$$\mathcal{O}[\![c \,?\, x]\!] = \mathcal{O}[\![c \,!\, s]\!]$$

but

$$\mathcal{O}[\![c \,?\, x \parallel c \,!\, s]\!] \neq \mathcal{O}[\![c \,!\, s \parallel c \,!\, s]\!].$$

The above example shows us that the operational semantics is not compositional because it does not model

- changes of the syntactic state and syntactic store caused by the environment (a statement in parallel), and
- communication attempts.

If we extend the axioms for the send and receive statement

(7′) $[c \,!\, s,\, \rho] \rightarrow_1 [\mathrm{E},\, c \,!\, s]$

(8′) $[c \,?\, x,\, \rho] \rightarrow_1 [\mathrm{E},\, c \,?\, x]$

and also model (configurations denoting) communication attempts (and changes of the syntactic state and syntactic store), then the modified operational semantics $\mathcal{O}'$ still lacks compositionality.

Example 2.11 We have that

$$\mathcal{O}'[\![c_1 \,?\, x \,;\, (c_2 \,?\, x + c_3 \,?\, x)]\!] = \mathcal{O}'[\![(c_1 \,?\, x \,;\, c_2 \,?\, x) + (c_1 \,?\, x \,;\, c_3 \,?\, x)]\!]$$

but

$$\mathcal{O}'[\![(c_1 \,?\, x \,;\, (c_2 \,?\, x + c_3 \,?\, x)) \parallel (c_1 \,!\, s \,;\, c_2 \,!\, s)]\!] \neq \mathcal{O}'[\![((c_1 \,?\, x \,;\, c_2 \,?\, x) + (c_1 \,?\, x \,;\, c_3 \,?\, x)) \parallel (c_1 \,!\, s \,;\, c_2 \,!\, s)]\!].$$

The modified operational semantics is not compositional, because it does not record the branching points of the transition system (caused by + and $\parallel$).

3 Denotational semantics

We need a space containing more structure than the sets of sequences used in the operational semantics to give a denotational semantics. In Definition 3.5 we introduce the complete (1-bounded ultrametric) space $\mathbb{P}$ (see Definition A.1 and A.5). The elements of this space can be viewed as tree like objects. It will turn out that this space is rich enough to model parallel composition (and all other constructs) compositionally. The denotational semantics assigns to each statement an element of $\mathbb{P}$.

In the definition of the space $\mathbb{P}$ we encounter the spaces $SemState$, $SemStore$, $SemCom$, and $SemAct$—the semantic counterparts of the sets $SynState$, $SynStore$, $SynCom$, and $SynAct$, respectively. Before defining these spaces we first turn the sets Var, Val, $SVar$, $Chan$, and $\{\mathrm{E}\}$ into spaces by endowing them with the discrete metric (see Definition A.2).

The only difference between syntactic states and semantic states is that the latter are endowed with a metric (see Definition A.8) and the former are not.

Definition 3.1 The space ($\varsigma \in$) $SemState$ of *semantic states* is defined by

$$SemState = Var \rightarrow Val.$$

A semantic store assigns to each statement variable the denotation of a statement rather than a statement as a syntactic store does.

Definition 3.2 The space $(\vartheta \in)\ \mathit{SemStore}$ of *semantic stores* is defined by

$$\mathit{SemStore} = \mathit{SVar} \to \tfrac{1}{2} \cdot I\!P.$$

The role of the $\frac{1}{2}\cdot$ (see Definition A.3) in the above definition is discussed later. It will turn out that the $\frac{1}{2}\cdot$ is essential both in the definition of the space $I\!P$ and the denotational semantics $\mathcal{D}$.

Instead of sending statements as we did operationally, denotationally we send denotations of statements.

Definition 3.3 The space $(\varpi \in)\ \mathit{SemCom}$ of *semantic communications* is defined by

$$\mathit{SemCom} = (\mathit{Chan} \times \tfrac{1}{2} \cdot I\!P) + (\mathit{Chan} \times \mathit{SVar}).$$

In the above definition we use the operations $\times$ and $+$ as introduced in Definition A.3. Instead of $(c, \bar{p})$ and (c, x) we write $c\,!\,\bar{p}$ and $c\,?\,x$, respectively. Forgetting the metric for a moment, we have

$$\mathit{SemCom} = \{\, c\,!\,\bar{p} \mid c \in \mathit{Chan} \wedge \bar{p} \in I\!P \,\} \cup \{\, c\,?\,x \mid c \in \mathit{Chan} \wedge x \in \mathit{SVar} \,\}$$

and

$$\mathit{SynCom} = \{\, c\,!\,s \mid c \in \mathit{Chan} \wedge s \in \mathit{Stat} \,\} \cup \{\, c\,?\,x \mid c \in \mathit{Chan} \wedge x \in \mathit{SVar} \,\}.$$

In Definition 2.5 we defined the set of syntactic actions by

$$\mathit{SynAct} = (\mathit{SynState} \times \mathit{SynStore}) \cup \mathit{SynCom}.$$

Its semantic counterpart is presented in the following definition.

Definition 3.4 The space $(\varrho \in)\ \mathit{SemAct}$ of *semantic actions* is defined by

$$\mathit{SemAct} = (\mathit{SemState} \times \mathit{SemStore}) + \mathit{SemCom}.$$

Note that we use ρ to range over syntactic actions and ϱ to range over semantic actions. Similarly, σ, θ, and π range over syntactic states, syntactic stores, and syntactic communications and ς, ϑ, and ϖ range over semantic states, semantic stores, and semantic communications.

As we have seen in the previous section, in order to be compositional the space $I\!P$ should record

- changes of the semantic state and semantic store caused by the environment,
- communication attempts, and
- branching points.

In the definition of the space $I\!P$ we use the operations $\to^1$ (see Definition A.9) and $\mathcal{P}_{nc}$ (see Definition A.7).

Definition 3.5 The space $(\bar{p} \in)\ I\!P$ is defined by the equation

$$I\!P \cong (\mathit{SemState} \times \mathit{SemStore} \to^1 \mathcal{P}_{nc}(\mathit{SemAct} \times \tfrac{1}{2} \cdot I\!P)) + \{\mathrm{E}\}.$$

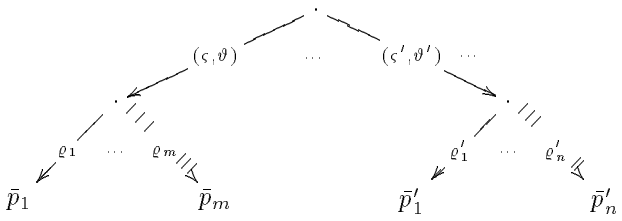
The elements of the space $I\!P$ can be viewed as tree like objects. We distinguish the following two subspaces of $I\!P$.

- $\{\mathrm{E}\}$: The semantic entity E we use to model successful termination. It can be seen as the empty tree consisting of one node and no edges.

- $(p \in)\ \mathbb{P} \setminus \{\mathrm{E}\}$: Let, for $\varsigma,\ \varsigma' \in \mathit{SemState}$ and $\vartheta,\ \vartheta' \in \mathit{SemStore}$,

$$\begin{aligned} p\,(\varsigma,\vartheta) &= \{\langle \varrho_1, \bar{p}_1\rangle, \ldots \langle \varrho_m, \bar{p}_m\rangle\} \\ p\,(\varsigma',\vartheta') &= \{\langle \varrho'_1, \bar{p}'_1\rangle, \ldots \langle \varrho'_n, \bar{p}'_n\rangle\} \end{aligned}$$

The semantic entity p can be viewed as the labelled tree



In the above picture the upper level of branching is due to the functional nature of p. It records the change of the semantic state and semantic store caused by the environment. The lower level of branching stems from the set structure of $p\,(\varsigma,\vartheta)$ and $p\,(\varsigma',\vartheta')$. It records the branching points. The labels at the lower level are semantic actions. This allows us to model communication attempts. For example, if the environment changes the semantic state to ς and the semantic store to ϑ, then the semantic actions $\varrho_1, \ldots, \varrho_m$ are possible followed by $\bar{p}_1, \ldots, \bar{p}_m$, respectively.

Note that we use the convention that p ranges over $\mathbb{P} \setminus \{\mathrm{E}\}$ and $\bar{p}$ ranges over $\mathbb{P}$.

Let us explain how to solve the above equation. (Patience, the development of the denotational semantics resumes shortly.) The equation is of the form

$$\mathbb{P} \cong F\,(\mathbb{P})$$

with F being an operation assigning to each (nonempty complete) space another (nonempty complete) space. A solution of the equation is a (nonempty complete) space being isometric (see Definition A.11 to its F-image. We shall treat the isometries as identities and thus elide their use. They can be put in without any difficulties, but will clutter up the presentation. To conclude that the equation has a (unique) solution (up to isometry) we exploit the theory developed by America and Rutten in [AR89]. As is shown in that paper, the operation F can be extended to a functor F on a suitable category of (nonempty complete) spaces. Since the equation is of the form

$$\mathbb{P} \cong \ldots \mathbb{P} \ldots \to^1 \ldots \mathbb{P} \ldots,$$

i.e. there are both positive and negative[7] occurrences of $\mathbb{P}$ in the equation, nonexpansive embedding-projection pairs are used as arrows in the category. The arrows are such that the equation has a (unique) solution if and only if the functor has a (unique) fixed point. In Theorem 4.4 of [AR89] it is shown that a (locally contractive[8] and) contractive functor has a (unique) fixed point. The proof of this theorem relies on Banach's theorem (see Theorem A.13). The functor F satisfies both conditions. If we would have left out one of the $\frac{1}{2}$'s in the above definitions, the obtained functor would neither have been locally contractive nor contractive any more.

In order to define the denotational semantics we have to introduce for each syntactic operator a semantic counterpart. Apart from the semantic sequential composition and parallel composition, the semantic operators are defined straightforwardly (see Definition 3.7). The semantic sequential composition and parallel composition are defined as the unique fixed point (Banach's theorem) of a contractive function (see Definition A.12) from the nonempty complete space $\mathbb{P} \times \mathbb{P} \to^1 \mathbb{P}$ to itself. We only give the equations characterizing the semantic operators. For the details we refer the reader to [ABKR89, KR90] where various related semantic operators are defined in this way.

[7] An occurrence of $\mathbb{P}$ is negative if it is to the left, hereditarily, of an odd number of $\to^1$'s.

[8] This terminology is taken from [RT92]. In [AR89] hom-contractive is used instead.

Definition 3.6 The operator ; is the unique function $; : I\!P \times I\!P \to^1 I\!P$ satisfying

$$\bar{p}_1 ; \bar{p}_2 = \begin{cases} \bar{p}_2 & \text{if } \bar{p}_1 = \text{E} \\ \lambda(\varsigma,\vartheta).\{\, \langle \varrho, \bar{p}_1' ; \bar{p}_2 \rangle \mid \langle \varrho, \bar{p}_1' \rangle \in \bar{p}_1(\varsigma,\vartheta) \,\} & \text{if } \bar{p}_1 \neq \text{E} \end{cases}$$

The operator $\|$ is the unique function $\| : I\!P \times I\!P \to^1 I\!P$ satisfying

$$\bar{p}_1 \,\|\, \bar{p}_2 = \begin{cases} \text{E} & \text{if } \bar{p}_1 = \text{E} \text{ and } \bar{p}_2 = \text{E} \\ \bar{p}_2 & \text{if } \bar{p}_1 = \text{E} \text{ and } \bar{p}_2 \neq \text{E} \\ \bar{p}_1 & \text{if } \bar{p}_1 \neq \text{E} \text{ and } \bar{p}_2 = \text{E} \\ \bar{p}_1 \mathbin{\lfloor\!\lfloor} \bar{p}_2 + \bar{p}_2 \mathbin{\lfloor\!\lfloor} \bar{p}_1 + \bar{p}_1 \mathbin{\lfloor} \bar{p}_2 + \bar{p}_2 \mathbin{\lfloor} \bar{p}_1 & \text{if } \bar{p}_1 \neq \text{E} \text{ and } \bar{p}_2 \neq \text{E} \end{cases}$$

where

$$p_1 \mathbin{\lfloor\!\lfloor} p_2 = \lambda(\varsigma,\vartheta).\{\, \langle \varrho, \bar{p}_1 \,\|\, p_2 \rangle \mid \langle \varrho, \bar{p}_1 \rangle \in p_1(\varsigma,\vartheta) \,\}$$

and

$$p_1 \mathbin{\lfloor} p_2 = \lambda(\varsigma,\vartheta).\{\, \langle (\varsigma, \vartheta\{\bar{p}/x\}), \bar{p}_1 \,\|\, \bar{p}_2 \rangle \mid \langle c \,!\, \bar{p}, \bar{p}_1 \rangle \in p_1(\varsigma,\vartheta) \wedge \langle c \,?\, x, \bar{p}_2 \rangle \in p_2(\varsigma,\vartheta) \,\}$$

and

$$p_1 + p_2 = \lambda(\varsigma,\vartheta).p_1(\varsigma,\vartheta) \cup p_2(\varsigma,\vartheta).$$

Having introduced the space $I\!P$ and the (nontrivial) semantic operators we are ready to give the denotational semantics.

Definition 3.7 The *denotational semantics* $\mathcal{D} : Stat \to I\!P$ is defined by

$$\begin{aligned}
\mathcal{D}[\![v := e]\!] &= \lambda(\varsigma,\vartheta).\{\langle (\varsigma\{\alpha/v\}, \vartheta), \text{E}\rangle\}, \text{ where } \alpha = \mathcal{E}[\![e]\!](\varsigma) \\
\mathcal{D}[\![s_1 ; s_2]\!] &= \mathcal{D}[\![s_1]\!] ; \mathcal{D}[\![s_2]\!] \\
\mathcal{D}[\![\textbf{\textit{if}}\ b\ \textbf{\textit{then}}\ s_1\ \textbf{\textit{else}}\ s_2\ \textbf{\textit{fi}}]\!] & \\
&= \lambda(\varsigma,\vartheta).\begin{cases} \mathcal{D}[\![s_1]\!](\varsigma,\vartheta) \text{ if } \mathcal{B}[\![b]\!](\varsigma) = \mathit{true} \\ \mathcal{D}[\![s_2]\!](\varsigma,\vartheta) \text{ if } \mathcal{B}[\![b]\!](\varsigma) = \mathit{false} \end{cases} \\
\mathcal{D}[\![s_1 \,\|\, s_2]\!] &= \mathcal{D}[\![s_1]\!] \,\|\, \mathcal{D}[\![s_2]\!] \\
\mathcal{D}[\![c \,!\, s]\!] &= \lambda(\varsigma,\vartheta).\{\langle c \,!\, \mathcal{D}[\![s]\!], \text{E}\rangle\} \\
\mathcal{D}[\![c \,?\, x]\!] &= \lambda(\varsigma,\vartheta).\{\langle c \,?\, x, \text{E}\rangle\} \\
\mathcal{D}[\![x]\!] &= \lambda(\varsigma,\vartheta).\{\langle (\varsigma,\vartheta), \vartheta(x)\rangle\}
\end{aligned}$$

Of course one has to check that, for all $s \in Stat$, $\mathcal{D}[\![s]\!] \in I\!P$. This can be verified by structural induction on s. We only consider the case $s = x$ as it shows us the importance of the positioning of the $\frac{1}{2}\cdot$'s in the above definitions. Obviously, for all $\varsigma \in SemState$ and $\vartheta \in SemStore$, the set $\{((\varsigma,\vartheta),\vartheta(x))\}$ is compact. Let $\varsigma,\, \varsigma' \in SemState$ and $\vartheta,\, \vartheta' \in SemStore$. We have that

$$\begin{aligned}
& d(\{((\varsigma,\vartheta),\vartheta(x))\}, \{((\varsigma',\vartheta'),\vartheta'(x))\}) \\
&= d(((\varsigma,\vartheta),\vartheta(x)),((\varsigma',\vartheta'),\vartheta'(x))) \\
&= \max\{d((\varsigma,\vartheta),(\varsigma',\vartheta')), \tfrac{1}{2}\cdot d(\vartheta(x),\vartheta'(x))\} \\
&\leq \max\{d((\varsigma,\vartheta),(\varsigma',\vartheta')), d(\vartheta,\vartheta')\} \quad \text{[see below]} \\
&= d((\varsigma,\vartheta),(\varsigma',\vartheta')).
\end{aligned}$$

The $\frac{1}{2}\cdot$ in Definition 3.2 is essential, since

$$\begin{aligned}
& d_{SemStore}(\vartheta,\vartheta') \\
&= \sup_{x \in SVar} d_{\frac{1}{2}\cdot I\!P}(\vartheta(x),\vartheta'(x)) \\
&= \sup_{x \in SVar} \tfrac{1}{2}\cdot d_{I\!P}(\vartheta(x),\vartheta'(x)) \\
&\geq \tfrac{1}{2}\cdot d_{I\!P}(\vartheta(x),\vartheta'(x)).
\end{aligned}$$

4 Relating operational and denotational semantics

The operational and denotational semantics differ in various aspects:

- the operational semantics is defined in terms of transitions,
- the denotational semantics is compositional,
- the operational semantics uses syntactic stores whereas the denotational semantics employs semantic stores,
- the denotational semantics records communication attempts,
- the operational semantics models deadlock, and
- the operational semantics makes use of a linear space—sets of sequences—whereas the denotational semantics utilizes a branching space—tree like objects.

Relating the operational and denotational semantics we use the following three operators.

- The linearize operator LIN. This operator abstracts from the additional structure of the branching space arriving at a linear space. It removes (unsuccessful) communication attempts and adds deadlock information.
- The semantify operator sem. This operator assigns to each syntactic store (action) a corresponding semantic store (action).
- The semantify operator SEM. This operator is an obvious extension of sem from a syntactic linear space to a semantic linear space.

In the rest of this section we prove

Theorem 4.1 *For all $s \in Stat$, $\sigma \in SynState$, and $\theta \in SynStore$,*

$$SEM\,(\mathcal{O}\,[\![s]\!](\sigma,\theta)) = LIN\,(\mathcal{D}\,[\![s]\!])(sem\,(\sigma,\theta)).$$

To prove the above theorem, the operational semantics $\mathcal{O}$ suggests the use of induction on transitions whereas the denotational semantics $\mathcal{D}$ hints at using structural induction. We introduce an intermediate semantics $\mathcal{I}$ which is defined in terms of transitions and is compositional. The proof of the above theorem is divided into two parts. First, (an extension of) the denotational semantics is shown to be equivalent to the intermediate semantics. This is proved by structural induction. Second, the intermediate semantics is related to (an extension of) the operational semantics by means of the linearize operator and the semantify operators. This relation is proved by a co-inductive argument (see Section 7 of [Rut93]).

Like the denotational semantics the intermediate semantics uses the space $\mathbb{P}$ as codomain. This requires that the transition system employs semantic actions rather than syntactic ones. Clause (9) of Definition 2.6 then obtains the form

$$[x, \varsigma, \vartheta] \rightarrow_2 [\vartheta\,(x), \varsigma, \vartheta].$$

As a consequence, denotations of statements appear in the configurations. Besides elements of $\mathbb{P}$ we also encounter mixed terms like, e.g., $p\,;s$. We extend the set $Stat$ of statements and the set $\mathbb{P} \setminus \{\mathrm{E}\}$ (forgetting about the metric for a moment) with a restricted set of mixed terms in

Definition 4.2 The set $(r \in)\ Stat^*$ is defined by

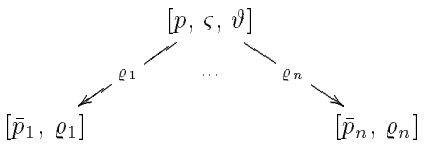
$$r ::= s \mid p \mid r\,;s \mid r \parallel r.$$

Note that we do not consider E nor mixed terms built from E in the set $Stat^*$ of extended statements. We comment on this choice later. We add E in

Definition 4.3 The set $(\bar{r} \in)\ Stat^*_{\mathrm{E}}$ is defined by

$$\bar{r} ::= \mathrm{E} \mid r.$$

The configurations of the transition system consist of an extended statement and a semantic action. We introduce an axiom such that the configuration $[p, \varsigma, \vartheta]$, with $p\,(\varsigma, \vartheta) = \{\langle \varrho_1, \bar{p}_1\rangle, \ldots, \langle \varrho_n, \bar{p}_n\rangle\}$, can make the following transitions:



(cf. the picture following Definition 3.5). The other axioms and rules are straightforward modifications of the axioms and rules of Definition 2.6.

Definition 4.4 The transition relation $\rightarrow_2$ is defined as the smallest subset of

$$(Stat^*_{\mathrm{E}} \times SemAct) \times (Stat^*_{\mathrm{E}} \times SemAct)$$

satisfying the following axioms and rules.

(1) $[v := e\,,\ \varsigma,\ \vartheta] \rightarrow_2 [\mathrm{E},\ \varsigma\{\alpha/v\},\ \vartheta]$, where $\alpha = \mathcal{E}[\![e]\!](\varsigma)$

(2) $$\frac{[s_1,\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{s}_1,\ \varrho]}{[\textbf{\textit{if}}\ b\ \textbf{\textit{then}}\ s_1\ \textbf{\textit{else}}\ s_2\ \textbf{\textit{fi}},\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{s}_1,\ \varrho]}\ ,\ \text{if } \mathcal{B}[\![b]\!](\varsigma) = \mathit{true}$$

(3) $$\frac{[s_2,\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{s}_2,\ \varrho]}{[\textbf{\textit{if}}\ b\ \textbf{\textit{then}}\ s_1\ \textbf{\textit{else}}\ s_2\ \textbf{\textit{fi}},\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{s}_2,\ \varrho]}\ ,\ \text{if } \mathcal{B}[\![b]\!](\varsigma) = \mathit{false}$$

(4) $[c\,!\,s,\ \varsigma,\ \vartheta] \rightarrow_2 [\mathrm{E},\ c\,!\,\mathcal{D}[\![s]\!]]$

(5) $[c\,?\,x,\ \varsigma,\ \vartheta] \rightarrow_2 [\mathrm{E},\ c\,?\,x]$

(6) $[x,\ \varsigma,\ \vartheta] \rightarrow_2 [\vartheta\,(x),\ \varsigma,\ \vartheta]$

(7) $[p,\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{p},\ \varrho]$, if $\langle \varrho, \bar{p}\rangle \in p\,(\varsigma,\ \vartheta)$

(8) $$\frac{[r,\ \varrho] \rightarrow_2 [\bar{r},\ \varrho']}{[r\ ;s,\ \varrho] \rightarrow_2 [\bar{r}\ ;_{\mathrm{E}}\ s,\ \varrho']}$$

(9) $$\frac{[r_1,\ \varrho] \rightarrow_2 [\bar{r}_1,\ \varrho']}{\begin{array}{l}[r_1 \parallel r_2,\ \varrho] \rightarrow_2 [\bar{r}_1 \parallel_{\mathrm{E}} r_2,\ \varrho'] \\ [r_2 \parallel r_1,\ \varrho] \rightarrow_2 [r_2 \parallel_{\mathrm{E}} \bar{r}_1,\ \varrho']\end{array}}$$

(10) $$\frac{[r_1,\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{r}_1,\ c\,!\,p] \qquad [r_2,\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{r}_2,\ c\,?\,x]}{\begin{array}{l}[r_1 \parallel r_2,\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{r}_1 \parallel_{\mathrm{E}} \bar{r}_2,\ \varsigma,\ \vartheta\{p/x\}] \\ [r_2 \parallel r_1,\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{r}_2 \parallel_{\mathrm{E}} \bar{r}_1,\ \varsigma,\ \vartheta\{p/x\}]\end{array}}$$

Note that if we would also have considered E in Definition 4.2 then we should have changed, e.g., rule (8) in order to deal with configurations like $[\mathrm{E}\ ;s,\ \varsigma,\ \vartheta]$.

From the above introduced transition system we derive an intermediate semantics

$$\mathcal{I} : Stat^*_{\mathrm{E}} \rightarrow I\!P$$

satisfying

$$\mathcal{I}\,(\mathrm{E}) = \mathrm{E}$$
$$\mathcal{I}\,(r) = \lambda(\varsigma, \vartheta).\{\ \langle \varrho, \mathcal{I}\,(\bar{r})\rangle \mid [r,\ \varsigma,\ \vartheta] \rightarrow_2 [\bar{r},\ \varrho]\ \}$$

To conclude that, for all $\bar{r} \in Stat^*_{\mathrm{E}}$, $\mathcal{I}\,(\bar{r}) \in I\!P$, we have to check that, for all $r \in Stat^*$, $\varsigma \in SynState$, and $\vartheta \in SynStore$, the set $\mathcal{I}\,(r)(\varsigma, \vartheta)$ is compact. Usually, the compactness is derived from the fact that the transition system is finitely branching. However, the transition system at hand is not finitely branching.

Example 4.5 Consider

$$p = \lambda(\varsigma,\vartheta).\{\,\langle(\varsigma,\vartheta),\bar{p}_n\rangle \mid n \in \mathbb{N}\cup\{\omega\}\,\},$$

where

$$\bar{p}_n = \begin{cases} \mathrm{E} & \text{if } n = 1 \\ \lambda(\varsigma,\vartheta).\{\langle(\varsigma,\vartheta),\bar{p}_{n-1}\rangle\} & \text{if } n > 1 \end{cases}$$

and $\bar{p}_\omega$ is the unique element of $I\!P$ satisfying

$$\bar{p}_\omega = \lambda(\varsigma,\vartheta).\{\langle(\varsigma,\vartheta),\bar{p}_\omega\rangle\}.$$

We have, for all $n \in \mathbb{N}\cup\{\omega\}$,

$$[p,\,\varsigma,\,\vartheta] \rightarrow_2 [\bar{p}_n,\,\varsigma,\,\vartheta].$$

Consequently, the set $\mathcal{FB}([p,\,\varsigma,\,\vartheta])$ is infinite.

By endowing the configurations of the transition system with a suitable metric, we are able to show that the obtained *metric transition system* is *compactly branching*: every configuration has a compact set of outgoing transitions and transitioning is nonexpansive, i.e., for all $\bar{r}_1, \bar{r}_2, \bar{r}'_1 \in \mathit{Stat}^*_{\mathrm{E}}$, and $\varrho_1, \varrho_2, \varrho'_1 \in \mathit{SemAct}$, if

$$[\bar{r}_1,\,\varrho_1] \rightarrow_2 [\bar{r}'_1,\,\varrho'_1]$$

then there exist $\bar{r}'_2 \in \mathit{Stat}^*_{\mathrm{E}}$ and $\varrho'_2 \in \mathit{SemAct}$ such that

$$[\bar{r}_2,\,\varrho_2] \rightarrow_2 [\bar{r}'_2,\,\varrho'_2]$$

and

$$d([\bar{r}'_1,\,\varrho'_1],[\bar{r}'_2,\,\varrho'_2]) \leq d([\bar{r}_1,\,\varrho_1],[\bar{r}_2,\,\varrho_2]).$$

This will turn out to be sufficient to prove that the intermediate semantics assigns to each extended statement an element of $I\!P$. The set of extended statements is turned into a space by the metric introduced in

Definition 4.6 The metric $d : \mathit{Stat}^*_{\mathrm{E}} \times \mathit{Stat}^*_{\mathrm{E}} \rightarrow [0,1]$ is defined by

$$d(\bar{r}_1,\bar{r}_2) = 0$$

if $\bar{r}_1 = \bar{r}_2$, otherwise

$$d(\bar{r}_1,\bar{r}_2) = \begin{cases} d_{I\!P}(p_1,p_2) & \text{if } \bar{r}_1 = p_1 \text{ and } \bar{r}_2 = p_2 \\ d(r_1,r_2) & \text{if } \bar{r}_1 = r_1\,;s \text{ and } \bar{r}_2 = r_2\,;s \\ \max\{d(r_1,r_2),d(r'_1,r'_2)\} & \text{if } \bar{r}_1 = r_1 \parallel r'_1 \text{ and } \bar{r}_2 = r_2 \parallel r'_2 \\ 1 & \text{otherwise} \end{cases}$$

The metric on $\mathit{Stat}^*_{\mathrm{E}}$ is designed in such a way that the metric transition system is compactly branching. The other component of the configurations, the semantic actions, are endowed with the metric introduced in Definition 3.4.

Now we are ready to prove that the metric transition system—the transition system the configurations of which are endowed with the just introduced metric—is compactly branching.

Property 4.7 *The function* $\mathcal{CB} : \mathit{Stat}^*_{\mathrm{E}} \times \mathit{SemAct} \rightarrow \mathcal{P}(\mathit{Stat}^*_{\mathrm{E}} \times \mathit{SemAct})$ *defined by*

$$\mathcal{CB}([\bar{r},\,\varrho]) = \{\,[\bar{r}',\,\varrho'] \mid [\bar{r},\,\varrho] \rightarrow_2 [\bar{r}',\,\varrho']\,\}$$

is an element of $\mathit{Stat}^*_{\mathrm{E}} \times \mathit{SemAct} \rightarrow^1 \mathcal{P}_c((\frac{1}{2}\cdot \mathit{Stat}^*_{\mathrm{E}}) \times \mathit{SemAct})$.

Proof First, we prove that, for all $\bar{r} \in \mathit{Stat}^*_{\mathrm{E}}$ and $\varrho \in \mathit{SemAct}$, the set $\mathcal{CB}([\bar{r},\,\varrho])$ is compact by structural induction on $\bar{r}$. We only consider the following two cases.

1. Let $\bar{r} = p$ and $\varrho = (\varsigma, \vartheta)$. Since

$$\mathcal{CB}([p, \varsigma, \vartheta]) = \{ [\bar{p}, \varrho] \mid \langle \varrho, \bar{p} \rangle \in p(\varsigma, \vartheta) \}$$

and the set $p(\varsigma, \vartheta)$ is compact, also the set $\mathcal{CB}([p, \varsigma, \vartheta])$ is compact.

2. Let $\bar{r} = r ; s$. We have that

$$\mathcal{CB}([r ; s, \varrho]) = \{ [\bar{r} ;_{\mathrm{E}} s, \varrho'] \mid [\bar{r}, \varrho'] \in \mathcal{CB}([r, \varrho]) \}.$$

By induction, the set $\mathcal{CB}([r, \varrho])$ is compact. One can easily verify that the extended operator $;_{\mathrm{E}}$ is nonexpansive. Because the nonexpansive image of a compact set is compact (a consequence of Theorem III of [Ale27]), we can conclude that the set $\mathcal{CB}([r ; s, \varrho])$ is compact.

Second, we prove that, for all $\bar{r}_1, \bar{r}_2 \in Stat^*_{\mathrm{E}}$ and $\varrho_1, \varrho_2 \in SemAct$,

$$d(\mathcal{CB}([\bar{r}_1, \varrho_1]), \mathcal{CB}([\bar{r}_2, \varrho_2])) \leq d([\bar{r}_1, \varrho_1], [\bar{r}_2, \varrho_2])$$

by structural induction on $\bar{r}_1$ and $\bar{r}_2$. We only consider those $\bar{r}_1$ and $\bar{r}_2$ satisfying $0 < d(\bar{r}_1, \bar{r}_2) < 1$, since for all other $\bar{r}_1$ and $\bar{r}_2$ the above equation is vacuously true. Only two cases are elaborated on.

1. Let $\bar{r}_1 = p_1$ and $\bar{r}_2 = p_2$, and $\varrho_1 = (\varsigma, \vartheta_1)$ and $\varrho_2 = (\varsigma, \vartheta_2)$.

$$\begin{aligned}
& d(\mathcal{CB}([p_1, \varsigma, \vartheta_1]), \mathcal{CB}([p_2, \varsigma, \vartheta_2])) \\
&= d(\{ [\bar{p}_1, \varrho_1] \mid \langle \varrho_1, \bar{p}_1 \rangle \in p_1(\varsigma, \vartheta_1) \}, \{ [\bar{p}_2, \varrho_2] \mid \langle \varrho_2, \bar{p}_2 \rangle \in p_2(\varsigma, \vartheta_2) \}) \\
&= d(p_1(\varsigma, \vartheta_1), p_2(\varsigma, \vartheta_2)) \\
&\leq \max\{d(p_1(\varsigma, \vartheta_1), p_2(\varsigma, \vartheta_1)), d(p_2(\varsigma, \vartheta_1), p_2(\varsigma, \vartheta_2))\} \quad \text{[ultrametricity]} \\
&\leq \max\{d(p_1, p_2), d((\varsigma, \vartheta_1), (\varsigma, \vartheta_2))\} \quad [p_2 \text{ is nonexpansive}] \\
&= d([p_1, \varsigma, \vartheta_1], [p_2, \varsigma, \vartheta_2]).
\end{aligned}$$

2. Let $\bar{r}_1 = r_1 ; s$ and $\bar{r}_2 = r_2 ; s$.

$$\begin{aligned}
& d(\mathcal{CB}([r_1 ; s, \varrho_1]), \mathcal{CB}([r_2 ; s, \varrho_2])) \\
&= d(\{ [\bar{r}_1 ;_{\mathrm{E}} s, \varrho_1'] \mid [\bar{r}_1, \varrho_1'] \in \mathcal{CB}([r_1, \varrho_1]) \}, \{ [\bar{r}_2 ;_{\mathrm{E}} s, \varrho_2'] \mid [\bar{r}_2, \varrho_2'] \in \mathcal{CB}([r_2, \varrho_2]) \}) \\
&\leq d(\mathcal{CB}([r_1, \varrho_1]), \mathcal{CB}([r_2, \varrho_2])) \quad [;_{\mathrm{E}} \text{ is nonexpansive}] \\
&\leq d([r_1, \varrho_1], [r_2, \varrho_2]) \quad \text{[induction]} \\
&= d([r_1 ; s, \varrho_1], [r_2 ; s, \varrho_2]).
\end{aligned}$$

□

If we would leave out the $\frac{1}{2}\cdot$ then we would obtain a more restrictive condition which the metric transition system does not satisfy (consider, e.g., the extended statements $\bar{p}_1$ and $\bar{p}_2$ of Example 4.5).

The intermediate semantics $\mathcal{I}$ is defined as the unique fixed point of the function Φ introduced in

Property 4.8 The metric transition system $(Stat^*_{\mathrm{E}} \times SemAct, \rightarrow_2)$ induces the function

$$\Phi : (Stat^*_{\mathrm{E}} \rightarrow^1 I\!P) \rightarrow (Stat^*_{\mathrm{E}} \rightarrow^1 I\!P)$$

defined by

$$\begin{aligned}
&\Phi(\phi)(\mathrm{E}) = \mathrm{E} \\
&\Phi(\phi)(r) = \lambda(\varsigma, \vartheta).\{ \langle \varrho, \phi(\bar{r}) \rangle \mid [r, \varsigma, \vartheta] \rightarrow_2 [\bar{r}, \varrho] \}
\end{aligned}$$

Proof We prove that

1. for all $\phi \in Stat^*_{\mathrm{E}} \rightarrow^1 I\!P$, $r \in Stat^*$, $\varsigma \in SemState$, and $\vartheta \in SemStore$, the set $\Phi(\phi)(r)(\varsigma, \vartheta)$ is nonempty and compact,
2. for all $\phi \in Stat^*_{\mathrm{E}} \rightarrow^1 I\!P$ and $r \in Stat^*$, the function $\Phi(\phi)(r)$ is nonexpansive, and
3. for all $\phi \in Stat^*_{\mathrm{E}} \rightarrow^1 I\!P$, the function $\Phi(\phi)$ is nonexpansive.

In this proof we frequently use that

$$\Phi(\phi)(r)(\varsigma,\vartheta) = \{\, \langle \varrho, \phi(\bar{r})\rangle \mid [\bar{r},\, \varrho] \in \mathcal{CB}([r,\, \varsigma,\, \vartheta]) \,\}.$$

We start with 1. Since the set $\mathcal{CB}([r,\, \varsigma,\, \vartheta])$ is nonempty (as can easily be verified), we can conclude that the set $\Phi(\phi)(r)(\varsigma,\vartheta)$ is also nonempty. According to Property 4.7, the set $\mathcal{CB}([r,\, \varsigma,\, \vartheta])$ is compact. Because ϕ is nonexpansive and the nonexpansive image of a compact set is compact, the set $\Phi(\phi)(r)(\varsigma,\vartheta)$ is compact.
We continue with 2. For all $\varsigma_1,\, \varsigma_2 \in \mathit{SemState}$ and $\vartheta_1,\, \vartheta_2 \in \mathit{SemStore}$,

$$\begin{array}{ll} & d(\Phi(\phi)(r)(\varsigma_1,\vartheta_1), \Phi(\phi)(r)(\varsigma_2,\vartheta_2)) \\ = & d(\{\, \langle \varrho_1, \phi(\bar{r}_1)\rangle \mid [\bar{r}_1,\, \varrho_1] \in \mathcal{CB}([r,\, \varsigma_1,\, \vartheta_1]) \,\}, \{\, \langle \varrho_2, \phi(\bar{r}_2)\rangle \mid [\bar{r}_2,\, \varrho_2] \in \mathcal{CB}([r,\, \varsigma_2,\, \vartheta_2]) \,\}) \\ \leq & d(\mathcal{CB}([r,\, \varsigma_1,\, \vartheta_1]), \mathcal{CB}([r,\, \varsigma_2,\, \vartheta_2])) \quad [\phi \text{ is nonexpansive}] \\ \leq & d([r,\, \varsigma_1,\, \vartheta_1], [r,\, \varsigma_2,\, \vartheta_2]) \quad [\text{Property 4.7}] \\ = & d((\varsigma_1,\vartheta_1), (\varsigma_2,\vartheta_2)). \end{array}$$

We conclude with 3. For all $r_1,\, r_2 \in \mathit{Stat}^*$, $\varsigma \in \mathit{SemState}$, and $\vartheta \in \mathit{SemStore}$,

$$\begin{array}{ll} & d(\Phi(\phi)(r_1)(\varsigma,\vartheta), \Phi(\phi)(r_2)(\varsigma,\vartheta)) \\ \leq & d([r_1,\, \varsigma,\, \vartheta], [r_2,\, \varsigma,\, \vartheta]) \quad [\text{as in the proof of 2.}] \\ = & d(r_1, r_2). \end{array}$$

□

In the above property we restricted ourselves to nonexpansive functions from the space $\mathit{Stat}^*_{\mathrm{E}}$ to $\mathbb{P}$. Without this restriction the property is not valid (consider, e.g., p of Example 4.5 and ϕ satisfying, for all $n \in \mathbb{N}$, $\phi(\bar{p}_n) = \bar{p}_n$ and $\phi(\bar{p}_\omega) = \mathrm{E}$).

The function Φ is a mapping from the nonempty complete space $\mathit{Stat}^*_{\mathrm{E}} \to^1 \mathbb{P}$ to itself. The fact that Φ is a contraction can be easily proved. According to Banach's theorem, Φ has a unique fixed point: the intermediate semantics.

Definition 4.9 The *intermediate semantics* $\mathcal{I} : \mathit{Stat}^*_{\mathrm{E}} \to^1 \mathbb{P}$ is defined by

$$\mathcal{I} = \mathit{fix}(\Phi).$$

The intermediate semantics $\mathcal{I}$ is shown to be equivalent to the extended denotational semantics $\mathcal{D}^*$. This denotational semantics is a natural extension of the denotational semantics $\mathcal{D}$.

Definition 4.10 The *extended denotational semantics* $\mathcal{D}^* : \mathit{Stat}^*_{\mathrm{E}} \to \mathbb{P}$ is defined by

$$\begin{array}{ll} \mathcal{D}^*(\mathrm{E}) & = \mathrm{E} \\ \mathcal{D}^*(s) & = \mathcal{D}[\![s]\!] \\ \mathcal{D}^*(p) & = p \\ \mathcal{D}^*(r\,;s) & = \mathcal{D}^*(r)\,;\mathcal{D}^*(s) \\ \mathcal{D}^*(r_1 \parallel r_2) & = \mathcal{D}^*(r_1) \parallel \mathcal{D}^*(r_2) \end{array}$$

The semantic models $\mathcal{I}$ and $\mathcal{D}^*$ are shown to be equivalent by uniqueness of fixed point, viz we show that $\mathcal{D}^*$ is a fixed point of Φ. This proof technique is due to Kok and Rutten [KR90].

Property 4.11 $\Phi(\mathcal{D}^*) = \mathcal{D}^*$.

Proof First, we should check that $\mathcal{D}^*$ is nonexpansive, which can be verified by structural induction. Second, we should show that $\mathcal{D}^*$ is a fixed point of Φ, which can also be proved by structural induction. □

By uniqueness of fixed point we can conclude

Lemma 4.12 $\mathcal{I} = \mathcal{D}^*$.

Proof Immediate consequence of Definition 4.9, Property 4.11, and Banach's theorem. □

Next we will relate the intermediate semantics and (an extension of) the operational semantics. For that purpose we introduce the already mentioned linearize operator and semantify operators.

The linearize operator abstracts from the additional structure of the branching space $\mathbb{P}$ arriving at the linear space

$$SemState \times SemStore \to^1 \mathcal{P}_{nc}((SemState \times SemStore)^\infty_\delta)$$

where the space $(SemState \times SemStore)^\infty_\delta$ is an instance of

Definition 4.13 Let $(x \in) X$ be a nonempty complete space. The space $(w \in) X^\infty_\delta$ is defined by the equation

$$X^\infty_\delta \cong \{\varepsilon\} + \{\delta\} + X \times \tfrac{1}{2} \cdot X^\infty_\delta .$$

The elements of the space X^∞_δ are

- finite sequences over X,
- finite sequences over X followed by δ, and
- infinite sequences over X.

Instead of $(x_1, (x_2, \ldots, (x_n, \varepsilon)\ldots))$, $(x_1, (x_2, \ldots, (x_n, \delta)\ldots))$, and $(x_1, (x_2, \ldots))$ we write $x_1x_2\ldots x_n$, $x_1x_2\ldots x_n\delta$, and $x_1x_2\ldots$, respectively.

If we endow $SynState \times SynStore$ with the discrete metric the above definition gives us the set introduced in Definition 2.8 endowed with a Baire-like [Bai09] metric as presented in, e.g., [Niv79].

The linearize operator LIN

- removes (unsuccessful) communication attempts,
- adds deadlock information,
- removes the changes caused by the environment, and
- collapses the branching structure.

As the semantic sequential composition and parallel composition, LIN is defined as the unique fixed point of a contractive function from a nonempty complete space to itself.

Definition 4.14 The function LIN is the unique function

$$LIN : \mathbb{P} \to^1 (SemState \times SemStore) \to^1 \mathcal{P}_{nc}((SemState \times SemStore)^\infty_\delta)$$

satisfying

$$LIN\ (\mathrm{E})(\varsigma, \vartheta) = \{\varepsilon\}$$

$$LIN\ (p)(\varsigma, \vartheta) = \begin{cases} \{\delta\} & \text{if } p\,(\varsigma, \vartheta) \subseteq SemCom \times \mathbb{P} \\ \{\, (\varsigma', \vartheta')\, w \mid \langle (\varsigma', \vartheta'), \bar{p}\rangle \in p\,(\varsigma, \vartheta) \wedge w \in LIN\ (\bar{p})(\varsigma', \vartheta') \,\} & \text{otherwise} \end{cases}$$

The condition $p\,(\varsigma, \vartheta) \subseteq SemCom \times \mathbb{P}$ is the semantic counterpart of the syntactic deadlocking condition introduced in Definition 2.9. More precisely, from Theorem 4.1 we can derive that, for all $s \in Stat$, $\sigma \in SynState$, and $\theta \in SynStore$,

$[s, \sigma, \theta]$ deadlocks if and only if $\mathcal{D}\,[\![s]\!](sem(\sigma, \theta)) \subseteq SemCom \times \mathbb{P}$.

The semantify operator sem assigns to each syntactic action a corresponding semantic action. This operator is defined in terms of the denotational semantics $\mathcal{D}$. The semantify operator SEM is the obvious extension of sem from the syntactic linear space $\mathcal{P}_{nc}((SynState \times SynStore)^\infty_\delta)$ to the semantic linear space $\mathcal{P}_{nc}((SemState \times SemStore)^\infty_\delta)$.

Definition 4.15 The function $sem : SynAct \to SemAct$ is defined by

$$\begin{aligned} sem\,(\sigma,\theta) &= (\sigma,\, \lambda x.\mathcal{D}\,[\![\theta\,(x)]\!]) \\ sem\,(c\,!\,s) &= c\,!\mathcal{D}\,[\![s]\!] \\ sem\,(c\,?\,x) &= c\,?\,x \end{aligned}$$

The function Sem is the unique function

$$Sem : (SynState \times SynStore)^{\infty}_{\delta} \to^{1} (SemState \times SemStore)^{\infty}_{\delta}$$

satisfying

$$\begin{aligned} Sem\,(\varepsilon) &= \varepsilon \\ Sem\,(\delta) &= \delta \\ Sem\,((\sigma,\theta)\,w) &= sem\,(\sigma,\theta)\,Sem\,(w) \end{aligned}$$

The function $SEM : \mathcal{P}_{nc}\,((SynState \times SynStore)^{\infty}_{\delta}) \to^{1} \mathcal{P}_{nc}\,((SemState \times SemStore)^{\infty}_{\delta})$ is defined by

$$SEM\,(W) = \{\, Sem\,(w) \mid w \in W \,\}.$$

The operational semantics is extended in

Definition 4.16 The *extended operational semantics*

$$\mathcal{O}^{*} : Stat_{\mathrm{E}} \times SynState \times SynStore \to \mathcal{P}\,((SynState \times SynStore)^{\infty}_{\delta})$$

is defined by

$$\begin{aligned} \mathcal{O}^{*}\,([\mathrm{E},\, \sigma,\, \theta]) &= \{\varepsilon\} \\ \mathcal{O}^{*}\,([s,\, \sigma,\, \theta]) &= \mathcal{O}\,[\![s]\!](\sigma,\theta) \end{aligned}$$

The intermediate semantics $\mathcal{I}$ and the extended operational semantics $\mathcal{O}^{*}$ are related by means of the linearize operator LIN and the semantify operators sem and SEM. We show that, for all $\bar{s} \in Stat_{\mathrm{E}}$, $\sigma \in SynState$, and $\theta \in SynStore$,

$$LIN\,(\mathcal{I}\,(\bar{s}))(sem\,(\sigma,\theta)) = SEM\,(\mathcal{O}^{*}\,([\bar{s},\, \sigma,\, \theta])).$$

It will be convenient to write $\mathcal{H}\,([\bar{s},\, \sigma,\, \theta])$ for the left-hand side.

Definition 4.17 The function

$$\mathcal{H} : Stat_{\mathrm{E}} \times SynState \times SynStore \to \mathcal{P}_{nc}\,((SemState \times SemStore)^{\infty}_{\delta})$$

is defined by

$$\mathcal{H}\,([\bar{s},\, \sigma,\, \theta]) = LIN\,(\mathcal{I}\,(\bar{s}))(sem\,(\sigma,\theta)).$$

The equivalence of $\mathcal{H}$ and $SEM \circ \mathcal{O}^{*}$ is proved by uniqueness of fixed point. We show that $\mathcal{H}$ and $SEM \circ \mathcal{O}^{*}$ are both a fixed point of Ψ.

Property 4.18 The transition system $(Stat_{\mathrm{E}} \times SynAct, \to_1)$ induces the function

$$\begin{aligned} \Psi : (&Stat_{\mathrm{E}} \times SynState \times SynStore \to \mathcal{P}_{nc}\,((SemState \times SemStore)^{\infty}_{\delta})) \\ &\to (Stat_{\mathrm{E}} \times SynState \times SynStore \to \mathcal{P}_{nc}\,((SemState \times SemStore)^{\infty}_{\delta})) \end{aligned}$$

defined by

$$\begin{aligned} \Psi\,(\psi)([\mathrm{E},\, \sigma,\, \theta]) &= \{\varepsilon\} \\ \Psi\,(\psi)([s,\, \sigma,\, \theta]) &= \begin{cases} \{\delta\} & \text{if } [s,\, \sigma,\, \theta] \text{ deadlocks} \\ \{\, sem\,(\sigma',\theta')\,w \mid [s,\, \sigma,\, \theta] \to_1 [\bar{s},\, \sigma',\, \theta'] \wedge w \in \psi\,([\bar{s},\, \sigma',\, \theta']) \,\} & \text{otherwise} \end{cases} \end{aligned}$$

Proof Similar to the proof of Property 4.8 using Property 2.7. □

The space $Stat_{\mathrm{E}} \times SynState \times SynStore \to \mathcal{P}_{nc}((SemState \times SemStore)^{\infty}_{\delta})$ is nonempty and complete. We leave it to the reader to verify that Ψ is contractive.

We show that $\mathcal{H}$ is a fixed point of Ψ in

Property 4.19 $\Psi(\mathcal{H}) = \mathcal{H}$.

Proof First, we relate the transition relations $\rightarrow_1$ and $\rightarrow_2$

1. For all $\bar{s}$, $\bar{s}' \in Stat_{\mathrm{E}}$, and ρ, $\rho' \in SynAct$, if

   $$[\bar{s},\, \rho] \rightarrow_1 [\bar{s}',\, \rho']$$

   then there exists a $\bar{r} \in Stat^*_{\mathrm{E}}$ such that

   $$[\bar{s},\, sem(\rho)] \rightarrow_2 [\bar{r},\, sem(\rho')]$$

   and $\mathcal{I}(\bar{s}') = \mathcal{I}(\bar{r})$.
2. For all $\bar{s} \in Stat_{\mathrm{E}}$, $\bar{r} \in Stat^*_{\mathrm{E}}$, $\rho \in SynAct$, and $\varrho' \in SemAct$, if

   $$[\bar{s},\, sem(\rho)] \rightarrow_2 [\bar{r},\, \varrho']$$

   then there exist $\bar{s}' \in Stat_{\mathrm{E}}$ and $\rho' \in SynAct$ such that

   $$[\bar{s},\, \rho] \rightarrow_1 [\bar{s}',\, \rho']$$

   $sem(\rho') = \varrho'$, and $\mathcal{I}(\bar{r}) = \mathcal{I}(\bar{s}')$.

Both 1. and 2. can be proved by structural induction on $\bar{s}$ (cf. Lemma 4.15 of [BB93]).

Second, we show that, for all $\bar{s} \in Stat_{\mathrm{E}}$, $\sigma \in SynState$, and $\theta \in SynStore$,

$$\Psi(\mathcal{H})([\bar{s},\, \sigma,\, \theta]) = \mathcal{H}([\bar{s},\, \sigma,\, \theta]).$$

We only consider the case that $\bar{s} \neq \mathrm{E}$ and $[\bar{s},\, \sigma,\, \theta]$ does not deadlock. The other two cases are much simpler and left to the reader.

$$\begin{aligned}
&\Psi(\mathcal{H})([\bar{s},\, \sigma,\, \theta]) \\
&= \bigcup \{\, sem(\sigma', \theta')\, w \mid [\bar{s},\, \sigma,\, \theta] \rightarrow_1 [\bar{s}',\, \sigma',\, \theta'] \wedge w \in \mathcal{H}([\bar{s}',\, \sigma',\, \theta']) \,\} \\
&= \bigcup \{\, sem(\sigma', \theta')\, w \mid [\bar{s},\, \sigma,\, \theta] \rightarrow_1 [\bar{s}',\, \sigma',\, \theta'] \wedge w \in LIN(\mathcal{I}(\bar{s}'))(sem(\sigma', \theta')) \,\} \\
&= \bigcup \{\, (\varsigma, \vartheta)\, v \mid [\bar{s},\, sem(\sigma, \theta)] \rightarrow_2 [\bar{r},\, \varsigma,\, \vartheta] \wedge v \in LIN(\mathcal{I}(\bar{r}))(\varsigma, \vartheta) \,\} \quad [1.\text{ and } 2.\,] \\
&= \bigcup \{\, (\varsigma, \vartheta)\, v \mid \langle (\varsigma, \vartheta), p \rangle \in \mathcal{I}(\bar{s})(sem(\sigma, \theta)) \wedge v \in LIN(p)(\varsigma, \vartheta) \,\} \\
&= LIN(\mathcal{I}(\bar{s}))(sem(\sigma, \theta)) \\
&= \mathcal{H}([\bar{s},\, \sigma,\, \theta]).
\end{aligned}$$

□

Also $SEM \circ \mathcal{O}^*$ is a fixed point of Ψ.

Property 4.20 $\Psi(SEM \circ \mathcal{O}^*) = SEM \circ \mathcal{O}^*$.

Proof First, we have to check that, for all $\bar{s} \in Stat_{\mathrm{E}}$, $\sigma \in SynState$, and $\theta \in SynStore$, the set $\mathcal{O}^*([\bar{s},\, \sigma,\, \theta])$ is nonempty and compact. This can be proved using Property 2.7 (cf., e.g., the proof of Theorem 4.2.7 of [Bre94]).

Second, we show that, for all $\bar{s} \in Stat_{\mathrm{E}}$, $\sigma \in SynState$, and $\theta \in SynStore$,

$$\Psi(SEM \circ \mathcal{O}^*)([\bar{s},\, \sigma,\, \theta]) = (SEM \circ \mathcal{O}^*)([\bar{s},\, \sigma,\, \theta]).$$

Again we only consider the case that $\bar{s} \neq \mathrm{E}$ and $[\bar{s}, \varsigma, \theta]$ does not deadlock.

$$
\begin{aligned}
&\Psi\,(SEM \circ \mathcal{O}^*)([\bar{s}, \sigma, \theta]) \\
&\quad = \quad \{\, sem\,(\sigma', \theta')\, w \mid [\bar{s}, \sigma, \theta] \rightarrow_1 [\bar{s}', \sigma', \theta'] \wedge w \in (SEM \circ \mathcal{O}^*)\,([\bar{s}', \sigma', \theta'])\,\} \\
&\quad = \quad SEM\,(\{\, (\sigma', \theta')\, v \mid [\bar{s}, \sigma, \theta] \rightarrow_1 [\bar{s}', \sigma', \theta'] \wedge v \in \mathcal{O}^*\,([\bar{s}', \sigma', \theta'])\,\}) \\
&\quad = \quad (SEM \circ \mathcal{O}^*)\,([\bar{s}, \sigma, \theta]).
\end{aligned}
$$

□

By uniqueness of fixed point we can conclude

Lemma 4.21 *$SEM \circ \mathcal{O}^* = \mathcal{H}$.*

Proof Immediate consequence of Property 4.19 and 4.20 and Banach's theorem. □

Combining Lemma 4.12 and 4.21 we arrive at

Proof of Theorem 4.1

$$
\begin{aligned}
&SEM\,(\mathcal{O}\,[\![s]\!](\sigma, \theta)) \\
&\quad = \quad (SEM \circ \mathcal{O}^*)\,([s, \sigma, \theta]) \\
&\quad = \quad \mathcal{H}\,([s, \sigma, \theta])) \quad \text{[Lemma 4.21]} \\
&\quad = \quad LIN\,(\mathcal{I}\,(s))(sem\,(\sigma, \theta)) \\
&\quad = \quad LIN\,(\mathcal{D}^*\,(s))(sem\,(\sigma, \theta)) \quad \text{[Lemma 4.12]} \\
&\quad = \quad LIN\,(\mathcal{D}\,[\![s]\!])(sem\,(\sigma, \theta)).
\end{aligned}
$$

□

Conclusion

In the preceding four sections we introduced a simple imperative language with second order communication, presented an operational and a denotational semantics, and linked the two semantics. Next we discuss some related issues.

Bisimulation, a notion due to Milner and Park [Mil80, Par81, Mil94], plays an important role in theory of concurrency. Various notions of *higher order bisimulation* have been introduced (see, e.g., [AGR92, MS92, Tho90]). By means of the theory developed by Rutten and Turi in [RT92][9] we can define second order bisimulations on the statements in terms of the transition relation $\rightarrow_1$ and on the extended statements in terms of the transition relation $\rightarrow_2$. The latter gives rise to second order bisimulations on the elements of the space $I\!P$. We can show that the space $I\!P$ is strongly extensional: second order bisimilarity coincides with equality.

The denotational semantics $\mathcal{D}$ is not *fully abstract*—the full abstractness problem for programming languages was first raised by Milner [Mil77]—with respect to the operational semantics $\mathcal{O}$. The intermediate semantics $\mathcal{I}$ models second order bisimilarity (in terms of $\rightarrow_2$) in a fully abstract way.

A simplification with respect to the usual languages of this kind is that we assume one global state and store, rather than a distribution of *local* states and stores over the various parallel components. The design of a mechanism for local states and stores can be found in the work on the semantics of Philips' parallel object oriented language POOL [ABKR86, ABKR89, Rut90].

In a setting with local states and stores arbitrary combinations of sequential and parallel compositions might give rise to statements which are of very little significance. These combinations can be ruled out by replacing the language construct parallel composition by *process creation*. It has been shown by Aalbersberg and America [AA88] that the expressive power of parallel composition and process creation are incomparable. For metric semantic models for process creation we refer the reader to America and De Bakker's [AB88].

[9] Recently Lenisa [Len95] has extended this theory along the lines of Pitts' [Pit94] and applied it in our setting.

In a distributed setting it would be meaningful to transmit a *closure*, a pair consisting of a statement and a local store, rather than just a statement as we do here. This seems to be related to the explicit substitution in the $\lambda\rho$- and $\lambda\sigma$-calculus of Curien et al. [Cur88, ACCL91].

In our setting $c\,?\,x$ is not a *binder* (binding x) as it is in, e.g., ECCS [EN86], CHOCS [Tho90], and the π-calculus [MPW92]. Consider the following statement:

$$c\,!\,s_1\,;c\,!\,s_2 \parallel c\,?\,x \parallel c\,?\,x\,;x\,.$$

Which statement is stored for the statement variable x upon execution is dependent on the order the communications take place. This is a consequence of considering one global state and store. If we would consider local states and stores $c\,?\,x$ would become a binder.

In the definition of the space $\mathbb{P}$ and the denotational semantics $\mathcal{D}$ we exploited the fact that we restricted ourselves to *ultrametric spaces*—rather than using metric spaces. It seems that the ultrametricity is essential for giving metric semantics to higher order notions like second order communication.

References

[AA88] IJ.J. Aalbersberg and P. America, 1988. Personal communication.

[AB88] P. America and J.W. de Bakker. Designing Equivalent Models for Process Creation. *Theoretical Computer Science*, 60(2):109–176, September 1988.

[ABKR86] P. America, J.W. de Bakker, J.N. Kok, and J.J.M.M. Rutten. Operational Semantics of a Parallel Object-Oriented Language. In *Proceedings of the 13th Annual ACM Symposium on Principles of Programming Languages*, pages 194–208, St. Petersburg Beach, January 1986. ACM.

[ABKR89] P. America, J.W. de Bakker, J.N. Kok, and J.J.M.M. Rutten. Denotational Semantics of a Parallel Object-Oriented Language. *Information and Computation*, 83(2):152–205, November 1989.

[ACCL91] M. Abadi, L. Cardelli, P.-L. Curien, and J.-J. Lévy. Explicit Substitutions. *Journal of Functional Programming*, 1(4):375–416, 1991.

[AGR92] E. Astesiano, A. Giovini, and G. Reggio. Observational Structures and their Logics. *Theoretical Computer Science*, 96(1):249–283, April 1992.

[Ale27] P. Alexandroff. Über stetige Abbildungen kompakter Räume. *Mathematische Annalen*, 96:555–571, 1927.

[AR89] 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*, 39(3):343–375, December 1989.

[Bai09] R. Baire. Sur la Représentation des Fonctions Discontinues. *Acta Mathematica*, 32(1):97–176, 1909.

[Ban22] S. Banach. Sur les Opérations dans les Ensembles Abstraits et leurs Applications aux Equations Intégrales. *Fundamenta Mathematicae*, 3:133–181, 1922.

[Bar92] H.P. Barendregt. Lambda Calculi with Types. In S. Abramsky, Dov M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, Background: Computational Structures, chapter 2, pages 117–309. Clarendon Press, Oxford, 1992.

[BB93] J.W. de Bakker and F. van Breugel. Topological Models for Higher Order Control Flow. In S. Brookes, M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Proceedings of the 9th International Conference on Mathematical Foundations of Programming Semantics*, volume 802 of *Lecture Notes in Computer Science*, pages 122–142, New Orleans, April 1993. Springer-Verlag.

[Bou89] G. Boudol. Towards a Lambda-Calculus for Concurrent and Communicating Systems. In J. Diaz and F. Orejas, editors, *Proceedings of the International Joint Conference on Theory and Practice of Software Development*, volume 351 of *Lecture Notes in Computer Science*, pages 149–162, Barcelona, March 1989. Springer-Verlag.

[Bre94] F. van Breugel. *Topological Models in Comparative Semantics.* PhD thesis, Vrije Universiteit, Amsterdam, September 1994.

[BV95] J.W. de Bakker and E.P. de Vink. *Control Flow Semantics.* The MIT Press, Cambridge, 1995. To appear.

[Cur88] P.-L. Curien. The $\lambda\rho$-calculus: an abstract framework for environment machines. Report, LIENS, Paris, October 1988.

[EN86] U. Engberg and M. Nielsen. A Calculus of Communicating Systems with Label Passing. Report DAIMI PB-208, Aarhus University, Aarhus, May 1986.

[Eng89] R. Engelking. *General Topology*, volume 6 of *Sigma Series in Pure Mathematics.* Heldermann Verlag, Berlin, revised and completed edition, 1989.

[Hau14] F. Hausdorff. *Grundzüge der Mengenlehre.* Leipzig, 1914.

[Hen94] M. Hennessy. A Fully Abstract Denotational Model for Higher-Order Processes. *Information and Computation*, 112(1):55–95, July 1994.

[JP90] R. Jagadeesan and P. Panangaden. A Domain-Theoretic Model for a Higher-Order Process Calculus. In M.S. Paterson, editor, *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, volume 443 of *Lecture Notes in Computer Science*, pages 181–194, Coventry, July 1990. Springer-Verlag.

[KR90] J.N. Kok and J.J.M.M. Rutten. Contractions in Comparing Concurrency Semantics. *Theoretical Computer Science*, 76(2/3):179–222, November 1990.

[Kur56] K. Kuratowski. Sur une Méthode de Métrisation Complète des Certains Espaces d'Ensembles Compacts. *Fundamenta Mathematicae*, 43(1):114–138, 1956.

[Len95] M. Lenisa. Final Semantics for Higher Order Concurrent Languages. Draft, June 1995.

[Mil77] R. Milner. Fully Abstract Models of Typed λ-Calculi. *Theoretical Computer Science*, 4(1):1–22, February 1977.

[Mil80] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science.* Springer-Verlag, Berlin, 1980.

[Mil91] R. Milner. The Polyadic π-Calculus: a tutorial. Report ECS-LFCS-91-180, University of Edinburgh, Edinburgh, October 1991.

[Mil93] R. Milner. Elements of Interaction. *Communications of the ACM*, 36(1):78–89, January 1993. Turing Award Lecture.

[Mil94] R. Milner. David Michael Ritchie Park (1935–1990) in memoriam. *Theoretical Computer Science*, 133(2):187–200, October 1994.

[MPW92] R. Milner, J. Parrow, and D. Walker. A Calculus of Mobile Processes, I and II. *Information and Computation*, 100(1):1–40 and 41–77, September 1992.

[MS92] R. Milner and D. Sangiorgi. Barbed Bisimulation. In W. Kuich, editor, *Proceedings of the 19th International Colloquium on Automata, Languages and Programming*, volume 623 of *Lecture Notes in Computer Science*, pages 685–695, Vienna, July 1992. Springer-Verlag.

[Niv79] M. Nivat. Infinite Words, Infinite Trees, Infinite Computations. In J.W. de Bakker and J. van Leeuwen, editors, *Foundations of Computer Science III,* part 2: Languages, Logic, Semantics, volume 109 of *Mathematical Centre Tracts*, pages 3–52. Mathematical Centre, Amsterdam, 1979.

[Par81] D. Park. Concurrency and Automata on Infinite Sequences. In P. Deussen, editor, *Proceedings of 5th GI-Conference on Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183, Karlsruhe, March 1981. Springer-Verlag.

[Pit94] A.M. Pitts. A Co-Induction Principle for Recursively Defined Domains. *Theoretical Computer Science*, 124(2):195–219, February 1994.

[Plo81] G.D. Plotkin. A Structural Approach to Operational Semantics. Report DAIMI FN-19, Aarhus University, Aarhus, September 1981.

[RT92] J.J.M.M. Rutten and D. Turi. On the Foundations of Final Semantics: non-standard sets, metric spaces, partial orders. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Proceedings of the REX Workshop on Semantics: Foundations and Applications*, volume 666 of *Lecture Notes in Computer Science*, pages 477–530, Beekbergen, June 1992. Springer-Verlag.

[Rut90] J.J.M.M. Rutten. Semantic Correctness for a Parallel Object-Oriented Language. *SIAM Journal of Computation*, 19(2):341–383, April 1990.

[Rut92] J.J.M.M. Rutten. Processes as Terms: Non-Well-Founded Models for Bisimulation. *Mathematical Structures in Computer Science*, 2(3):257–275, September 1992.

[Rut93] J.J.M.M. Rutten. A Structural Co-Induction Theorem. In S. Brookes, M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Proceedings of the 9th International Conference on Mathematical Foundations of Programming Semantics*, volume 802 of *Lecture Notes in Computer Science*, pages 83–102, New Orleans, April 1993. Springer-Verlag.

[San92] D. Sangiorgi. *Expressing Mobility in Process Algebras: first-order and higher-order paradigms.* PhD thesis, University of Edinburgh, Edinburgh, 1992.

[Tho90] B. Thomsen. *Calculi for Higher Order Communicating Systems.* PhD thesis, Imperial College, London, September 1990.

A Ultrametric spaces

We present some notions from metric topology and Banach's fixed point theorem. For further details on (metric) topology we refer the reader to Engelking's standard work [Eng89].

We start with the definition of a basic notion: a 1-bounded ultrametric space.

Definition A.1 A *(1-bounded ultrametric) space* is a pair (X, d_X) consisting of

- a set X and
- a function $d_X : X \times X \to [0,1]$, called *(ultra-) metric*, satisfying, for all x, y, $z \in X$,
  - $d_X(x,y) = 0$ if and only if $x = y$,
  - $d_X(x,y) = d_X(y,x)$, and
  - $d_X(x,z) \leq \max\{d_X(x,y), d_X(y,z)\}$.

To simplify notations we shall usually write X instead of (X, d_X) and denote the metric of a space X by d_X.

An example of a metric is presented in

Definition A.2 Let X be a set. The *discrete* metric $d_X : X \times X \to [0,1]$ is defined by

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

From spaces one can build new spaces by means of operations like the shrinking operation $\frac{1}{2}\cdot$, the Cartesian product $\times$ and the disjoint union $+$.

Definition A.3 Let X and Y be spaces.

- The metric $(\frac{1}{2} \cdot d)_X : X \times X \to [0,1]$ is defined by

$$(\tfrac{1}{2} \cdot d)_X(x,y) = \tfrac{1}{2} \cdot d_X(x,y).$$

- The metric $d_{X \times Y} : (X \times Y) \times (X \times Y) \to [0,1]$ is defined by

$$d_{X \times Y}((v,w),(x,y)) = \max\{\, d_X(v,x), d_Y(w,y) \,\}.$$

- The metric $d_{X+Y} : (X+Y) \times (X+Y) \to [0,1]$ is defined by

$$d_{X+Y}(v,w) = \begin{cases} d_X(v,w) & \text{if } v \in X \text{ and } w \in X \\ d_Y(v,w) & \text{if } v \in Y \text{ and } w \in Y \\ 1 & \text{otherwise} \end{cases}$$

Below we will encounter some other operations on spaces.

The completeness of a space is essential in Banach's theorem. Before we introduce this notion we first present the definitions of converging and Cauchy sequence.

Definition A.4 Let X be a space. Let $(x_n)_n$ be a sequence in X and x an element of X.

- The sequence $(x_n)_n$ is said to *converge* to the element x if

$$\forall \epsilon > 0 : \exists N \in \mathbb{N} : \forall n \geq N : d_X(x_n, x) \leq \epsilon.$$

- The sequence $(x_n)_n$ is called *Cauchy* if

$$\forall \epsilon > 0 : \exists N \in \mathbb{N} : \forall m,n \geq N : d_X(x_m, x_n) \leq \epsilon.$$

As can be easily seen, every convergent sequence is Cauchy.

Definition A.5 A space is called *complete* if every Cauchy sequences in the space is convergent.

As one can easily verify, the operations $\frac{1}{2}\cdot$, $\times$, and $+$ preserve completeness.

Compactness, a generalization of finiteness, is introduced in

Definition A.6 A subset of a space is called *compact* if every sequences in the set has a converging subsequence.

The set $\mathcal{P}_{nc}(X)$ of nonempty and compact subsets of the space X is turned into a space by endowing it with the Hausdorff metric (see Chapter VIII, § 6 of [Hau14]) introduced in

Definition A.7 Let X be a space. The *Hausdorff* metric $d_{\mathcal{P}_{nc}(X)} : \mathcal{P}_{nc}(X) \times \mathcal{P}_{nc}(X) \to [0,1]$ is defined by

$$d_{\mathcal{P}_{nc}(X)}(A,B) = \max\{\, \sup\{\, \inf\{\, d_X(a,b) \mid b \in B \,\} \mid a \in A \,\}, \\ \sup\{\, \inf\{\, d_X(b,a) \mid a \in A \,\} \mid b \in B \,\} \,\}.$$

The operation $\mathcal{P}_{nc}$ preserves completeness (Lemma 3 of [Kur56]). The space $\mathcal{P}_c(X)$ of compact subsets of the space X is defined by

$$\mathcal{P}_c(X) = \mathcal{P}_{nc}(X) + \{\emptyset\}.$$

The set $X \to Y$ of functions from the space X to the space Y is turned into a space by endowing it with the metric introduced in

Definition A.8 Let X and Y be spaces. The metric $d_{X \to Y} : (X \to Y) \times (X \to Y) \to [0,1]$ is defined by

$$d_{X \to Y}(f,g) = \sup\{d_Y(f(x), g(x)) \mid x \in X\}.$$

Frequently we restrict ourselves to the subspace of nonexpansive functions.

Definition A.9 Let X and Y be spaces. A function $f : X \to Y$ is called *nonexpansive* if, for all x, $y \in X$,

$$d_Y(f(x), f(y)) \leq d_X(x,y).$$

We denote the space of nonexpansive functions from the space X to the space Y by $X \to^1 Y$. The operations $\to$ and $\to^1$ preserve completeness as can easily be verified.

Next we will introduce an equivalence notion of spaces.

Definition A.10 Let X and Y be spaces. A function $f : X \to Y$ is called *isometric* if, for all x, $y \in X$,

$$d_Y(f(x), f(y)) = d_X(x,y).$$

Note that an isometric function is injective.

Definition A.11 The spaces X and Y are called *isometric*, denoted by $X \cong Y$, if there exists an isometric function from X to Y which is surjective.

Besides the completeness of the space, the contractiveness of the function is another essential ingredient of Banach's theorem.

Definition A.12 Let X and Y be spaces. A function $f : X \to Y$ is called *contractive* if there exists an ϵ, with $0 \leq \epsilon < 1$, such that, for all x, $y \in X$,

$$d_Y(f(x), f(y)) \leq \epsilon \cdot d_X(x,y).$$

We conclude with Banach's fixed point theorem.

Theorem A.13 (Banach) *Let X be a nonempty complete space. If the function $f : X \to X$ is contractive then it has a unique fixed point $fix(f)$.*

Proof See Theorem II.6 of [Ban22]. □