REIHE INFORMATIK
7/2000

Underspecification for Process Algebras

M.E. Majster-Cederbaum
Universität Mannheim
Fakultät für Mathematik & Informatik
D-68131 Mannheim

1 Introduction

During the design of a reactive system it may occur that certain information relevant for the present stage of the specification is either not yet available or should be supressed but will be provided at a later stage. This is where the concept of underspecification or partial specification comes in. One way to handle the situation of incomplete information is to admit a specification that leaves room for later refinement steps in which additional knowledge can be incorporated. The loose specification in the area of algebraic specification of abstract data types is an example for such an approach. There are various ways how underspecification or partial specification can be incorporated into a process algebra setting. [LT91] consider e.g. partial specifications, where the specification of some components is left open. The interpretation is here that these parts may show any behaviour. Such a partial specification S defines an infinite set of concrete processes, i.e. all processes being equivalent to some instance of S, and can e.g. be used to reduce the complexity of compositional verification.
We are here interested in a more restricted form of underspecification which arises from a situation where at a certain point in the system design the decision between various known options of system behaviour is to be postponed. Hence the degree of underspecification is less than in the case of the partial specification. One would like to describe the alternatives within one process expression but at the same time it should be clear that only some of the options will be chosen in a later design step. This can be achieved by introducing a special underspecification operator via which the options are combined. The interpretation of such an underspecification term is that it describes a set of alternative processes, among which we have to choose in a later step. Refinement can be modelled by inclusion between sets of alternatives.
In a recent paper [VD98] Veglioni and De Nicola consider a simple process algebra **BP** of finite processes and propose an alternative approach which avoids the introduction of an additional operator by giving a new interpretation to the choice operator $+$. The operator $+$ is interpreted as underspecification whenever it combines processes which have some initial action in common. The meaning of such a term is a set of deterministic trees (possible worlds). Refinement can then again be modelled by inclusion between sets of possible worlds. This view of underspecification can be illustrated by the coffee dispenser example of [VD98]. The coffee dispenser provides a maximum of n units of coffee, where n is determined by the size of the coffee container. If, for some reason, the actual size has not yet been decided a specification might be given by:

$$CofMach = cof + cof.cof + cof.cof.cof + \ldots$$

This system is not meant to be nondeterministic, but allows for different implementations (possible worlds)

$$CofMach_1 = cof;\ \ CofMach_2 = cof.cof;\ \ldots$$

In a subsequent design step one may then take a decision for one of these variants thus performing a refinement step.

This use of the choice operator, though debatable, allows for a simple operational semantics that is derived from the classical transition system semantics and that is equivalent to a very intuitive denotational semantics of underspecified processes in terms of sets of deterministic trees. It also allows for the definition of a possible worlds equivalence that can be related to other equivalences for the language **BP**. It is restricted, however, as far as the modelling of underspecification combining two arbitrary processes is concerned. [VD98] introduce for this purpose a special underspecification operator $\oplus$ that is used together with the $+$ operator understood as above.

In this paper we consider two variants of underspecification:

I) the variant of [VD98] for an extended language expressing underspecification by the $+$ operator for processes with some common initial action, which we adapt to allow for expressing underspecification among arbitrary processes.

II) an approach where $+$ has its classical meaning and underspecification is described via a separate operator.

The language **BP** of [VD98] of finite, sequential processes is not powerful enough to specify complex processes. At least recursion and a parallel construct have to be added for a more realistic setting. In a first part of the paper we discuss how the idea of [VD98] introduced for finite processes can be carried over to a larger class of processes including recursion. To handle recursion we use metric spaces. We give a denotational semantics for underspecified processes in terms of compact sets of trees which induces a denotational possible worlds equivalence and study its properties. We then consider general underspecification and discuss the approaches I) and II). We treat the question of an operational semantics. It turns out that the equivalence of denotational and operational semantics that holds for finite processes is not valid in general. We also sketch possible extensions.

The paper is organized as follows. Section 2 contains the definitions. In section 3 we describe a domain of trees upon which we will build our semantics. Section 4 presents the denotational possible worlds semantics for the language of recursive processes. In section 5 properties of the possible worlds semantics are presented. In section 6 we discuss general underspecification. Section 7 treats operational semantics. Section 8 contains extensions and connections to related work.

2 Definitions and elementary facts about metric spaces

In the next sections we give a summary of the concepts of processes and metric topology.

2.1 Processes

W consider processes that are able to perform *actions* from a given set **Act**. An action represents any activity of a system at a chosen level of abstraction.

A *transition system over* **Act** is a pair $(\mathbf{A}, \rightarrow)$, where $\mathbf{A}$ is the class of processes (or states) and $\rightarrow \subseteq \mathbf{A} \times \mathbf{Act} \times \mathbf{A}$ is the transition relation. We write $p \xrightarrow{a} q$ for $(p, a, q) \in \rightarrow$. On transition systems a variety of semantic equivalences have been investigated as e.g. presented in [vG90].

We first extend the class **BP** of [VD98] of finite processes by recursion to model infinite behaviour. A parallel construct will be introduced in section 6, concatenation and infinite summation are discussed in section 8.
The class **RBP** of processes given by

- $0 \in \mathbf{RBP}$
- $a.P \in \mathbf{RBP}$ (prefix) for all $a \in \mathbf{Act}$, $P \in \mathbf{RBP}$
- $X \in \mathbf{RBP}$ for all $X \in Idf$
- $P + Q \in \mathbf{RBP}$ (sum) for all $P, Q \in \mathbf{RBP}$
- $fix(X = P)$ for all $X \in Idf$, $P \in \mathbf{RBP}$ such that X is guarded in P

Here Idf is a set of identifiers. An occurrence of $X \in Idf$ is *free* in P iff it does not occur within a subterm of the form $fix(X = Q)$. An identifier $X \in Idf$ is *guarded* in P iff each free occurence of X in P is in the scope of a prefix operation. A process is *closed* iff it does not contain any free occurrences of identifiers. For $P, Q \in \mathbf{RBP}$, $X \in Idf$, $P[X/Q]$ denotes the process where each free occurrence of X in P is substituted by Q.
RBP yields a labelled transition system with the transitions $a.P \xrightarrow{a} P$, $P + Q \xrightarrow{a} P'$ if $P \xrightarrow{a} P'$ or $Q \xrightarrow{a} P'$, $fix(X = P) \xrightarrow{a} P'$ if $P[X/fix(X = P)] \xrightarrow{a} P'$. Processes can be drawn as process graphs, i.e. rooted, connected, directed graphs. The nodes, edges and root of a graph G are denoted by $N(G)$, $E(G)$ and $R(G)$. We want to define the subclass of deterministic processes. For this we introduce a set $INIT$ of assignments that associate a set of actions with each identifier

$$INIT = \{\sigma | \sigma : Idf \rightarrow \mathcal{P}(\mathbf{Act})\}.$$

For $X, Y \in Idf, U \in \mathcal{P}(\mathbf{Act})$ we put

$$\sigma[X/U](Y) := \begin{cases} \sigma(Y) & Y \neq X \\ U & Y = X \end{cases}$$

The function $I : \mathbf{RBP} \rightarrow INIT \rightarrow \mathcal{P}(\mathbf{Act})$ giving the set of initial actions for a process P is defined as follows:

$$I(0)(\sigma) = \emptyset \qquad I(X)(\sigma) = \sigma(X)$$
$$I(a.P)(\sigma) = \{a\} \qquad I(P + Q)(\sigma) = I(P)(\sigma) \cup I(Q)(\sigma)$$
$$I(fix(X = P))(\sigma) = lf\rho_{P,X}(\sigma)$$

where $lf\rho_{P,X}(\sigma)$ is the least fixed point of $\rho_{P,X}(\sigma) : \mathcal{P}(\mathbf{Act}) \rightarrow \mathcal{P}(\mathbf{Act})$ given by $\rho_{P,X}(\sigma)(U) = I(P)\sigma[X/U]$.

Let $P \in \mathbf{RBP}$, $\sigma \in INIT$. The relation $\delta \subset \mathbf{RBP} \times INIT$ characterizes the processes that are deterministic under an assignment σ and is given as follows

$(0,\sigma) \in \delta$
$(X,\sigma) \in \delta$
$(a.P,\sigma) \in \delta \quad$ if $(P,\sigma) \in \delta$
$(P_1 + P_2,\sigma) \in \delta \quad$ if $I(P_1)\sigma \cap I(P_2)\sigma = \emptyset$ and $(P_i,\sigma) \in \delta \quad i = 1,2$
$(fix(X = P),\sigma) \in \delta \quad$ if $(P,\sigma) \in \delta$

$(P,\sigma) \in \delta$ means that P is deterministic under σ.

Example 1. $(X + Y,\sigma) \in \delta$ for $\sigma(X) = \{a\}, \sigma(Y) = \{b,c\}$

Let us call a process P *deterministic* if $(P,\sigma) \in \delta$ for all σ.

Example 2. 0, X, $fix(X = a.X + b.Y)$ are deterministic; if P,Q are deterministic, so is $a.P + b.Q$; $fix(X = a.X + Y)$ is not deterministic.

To handle recursion we will use the metric setting, proposed by [Niv79] and first investigated in [dBZ82].

2.2 Metric Spaces

We recall some basic facts from (metric) topology. We presuppose the notions of *metric space*, *compactness*, *completeness* of a metric space and the theorem that each metric space has a unique completion. Given a metric d on M, then d and $\frac{d}{1+d}$ induce the same topology on M, hence in the following we will *restrict* ourselves to metric spaces (M,d_M) with $d_M : M \times M \to [0,1]$. A metric space (M,d_M) is called *discrete* iff for every $x \in M$ there exists $\varepsilon > 0$ such that $d_M(x,y) < \varepsilon$ implies $x = y$. A n-ary function $f : M \times \ldots \times M \to N$ is called *non-distance-increasing* iff

$$d_N(f(x_1,\ldots,x_n), f(y_1,\ldots,y_n)) \leq \max_{i=1\ldots n} d_M(x_i,y_i)$$

A non-distance-increasing function $f : M \times \ldots \times M \to N$, where (N,d_N) is a complete metric space has a unique extension to the completion of $M \times \ldots \times M$.
$f : M \times \ldots \times M \to N$ is called *contractive* iff there exists a constant c, $0 \leq c < 1$ such that $d_N(f(x_1,\ldots,x_n), f(y_1,\ldots,y_n)) \leq c \cdot \max_{i=1\ldots n} d_M(x_i,y_i)$.
The fixed point theorem by Banach/Cacciopoli states that every contractive function $f : M \to M$ on a complete metric space M has a *unique fixed point* in M.
If (M,d_M) is a (complete) metric space then $(\mathcal{P}_{nco}(M), d_H)$ is a (complete) metric space where

$$\mathcal{P}_{nco}(M) = \{U \subseteq M |\ U \neq \emptyset,\ U \text{ compact}\}$$

and

$$d_H(X,Y) = \max\{\sup_{x\in X} \inf_{y\in Y} d(x,y),\ \sup_{y\in Y} \inf_{x\in X} d(x,y)\}$$

for $X, Y \in \mathcal{P}_{nco}(M)$.
If (M, d_M) is a discrete space then $\mathcal{P}_{nco}(M) = \mathcal{P}_{nf}(M)$, where

$$\mathcal{P}_{nf}(M) = \{U \subseteq M |\ U \neq \emptyset,\ U \text{ finite}\}$$

In section 4 we will define a semantics that associates with each process $P \in \mathbf{RBP}$ a *compact* set of trees. In the sequel we will frequently make use of the fact that a subset S of a metric space is *compact* if every sequence in S contains a subsequence that converges in S. In addition, the following theorems turn out to be useful.

Theorem 1. *Let (M, d) be a metric space. If $X \subseteq \mathcal{P}_{nco}(M)$ is compact and $X \neq \emptyset$ then $\bigcup_{A \in X} A \in \mathcal{P}_{nco}(M)$.*

Proof. Let U be an open cover for $\bigcup A$ and $A \in X$. As A is compact there must be a finite subset of U that covers A and yields an open neighbourhood $U(A)$ of A. Hence $\{U(A)\}_{A \in X}$ is an open cover for X, from where we obtain a finite cover of X , as X is compact. From this finite cover we obtain a finite cover for $\bigcup A$.

The next theorem will enable us to lift certain set-valued operations f defined on trees to operations defined on compact sets of trees by pointwise application of f. As the theorem is stated for arbitrary metric spaces it can also serve as a basis for a possible worlds semantics that is based on other models than trees.

Theorem 2. *Let $(M, d_M), (N, d_N)$ be metric spaces. $f : M \times M \to \mathcal{P}_{nco}(N)$ a non-distance-increasing function. We put for $U, V \in \mathcal{P}_{nco}(M)$*

$$\hat{f}(U, V) = \bigcup_{u \in U, v \in V} f(u, v)$$

then

i) $\hat{f}(U, V)$ is a nonempty compact subset of N for all $U, V \in \mathcal{P}_{nco}(M)$.
ii) $d_H(\hat{f}(U, V), \hat{f}(U', V')) \leq \max(d_H(U, U'), d_H(V, V'))$ for all $U, V \in \mathcal{P}_{nco}(M)$, i.e. $\hat{f} : \mathcal{P}_{nco}(M) \times \mathcal{P}_{nco}(M) \to \mathcal{P}_{nco}(N)$ is non-distance-increasing.

Proof. i) We first observe that $S = \{f(u, v) | u \in U,\ v \in V\}$ is a nonempty compact set for $U, V \in \mathcal{P}_{nco}(M)$: let $(f(u_i, v_i))_{i \in I}$ be a sequence in S, hence $((u_i, v_i))_{i \in I}$ is a sequence in $U \times V$, hence there is a subsequence $((u_{i_j}, v_{i_j}))_{j \in J}$ of $((u_i, v_i))_{i \in I}$ that converges to some (u_0, v_0) in $U \times V$. As f is non-distance-increasing $(f(u_{i_j}, v_{i_j}))_{j \in J}$ converges to $f(u_0, v_0) \in S$. Application of theorem 1 yields the result.
ii) Let $U, V \in \mathcal{P}_{nco}(M)$. We first observe that

$$d(\hat{f}(U, V), \hat{f}(U', V'))$$
$$= d(\bigcup_{u \in U, v \in V} f(u, v), \bigcup_{u' \in U', v' \in V'} f(u', v'))$$

$$\leq d(\{f(u,v) : u \in U,\ v \in V\}, \{f(u',v') : u' \in U', v' \in V'\})$$

$$= \max(\sup_{u\in U, v\in V} \inf_{u'\in U', v'\in V'} d(f(u,v), f(u',v')), \sup_{u'\in U', v'\in V'} \inf_{u\in U, v\in V} d(f(u,v), f(u',v')))$$

$$= \max(L_1, L_2).$$

As f is non-distance-increasing, we obtain for all $u \in U, v \in V$

$$\inf_{u'\in U', v'\in V'} d(f(u,v), f(u',v')) \leq \inf_{u'\in U', v'\in V'} \max(d(u,u'), d(v,v'))$$

$$= \max(\inf_{u'\in U'} d(u,u'), \inf_{v'\in V'} d(v,v'))$$

$$\leq \max(\sup_{u\in U} \inf_{u'\in U'} d(u,u'), \sup_{v\in V} \inf_{v'\in V'} d(v,v'))$$

$$\leq \max(d(U,U'), d(V,V')).$$

Hence

$$L_i \leq \max(d(U,U'), d(V,V')) \text{ for } i = 1,2.$$

3 A domain of trees

The semantics of **BP** is given in [VD98] in terms of finite sets of finite deterministic trees with edge labels in **Act**. In order to be able to model the meaning of recursive processes we have to admit infinite sets of possibly infinite trees. We define in the following a suitable metric space $(\mathbf{D}, d)$ of trees and use $\mathcal{P}_{nco}(\mathbf{D})$ as semantic domain for **RBP** (and the extended languages introduced sections 6 and 8), as compactness generalizes finiteness. The choice of $\mathbf{D}$ is justified as follows. Bisimular terms in **BP** obtain the same meaning in [VD98], hence this semantics can be viewed as a mapping from **BP** to $\mathcal{P}_{nf}(\mathbf{T}_{finbran}/\sim)$. Here $\mathbf{T}_{finbran}$ denotes the class of (isomorphism classes of) finitely branching trees with edge labels in **Act** and $\sim$ denotes bisimulation.
The natural metric on $\mathbf{T}_{finbran}$ is given by

$$d_T(t_1, t_2) = \inf\{\frac{1}{2^n} \mid t_1^{(n)} = t_2^{(n)}\}$$

where $t^{(n)}$ denotes the n-cut of t . With this metric $\mathbf{T}_{finbran}$ is a complete metric space. The metric carries over to $\mathbf{T}_{finbran}/\sim$ and yields an incomplete metric space $(\mathbf{T}_{finbran}/\sim, d_T)$. Let (Δ, δ) denote the completion of $(\mathbf{T}_{finbran}/\sim, d_T)$. (Δ, δ) can be given an alternative, more flexible characterization as follows. Let **CMS** be the category where the objects are complete metric spaces and the arrows are non-distance-increasing functions. The functor $\mathcal{F} : \mathbf{CMS} \to \mathbf{CMS}$ is given by

$$\mathcal{F}(M) = \{\emptyset\} \cup \mathcal{P}_{nco}(\mathbf{Act} \times M)$$

and

$$\mathcal{F}(f) = \lambda U.\{(a, f(m)) | (a,m) \in U\}.$$

It is a well known fact that $\mathcal{F}$ has a unique fixed point in **CMS** [MCZ91,dBZ82] that can be obtained as the metric completion $(\mathbf{D}, d)$ of $\bigcup_{i\geq 0} D_i$ where

$$D_0 = \{\emptyset\}, D_{i+1} = \mathcal{F}(D_i) \quad i \geq 0.$$

As D_i is discrete for $i \geq 0$, we have

$$D_{i+1} = \{\emptyset\} \cup \mathcal{P}_{nf}(\mathbf{A} \times D_i) \quad i \geq 0.$$

$\bigcup_{i\geq 0} D_i$ consists of finitely branching trees of finite height.

Theorem 3. $(\mathbf{D}, d)$ *and* (Δ, δ) *are isometric.*

Proof. This proof is due to [Bai94] and consists of the following observations:
1. the mappings $F_n : \mathbf{T}^{(n)}_{finbran}/\sim \quad \to D_n$, defined by

$$F_n([t_\emptyset]_\sim) = \emptyset \text{ and } F_n([t]_\sim) = \{(a, F_{n-1}([t']_\sim)) : t \xrightarrow{a} t'\}$$

for

$$t \in \mathbf{T}^{(n)}_{finbran} = \{t \in \mathbf{T}_{finbran} | 1 \leq height(t) \leq n\}$$

are welldefined, bijective and distance-preserving ,
2. for every $t \in \mathbf{T}_{finbran}$ the sequence $F([t^{(n)}]_\sim)_{n\geq 0}$ is a Cauchy sequence in $\mathbf{D}$,
3. the mapping $F : \mathbf{T}_{finbran}/\sim \quad \to \mathbf{D}$ given by $F([t]_\sim) = \lim_{n\to\infty} F_n([t^{(n)}]_\sim)$ is an embedding,
4. $F(\mathbf{T}_{finbran}/\sim)$ is dense in $\mathbf{D}$.

By standard arguments [dBZ82] one can introduce operators $\circ$, $+$, and $\cdot$ on $\mathbf{D}$ as follows:

$\circ$: corresponds to the empty tree $\emptyset$
$+$: $\bigcup_{i\geq 0} D_i \times \bigcup_{i\geq 0} D_i \to \bigcup_{i\geq 0} D_i$
$t_1 + t_2 := t_1 \cup t_2$
$+$ joins two trees at the root
$\cdot$: $\mathbf{Act} \times \bigcup_{i\geq 0} D_i \to \bigcup_{i\geq 0} D_i$
$a \cdot t := \{(a, t)\}$

As $+$ and $\cdot$ are non-distance-increasing on $\bigcup_{i\geq 0} D_i$ they may hence be uniquely extended to $\mathbf{D}$. The initial actions function I for trees in $\mathbf{D}$ is given by

$$I(t) = \{a : (a, x) \in t \text{ for some } x \in \bigcup_{i\geq 0} D_i\} \quad \text{for } t \in \bigcup_{i\geq 0} D_i.$$

For $t = \lim t_n \in \mathbf{D}$ we choose some $\varepsilon < \frac{1}{2}$ and determine N such that $d(t_n, t) < \frac{1}{2}$ for all $n \geq N$. We put

$$I(t) = \bigcup_{k\geq N} I(t_k)$$

Lemma 1. $I(t)$ *is finite for all* $t \in \mathbf{D}$.

The subset $\mathbf{D}^d \subseteq \mathbf{D}$ of *deterministic trees* is given by

$D_0^d = \{\emptyset\}$
$D_{i+1}^d = \{\emptyset\} \cup$
$\{U \in \mathcal{P}_{nf}(A \times D_i^d) : \forall a \forall b \forall x \forall y \ (a,x) \in U \wedge (b,y) \in U \wedge (a,x) \neq (b,y) \Rightarrow a \neq b\}$ for $i \geq 0$

$\mathbf{D}^d$ is the completion of $\bigcup D_i^d$. For $i \geq 0$ the subset $D_i^{rd} \subseteq D_i$ of *root deterministic trees* is given by $D_i^{rd} = \{t \in D_i : \forall a \forall b \forall x \forall y \quad ((a,x) \in t \wedge (b,y) \in t \wedge (a,x) \neq (b,y) \Rightarrow a \neq b\}$.

4 Denotational infinite possible worlds semantics for RBP

In this section we present the denotational semantics for **RBP** in terms of compact sets of deterministic trees where each tree represents one option of the specification at the present stage. The choice between the options is to be decided in a later design step thus performing a refinement. The interesting part of this semantics is the definition of an operator $\hat{*}$ on compact sets of trees that models the intended interpretation of $+$.

EXAMPLE Let $P_1 = a.b.0 + a.c.0$, $P_2 = a.e.0 + d.0$. The intended meaning of P_1, resp. of P_2 is shown in Figure 1 while the intended meaning of $P_1 + P_2$ is

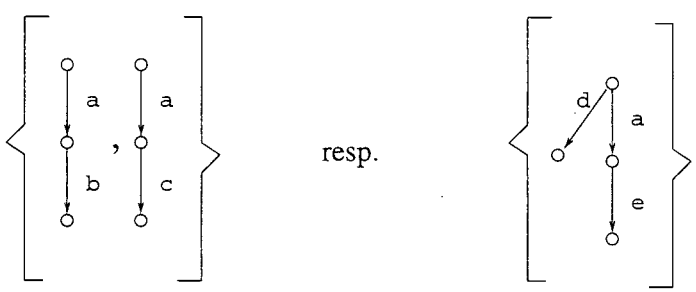


Fig. 1.

shown in Figure 2

For the definition of $\hat{*}$ we proceed as follows. We first define a function *rdet* that decomposes a tree t into a set $rdet(t)$ of root deterministic trees which have the same initial actions as t. We define an operation $*$ on root deterministic trees that already reflects the desired interpretation of $+$. This operation is extended to arbitrary trees by using the function *rdet* and then lifted to compact sets of trees using the theorem 2.

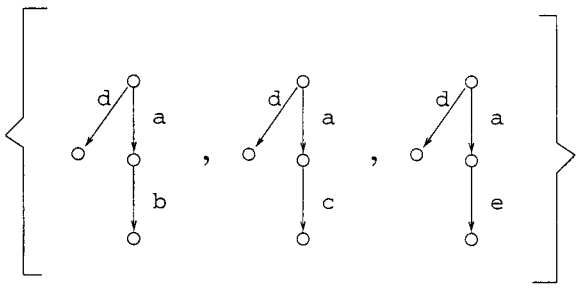


Fig. 2.

Remark 1. It should be noted that the corresponding operator $\star$ of [VD98] cannot be used for trees in $\mathbf{D}$. This is due to the fact that $\mathbf{D}$ contains infinitely branching trees for which $\star$ is not welldefined. However, $\star$ and $*$ coincide for (finite sets of) finite deterministic trees and hence finite processes obtain the same meaning in both semantics.

Definition 1. *Let* $t \in \bigcup_{i\geq 0} D_i$. *We put* $rdet(\emptyset) = \{\emptyset\}$. *For* $t \neq \emptyset$ *let*

$$F_t = \{f |\ f : I(t) \to \bigcup_{i\geq 0} D_i \text{ such that } (a, f(a)) \in t\}$$

and

$$rdet(t) = \{\{(a, f(a))|\ a \in I(t)\} \mid\ \ f \in F_t\}$$

Remark 2. $rdet(t)$ is finite for all $t \in \bigcup_{i\geq 0} D_i$, $t = \bigcup_{t' \in rdet(t)} t'$ for all $t \in \bigcup_{i\geq 0} D_i$ and $rdet(t) = \{t\}$ for each root deterministic $t \in \bigcup_{i\geq 0} D_i$.

$rdet$ decomposes a tree in $\bigcup_{i\geq 0} D_i$ into a set of trees t' in $\bigcup \mathbf{D}_i^{rd}$ with $I(t) = I(t')$.

Lemma 2. $rdet : \bigcup_{i\geq 0} D_i \to \mathcal{P}_{nco}(\mathbf{D}) \cup \{\{\emptyset\}\}$ *is non-distance-increasing.*

Proof. Case 1 $d_H(t_1, t_2) = 1$: is obvious.
Case 2 $d_H(t_1, t_2) = \frac{1}{2^k} < 1$ for some k: we show that

$$d_H(rdet(t_1), rdet(t_2)) \leq \frac{1}{2^k}.$$

Let $\tau \in rdet(t_1)$, $\tau = \{(a_1, \tau_1), \ldots, (a_n, \tau_n)\}$ where $a_i \neq a_j$ for $i \neq j$. As for $i = 1 \ldots n$ $(a_i, \tau_i) \in t_1$ and $\sup_{x\in t_1} \inf_{y\in t_2} d(x, y) \leq \frac{1}{2^k}$ there must be some $y \in t_2$ $y = (a_i, \lambda_i)$ with $d_H((a_i, \tau_i), (a_i, \lambda_i)) \leq \frac{1}{2^k}$.
Put $\rho = \{(a_1, \lambda_1), \ldots, (a_n, \lambda_n)\}$ then $d_H(\tau, \rho) \leq \frac{1}{2^k}$, hence

$$d_H(rdet(t_1), rdet(t_2)) =$$

$$\max(\sup_{x\in rdet(t_1)} \inf_{y\in rdet(t_2)} d(x, y), \sup_{y\in rdet(t_2)} \inf_{x\in rdet(t_1)} d(x, y)) \leq \frac{1}{2^k}.$$

Hence $rdet$ can be canonically extended to a non-distance-increasing map $rdet : \mathbf{D} \to \mathcal{P}_{nco}(\mathbf{D})$.

Example 3. Let $t \in \mathbf{D}$ and t_n as shown below in Figure 3. Then $t = lim\ t_n$ and $rdet(t) = lim(rdet(t_n))$.

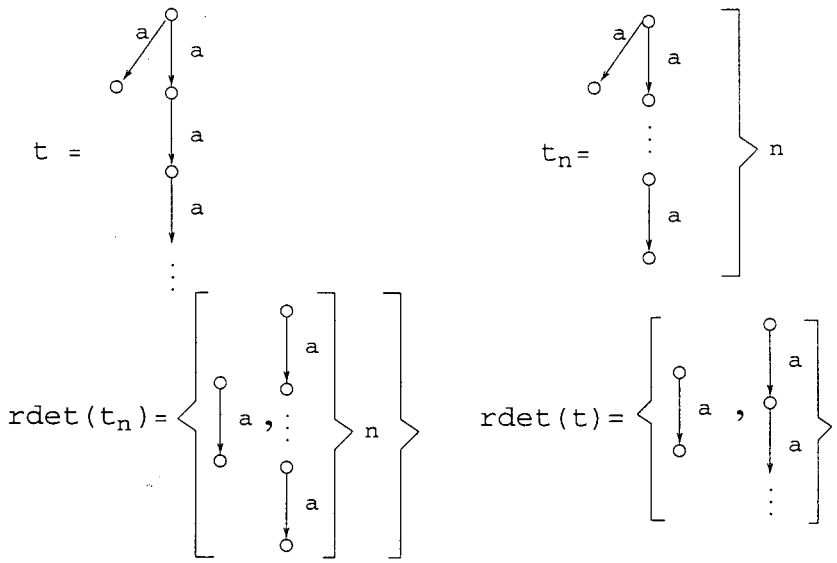


Fig. 3.

Definition 2. *Let $t, t' \in \bigcup_{i\geq 0} D_i$ be root deterministic.Let*

$$F_{t,t'} = \{f |\ f : I(t) \cup I(t') \to \bigcup_{i\geq 0} D_i \ \text{ such that } (a, f(a)) \in t \vee (a, f(a)) \in t'\}.$$

We put

$$t * t' = \{\{(a, f(a))|\ a \in I(t) \cup I(t')\} \mid f \in F_{t,t'}\}$$

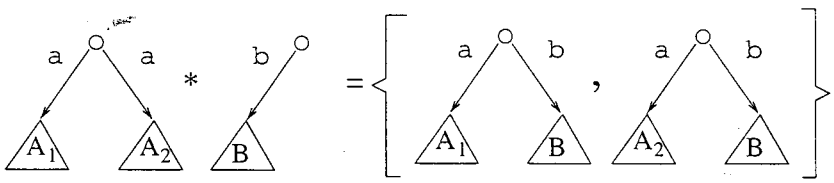
For $t_1, t_2 \in \bigcup_{i\geq 0} D_i$ we put

$$t_1 * t_2 = \bigcup_{\substack{t\in rdet(t_1)\\ t'\in rdet(t_2)}} t * t'$$

Lemma 3. *Let $t_1, t_2 \in \bigcup_{i\geq 0} D_i$ be deterministic trees. Then $t_1 * t_2$ contains only deterministic trees.*

Proof. By induction on the size of $I(t_1) \cap I(t_2)$.

Example 4.



Lemma 4. *Let $t_1, t_2 \in \bigcup_{i\geq 0} D_i$ be root deterministic then*

$$d_H(t_1 * t_2, t_1' * t_2') \leq \max(d_H(t_1, t_1'), d_H(t_2, t_2'))$$

Proof. The case that $d_H(t_1, t_1') = 1$ or $d_H(t_2, t_2') = 1$ is trivial. Let now $d_H(t_1, t_1') < 1$ and $d_H(t_2, t_2') < 1$. Then $I(t_1) = I(t_1')$ and $I(t_2) = I(t_2')$. Let $T = t_1 * t_2$, $T' = t_1' * t_2'$. Each $t \in T$ is a combination of parts of t_1 and parts of t_2, i.e.

$$t \;=\; \{(a_i, \tau_i) |\; i \in I\} \cup \{(b_j, \tau_j) |\; j \in J\},$$

where $(a_i, \tau_i) \in t_1$, $(b_j, \tau_j) \in t_2$.

For each $i \in I$ choose τ_i' such that $(a_i, \tau_i') \in t_1'$ and for each $j \in J$ choose τ_j' such that $(b_j, \tau_j') \in t_2$ and put

$$t' \;=\; \{(a_i, \tau_i') |\; i \in I\} \cup \{(b_j, \tau_j') |\; j \in J\}.$$

Then

$$d_H(t, t') \leq \max(d_H(t_1, t_1'), d_H(t_2, t_2'))$$

hence

$$\sup_{t\in T} \inf_{t'\in T'} d_H(t, t') \leq \max(d_H(t_1, t_1'), d_H(t_2, t_2')).$$

Theorem 4. *Let $t_1, t_2, t_1', t_2' \in \bigcup_{i\geq 0} D_i$, then*

$$d_H(t_1 * t_2, t_1' * t_2') \leq \max(d_H(t_1, t_1'), d_H(t_2, t_2')).$$

Proof.

$$d_H(t_1 * t_2, t_1' * t_2') \;=$$

$$d_H\Big(\bigcup_{\substack{t\in rdet(t_1)\\ \hat{t}\in rdet(t_2)}} t * \hat{t},\; \bigcup_{\substack{t'\in rdet(t_1')\\ \hat{t}'\in rdet(t_2')}} t' * \hat{t}'\Big).$$

By lemma 4 $*$ is a non-distance-increasing function

$$* : \bigcup_{i\geq 0} D_i^{rd} \times \bigcup_{i\geq 0} D_i^{rd} \rightarrow \mathcal{P}_{nco}(\mathbf{D}).$$

Hence by theorem 2

$$\hat{*} : \mathcal{P}_{nco}(\bigcup_{i\geq 0} D_i^{rd}) \times \mathcal{P}_{nco}(\bigcup_{i\geq 0} D_i^{rd}) \rightarrow \mathcal{P}_{nco}(\mathbf{D})$$

is a non-distance-increasing function. As $rdet(t_1), rdet(t_2), rdet(t_1'), rdet(t_2')$ are elements in $\mathcal{P}_{nco}(\bigcup_{i\geq 0} D_i^{rd})$ we get

$$d_H(rdet(t_1) \hat{*} rdet(t_2), rdet(t_1') \hat{*} rdet(t_2'))$$

$$\leq \max(d_H(rdet(t_1), rdet(t_1')), d_H(rdet(t_2), rdet(t_2')))$$

$$\leq \max(d_H(t_1, t_1'), d_H(t_2, t_2'))$$

as $rdet$ is non-distance-increasing.

Hence $*$ can be canonically extended to a non-distance-increasing map

$$* : \mathbf{D} \times \mathbf{D} \rightarrow \mathcal{P}_{nco}(\mathbf{D}).$$

Theorem 5. *Let T_1, T_2 be nonempty compact sets of trees in $\mathbf{D}$ then*

i) $T_1 \hat{*} T_2 := \bigcup_{t_i \in T_i} t_1 * t_2$ is a nonempty compact subset of $\mathbf{D}$
ii) $\hat{*} : \mathcal{P}_{nco}(\mathbf{D}) \times \mathcal{P}_{nco}(\mathbf{D}) \rightarrow \mathcal{P}_{nco}(\mathbf{D})$ a non-distance-increasing function.

Proof. By theorem 2 and theorem 4.

Remark 3. If T_1, T_2 consist of deterministic trees then $T_1 \hat{*} T_2$ consists of deterministic trees.

Definition 3. *Let $\mathbf{ENV} = \{\sigma | \sigma : Idf \rightarrow \mathcal{P}_{nco}(\mathbf{D})\}$ be the set of environments. For $T \in \mathcal{P}_{nco}(\mathbf{D})$, $X, Y \in Idf$*

$$\sigma[X/T](Y) := \begin{cases} \sigma(Y) & Y \neq X \\ T & Y = X \end{cases}$$

The meaning function $\langle\langle\cdot\rangle\rangle : \mathbf{RBP} \rightarrow \mathbf{ENV} \rightarrow \mathcal{P}_{nco}(\mathbf{D})$ is given by

$\langle\langle 0 \rangle\rangle(\sigma) = \{\emptyset\}$
$\langle\langle X \rangle\rangle(\sigma) = \sigma(X)$
$\langle\langle a.P \rangle\rangle(\sigma) = \{\{(a,t)\} | t \in \langle\langle P \rangle\rangle(\sigma)\}$
$\langle\langle P_1 + P_2 \rangle\rangle(\sigma) = \langle\langle P_1 \rangle\rangle(\sigma) \hat{*} \langle\langle P_2 \rangle\rangle(\sigma)$
$\langle\langle fix(X = P) \rangle\rangle(\sigma) = fix\ \Phi_{P,X}(\sigma)$

where $fix\ \Phi_{P,X}(\sigma)$ *is the unique fixed point of the contractive mapping*

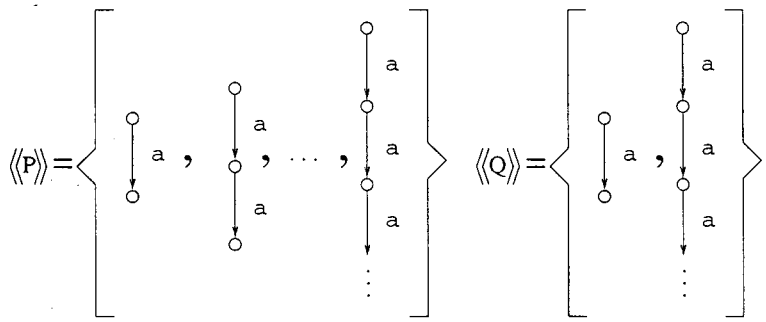
$$\Phi_{P,X}(\sigma) : \mathcal{P}_{nco}(\mathbf{D}) \to \mathcal{P}_{nco}(\mathbf{D})$$

defined by

$$\Phi_{P,X}(\sigma)(T) = \langle\langle P \rangle\rangle \sigma[X/T].$$

Remark 4. For each closed process P, $\langle\langle P \rangle\rangle$ is a set of deterministic trees in $\mathbf{D}$.

Example 5. Let $P = fix(X = a.0 + a.X)$ and $Q = a.0 + fix(X = a.X)$ then $\langle\langle P \rangle\rangle$ consists of infinitely many worlds while $\langle\langle Q \rangle\rangle$ has two possible worlds as shown below.



As in the finite case one obtains a refinement notion as inclusion between sets of the possible worlds.

Definition 4. *Let* $P, Q \in \mathbf{RBP}$ *be closed processes.* Q *is a (denotational) possible worlds refinement of* P*, written* $P \leq_D Q$ *iff* $\langle\langle Q \rangle\rangle \subseteq \langle\langle P \rangle\rangle$*.* P *and* Q *are (denotational) possible worlds equivalent,* $P =_D Q$*, iff* $\langle\langle P \rangle\rangle = \langle\langle Q \rangle\rangle$*.*

Example 6. $P \leq_D Q$ where P, Q are taken from example 5.

5 Properties of the infinite possible worlds refinement

[VD98] gave an axiomatic and operational characterization of the possible worlds refinement and showed that bisimulation implies denotational possible worlds equivalence for finite processes. They informally argue that bisimulation does not imply operational possible worlds equivalence for infinite processes, see also section 7. This is the starting point for this section. It is not difficult to see that the above possible worlds equivalence also satisfies the axioms given in [VD98]. In addition we establish an axiom for recursive processes, which will be of some importance when we look for an operational semantics that is equivalent to the denotational one. We then address the question of the relation between

bisimulation and possible worlds equivalence. We show that bisimulation implies denotational possible worlds equivalence also for infinite processes. This is done by a sequence of steps using three auxiliary functions

$$h_i : \mathbf{RBP} \to Env_i \to H_i,\ i = 1, 2$$

where $H_1 = \mathbf{D}$ and $H_2 = \mathbf{T}_{finbran}$ and

$$det : \mathbf{D} \to \mathcal{P}_{nco}(\mathbf{D})$$

that associates a set of deterministic trees with each $t \in \mathbf{D}$. We show that for each closed process P

$$h_1(P) = F([h_2(P)]_\sim)$$

where $F : \mathbf{T}_{finbran}/_\sim \to \mathbf{D}$, and

$$\langle\langle P\rangle\rangle = det(h_1(P))$$

and

$$(\mathbf{RBP}, \mathbf{Act}, \to, P) \text{ is bisimular to } (\mathbf{T}_{finbran}, \mathbf{Act}, \to, h_2(P))$$

hence $P_1 \sim P_2$ yields $\langle\langle P_1\rangle\rangle = \langle\langle P_2\rangle\rangle$.

The next technical lemma is needed in this section for arguments involving recursive processes.

Lemma 5. *Let $P, P_1, \ldots, P_n \in \mathbf{RBP}$ and $X_1, \ldots, X_n \in Idf$, $X_i \neq X_j$ for $i \neq j$, $\sigma \in \mathbf{ENV}$. Then*

$$\langle\langle P[X_1/P_1, \ldots, X_n/P_n]\rangle\rangle(\sigma)$$
$$= \langle\langle P\rangle\rangle\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)].$$

Proof. By induction on the structure of P. We only show the case

$$P = fix(X = P').$$

We assume w.l.o.g. that X does not appear free in $P_1, \ldots, P_n$.

Case 1 $X \notin \{X_1, \ldots, X_n\}$.Then

$$P[X_1/P_1, \ldots, X_n/P_n] = fix(X = P'[X_1/P_1, \ldots, X_n/P_n]).$$

By induction hypothesis

$$\langle\langle P'[X_1/P_1, \ldots, X_n/P_n]\rangle\rangle(\sigma) = \langle\langle P'\rangle\rangle\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)]$$

hence for all T

$$\Phi_{P'[X_1/P_1,\ldots,X_n/P_n],X}(\sigma)(T) =$$
$$\langle\langle P'[X_1/P_1, \ldots, X_n/P_n]\rangle\rangle\sigma[X/T]$$
$$\langle\langle P'\rangle\rangle\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma), X/T]$$

$$= \Phi_{P',X}(\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)](T)$$

hence

$$\langle\langle P[X_1/P_1, \ldots, X_n/P_n]\rangle\rangle(\sigma) =$$
$$fix\ \Phi_{P'[X_1/P_1,\ldots,X_n/P_n],X}(\sigma) =$$
$$fix\ \Phi_{P',X}\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)]$$

$$= \langle\langle P\rangle\rangle\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)].$$

Case 2 $X = X_i$ for some $i \in \{1 \ldots n\}$. W.l.o.g. we may choose $i = 1$. Then $X = X_1$ does not occur free in P. As X does not occur free in $P_1, \ldots, P_n$, it also does not occur free in $P[X_2/P_2, \ldots, X_n/P_n]$ and hence by *case 1*

$$\langle\langle P[X_2/P_2, \ldots, X_n/P_n]\rangle\rangle(\sigma) = \langle\langle P[X_2/P_2, \ldots, X_n/P_n]\rangle\rangle\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma)]$$

$$= \langle\langle P\rangle\rangle\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), X_2/\langle\langle P_2\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)]$$

$$= \langle\langle P\rangle\rangle\sigma[X_2/\langle\langle P_2\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)].$$

5.1 Axioms

Lemma 6. *The (denotational) possible worlds refinement satisfies the following axioms. Let P, Q, R be closed processes:*

A0: $a.P + a.Q \leq_D a.P$
A1: $P + Q =_D Q + P$
A2: $P + P =_D P$
A3: $(P + Q) + R =_D P + (Q + R)$
A4: $(P + 0) =_D P$
A5: $a.(b.P + b.Q + R) =_D a.(b.P + R) + a.(b.Q + R)$
A6: $fix(X = P') =_D P'[X/fix(X = P')]$

*where $P' \in$ **RBP** and X is the only variable occurring free in P'.*

Proof. The axioms A1 to A5 were already established by [VD98] and carry over to infinite processes. For A6 we show that

$$\langle\langle P'[X/fix(X = P')]\rangle\rangle$$

is a fixed point of

$$\Phi_{P',X}(\sigma)(T) = \langle\langle P'\rangle\rangle\sigma[X/T].$$

$$\langle\langle P'[X/fix(X = P')]\rangle\rangle = \langle\langle P'\rangle\rangle\sigma[X/\langle\langle fix(X = P')\rangle\rangle]$$

$$= \langle\langle P'\rangle\rangle\sigma[X/\langle\langle P'\rangle\rangle\sigma[X/\langle\langle fix(X = P')\rangle\rangle]]$$
$$= \langle\langle P'\rangle\rangle\sigma[X/\langle\langle P'[X/fix(X = P')]\rangle\rangle]$$

by lemma 5.

5.2 Bisimulation and denotational possible worlds equivalence

To establish the relation between the two equivalence notions we use auxiliary functions h_i, $i = 1, 2$, defined in the following.

Definition 5. *Let* $\mathbf{Env}_1 = \{\sigma : Idf \to \mathbf{D}\}$, $h_1 : \mathbf{RBP} \to \mathbf{Env}_1 \to \mathbf{D}$ *is given by*

$h_1(0)(\sigma) = \emptyset$
$h_1(X)(\sigma) = \sigma(X)$
$h_1(a.P)(\sigma) = a \cdot h_1(P)(\sigma)$
$h_1(P_1 + P_2)(\sigma) = h_1(P_1)(\sigma) + h_1(P_2)(\sigma)$
$h_1(fix(X = P))(\sigma) = fix\ \varphi_{P,X}(\sigma)$

where $fix\ \varphi_{P,X}(\sigma)$ *is the unique fixed point of the contractive mapping* $\varphi_{P,X}(\sigma) : \mathbf{D} \to \mathbf{D}$ *where* $\varphi_{P,X}(\sigma)(t) = h_1(P)\sigma[X/t]$

Lemma 7. *Let* P *be a deterministic process,* $\sigma \in \mathbf{Env}_1$

$$\langle\langle P\rangle\rangle(\tilde{\sigma}) = \{h_1(P)(\sigma)\}$$

where $\tilde{\sigma}(X) = \{\sigma(X)\}$ *for all* $X \in Idf$. *In particular* $\langle\langle P\rangle\rangle = \{h_1(P)\}$ *for all closed deterministic* P.

Proof. By induction on the structure of P. The basis of induction and the handling of the operators choice and prefixing are straightforward. We consider the case

$$P = fix(X = P')$$

and show that $\{h_1(P)(\sigma)\}$ is a fixed point of $\Phi_{P',X}(\tilde{\sigma})(T)$. By induction assumption

$$\begin{aligned}\Phi_{P',X}(\tilde{\sigma})(\{h_1(P)(\sigma)\}) &= \langle\langle P'\rangle\rangle\tilde{\sigma}[X/\{h_1(fix(X = P'))(\sigma)\}] \\ &= \{h_1(P')\sigma[X/h_1(fix(X = P'))(\sigma)]\} \\ &= \{h_1(fix(X = P')(\sigma))\} \\ &= \{h_1(P)(\sigma)\}.\end{aligned}$$

Definition 6. *Let* $\mathbf{Env}_2 = \{\sigma : Idf \to \mathbf{T}_{finbran}\}$, $h_2 : \mathbf{RBP} \to \mathbf{Env}_2 \to \mathbf{T}_{finbran}$ *is given by*

$h_2(0)(\sigma) = t_\emptyset$, *(the empty tree)*
$h_2(X)(\sigma) = \sigma(X)$
$h_2(a.P)(\sigma) = a \cdot h_2(P)(\sigma)$, *(prefixing of the tree)*
$h_2(P_1 + P_2)(\sigma) = h_2(P_1)(\sigma) + h_2(P_2)(\sigma)$, *(joining at the root)*
$h_2(fix(X = P))(\sigma) = fix\ f_{P,X}(\sigma)$

where $f_{P,X}(\sigma) : \mathbf{T}_{finbran} \to \mathbf{T}_{finbran}$, $f_{P,X}(\sigma)(t) = h_2(P)\sigma[X/t]$ *is a contractive mapping.*

Remark 5. Let $P, P_1, \ldots, P_n \in \mathbf{RBP}$ and $X_1, \ldots, X_n \in Idf, X_i \neq X_j$, for $i \neq j$.Then

$$h_2(P[X_1/h_2(P_1), \ldots, X_n/h_2(P_n)])(\sigma)$$
$$= h_2(P)\sigma[X_1/h_2(P_1)(\sigma), \ldots, X_n/h_2(P_n)(\sigma)]$$

holds in analogy to lemma 5.

Let $F : \mathbf{T}_{finbran}/_\sim \to \mathbf{D}$ be given by $F([t]_\sim) = \lim_{n\to\infty}(F_n[t^{(n)}])$ where F_n is defined by $F_n([t_\emptyset]_\sim) = \emptyset$, $F_n([t]_\sim) = \{(a, F_{n-1}[t']_\sim) : t \xrightarrow{a} t'\}$ for t with $1 \leq heigh(t) \leq n$, see also theorem 3.

Lemma 8. *Let $P \in \mathbf{RBP}$, $X_1, X_2, \ldots, X_n \in Idf$ the identifiers that occur free in P and $P_1, \ldots, P_n$ closed processes $\in \mathbf{RBP}$. Let $\tau \in \mathbf{Env}_2$ with $\tau(X_i) = h_2(P_i)$ and $\sigma \in \mathbf{Env}_1$ with $\sigma(X_i) = F([h_2(P_i)]_\sim)$. Then $h_1(P)\sigma = F([h_2(P)\tau]_\sim)$*

Proof. By induction on the structure of P. The basis of induction is obvious.
Induction step:
1. $P = a.P'$:

$$h_1(P)(\sigma) = \{(a, h_1(P')(\sigma))\} = \{(a, F[h_2(P')(\tau)]_\sim)\}$$
$$= \lim\{(a, F_{n-1}([(h_2(P')(\tau))^{(n-1)}]_\sim))\}$$
$$= \lim\ F_n([(a \cdot h_2(P')(\tau))^{(n)}]_\sim)$$
$$= F([a \cdot h_2(P')(\tau)]_\sim)$$
$$= F([h_2(a.P')(\tau)]_\sim) = F([h_2(P)(\tau)]_\sim)$$

2. $P = P_1 + P_2$: by simple calculation
3. $P = fix(X = P')$: we show that $F([h_2(fix(X = P'))(\tau)]_\sim)$ is a fixed point of $\varphi_{P',X}(\sigma)$.

$$F([h_2(fix(X = P'))(\tau)]_\sim) = F([h_2(P')\tau[X/h_2(P)(\tau)]]_\sim)$$
$$= F([h_2(P')\tau[X/h_2(P[X_1/P_1, \ldots, X_n/P_n])]]_\sim)$$
$$= F([h_2(P')(\tau')]_\sim) = h_1(P')(\sigma')$$
$$= h_1(P')\sigma[X/F([h_2(P[X_1/P_1, \ldots, X_n/P_n])]_\sim)]$$
$$= h_1(P')\sigma[X/F([h_2(P)(\tau)]_\sim)]$$

by induction assumption, where $\sigma'(X) = F([h_2(P[X_1/P_1, \ldots, X_n/P_n])]_\sim)$ and $\sigma'(Y) = \sigma(Y)$ for $Y \neq X$

Lemma 9. *Let $P, P_1, \ldots, P_n \in \mathbf{RBP}$ and $X_1, \ldots, X_n \in Idf$, $X_i \neq X_j$, for $i \neq j$, identifiers in P, $a \in Act$. If $X_1, \ldots, X_n$ are guarded in P then $P[X_1/P_1, \ldots, X_n/P_n] \xrightarrow{a} Q$ implies the existence of $P' \in \mathbf{RBP}$ such that $P \xrightarrow{a} P'$ and $P'[X_1/P_1, \ldots, X_n/P_n] = Q$*

Proof. [BMC94]

Lemma 10. *Let* $P \in \mathbf{RBP}$, $a \in \mathbf{Act}$, $\sigma \in \mathbf{Env}_2$. *If* $P \xrightarrow{a} P'$ *then* $h_2(P)(\sigma) \xrightarrow{a} h_2(P')(\sigma)$.

Proof. By structural induction. We only show the case

$$P = fix(X = Q).$$

Let $P \xrightarrow{a} P'$, i.e. $Q[X/P] \xrightarrow{a} P'$. By lemma 9 there exists $Q' \in \mathbf{RBP}$ such that $Q \xrightarrow{a} Q'$ and $Q'[X/P] = P'$. We apply the induction hypothesis to Q, hence

$$h_2(Q)(\sigma) \xrightarrow{a} h_2(Q')(\sigma) \quad \text{for all } \sigma$$

hence

$$h_2(Q)\sigma[X/h_2(P)(\sigma)] \xrightarrow{a} h_2(Q')\sigma[X/h_2(P)(\sigma)] \quad \text{for all } \sigma$$

hence $h_2(P)(\sigma) \xrightarrow{a} h_2(P')(\sigma)$ by remark 5.

Lemma 11. *Let* $P \in \mathbf{RBP}$ *and* $X_1, \ldots, X_n$ *the identifiers that occur free in* P, $X_i \neq X_j$, *for* $i \neq j$. *Let* $\sigma \in \mathbf{Env}_2$ *such that* $\sigma(X_i) = h_2(P_i)$ *where* $P_i \in \mathbf{RBP}$ *is closed. If* $X_1, \ldots, X_n$ *are guarded in* P *then for all* $t' \in \mathbf{T}_{finbran}$ *if* $h_2(P)\sigma \xrightarrow{a} t'$ *then there exists* $P' \in \mathbf{RBP}$, P' *closed , such that* $P[X_1/P_1, \ldots, X_n/P_n] \xrightarrow{a} P'$ *and* $h_2(P') = t'$

Proof. By structural induction, lemma 9 and remark 5.

Lemma 12. $R = \{(Q, h_2(Q)) |\ \ Q \in \mathbf{RBP}$ *closed* $\}$ *is a bisimulation between* $(\mathbf{RBP}, \mathbf{Act}, \rightarrow, P)$ *and* $(\mathbf{T}_{finbran}, \mathbf{Act}, \rightarrow, h_2(P))$.

Proof. By lemma 10 and lemma 11.

Definition 7. *Let* $t \in \bigcup_{i \geq 0} D_i$, $I(t) = \{a_1, \ldots, a_n\}$.

$$t(a) := \{t' |\ (a, t') \in t\} \quad a \in \mathbf{Act}$$

$det(t) := \{\{(a_1, x_1), \ldots, (a_n, x_n)\}$ *where* $x_i \in det(t'), t' \in t(a_i)$ *for* $i = 1 \ldots n$ $\}$

Lemma 13. $det(t)$ *is non-distance-increasing on* $\bigcup_{i \geq 0} D_i$

Proof. By lemma 2.

Hence *det* can be extended to $det : \mathbf{D} \rightarrow \mathbf{D}$.

Lemma 14. *Let* $P \in \mathbf{RBP}$, $\sigma \in \mathbf{Env}_1$, $\tilde{\bar{\sigma}}(X) = det(\sigma(X)$ *.Then*

$$\langle\langle P \rangle\rangle(\tilde{\bar{\sigma}}) = det(h_1(P)(\sigma)).$$

In particular $\langle\langle P \rangle\rangle = det(h_1(P))$ *for all closed processes* P.

Proof. By induction on the structure of P. We consider the case

$$P = fix(X = P')$$

and show that $det(h_1(P)(\sigma)) = det(h_1(fix(X = P'))(\sigma))$ is a fixed point of $\Phi_{P',X}(\bar{\tilde{\sigma}})$.

$$\begin{aligned} det(h_1(P)(\sigma)) &= det(h_1(P'))\sigma[X/h_1(P)(\sigma)]) \\ &= det(h_1(P')(\sigma')) \\ &= \langle\langle P'\rangle\rangle(\bar{\tilde{\sigma}}) \end{aligned}$$

by induction hypothesis, where

$$\sigma'(Y) = \sigma(Y) \text{ for} Y \neq X,\ \sigma'(X) = \langle P\rangle(\sigma).$$

hence

$$det(h_1(P)(\sigma)) = \langle\langle P'\rangle\rangle\bar{\tilde{\sigma}}[X/det(h_1(P)(\sigma)]$$

hence $det(h_1(P)(\sigma))$ is a fixed point of $\Phi_{P',X}(\bar{\tilde{\sigma}})$ and $\langle\langle P\rangle\rangle(\bar{\tilde{\sigma}}) = det(h_1(P)(\sigma))$

Theorem 6. *Let $P, Q \in$* **RBP** *be closed processes. If $P \sim Q$ then $P =_D Q$*

Proof. By lemma 8, lemma 12 and lemma 14.

6 General underspecification

It is desirable to be able to describe underspecification that allows to combine arbitrary processes to an underspecified term. [VD98] introduce for this purpose an operator $\oplus$ that is used in combination with $+$ interpreted as above. So e.g. $(a.P + a.Q) \oplus c.R$ displays underspecification twice on the top level. Its meaning is a set consisting of three trees, provided P, Q and R are deterministic. It seems clearer to use a single concept for underspecification. In this section we sketch two alternatives how this can be achieved.

One way to incorporate general underspecification is to use the $+$ as underspecification operator as above together with special symbols that allow to treat arbitrary processes within this setting. Let $\delta_i, i \geq 0$, be symbols $\notin$ **Act**. If we want to model that the decision between a process P and a process Q that e.g. have disjoint sets of initial actions is to be postponed at the present design step, one could express this by $\delta_i.P + \delta_i.Q$ for some i. Taking this option has some advantages. One can use our meaning function for assigning a meaning to underspecified terms that reflects the intended meaning. Consider e.g. the term $\delta_i.P + \delta_i.Q$, then

$$\begin{aligned} \langle\langle \delta_i.P + \delta_i.Q\rangle\rangle &= \langle\langle \delta_i.P\rangle\rangle \hat{*} \langle\langle \delta_i.Q\rangle\rangle \\ &= \delta_i.(\langle\langle P\rangle\rangle \cup \langle\langle Q\rangle\rangle), \end{aligned}$$

i.e. we obtain as meaning all options of P plus all options of Q prefixed by δ_i. The deltas may be now discarded if we are not interested in reconstructing the

points at which we expressed the underspecification and should be kept otherwise. Also, in contrast to an additional underspecification operator one may here use classical transition systems for operational semantics. Equivalence notions that take care of the role of the special symbols δ_i have then to be defined appropriately. One problem with this view and also the mixed one of [VD98] arises if we want to introduce a parallel construct into the language. If we maintain that the meaning of a process should be a set of deterministic trees then the meaning of $a.b.0|a.c.0$ has to be a set consisting of two trees. In a refinement step one of them would be discarded which means that certain computation paths would be lost. On the other hand the meaning of $a.b.0|c.d.0$ is a singleton set and there is no further refinement. This problem can be solved by admitting sets of nondeterministic trees as semantics for a language including a parallel construct. It should be noted that $\mathcal{P}_{nco}(\mathbf{D})$ is still a suitable domain for such an interpretation. However, the definition of the operation $*$ has to be modified as in its present definition it resolves any nondeterminism of its arguments.
Another way to incorporate underspecification is to separate the issues of nondeterminism and underspecification completely by introducing a separate underspecification operator and interpreting $+$ in the standard way. In this setting a parallel construct can be easily incorporated.
We introduce a language **UP** that contains an underspecification operator which is semantically interpreted by the union of sets of trees. The operator $+$ has its conventional meaning and the parallel construct is given an interleaving interpretation. The meaning of processes in **UP** will be given by sets of trees that are no longer deterministic. **UP** is given by

- $0 \in \mathbf{UP}$
- $a.P \in \mathbf{UP}$ $a \in \mathbf{Act}, P \in \mathbf{UP}$
- $X \in \mathbf{UP}$ $X \in Idf$
- $P + Q \in \mathbf{UP}$ $P, Q \in \mathbf{UP}$ (choice)
- $P \Box Q \in \mathbf{UP}$ $P, Q \in \mathbf{UP}$ (underspecification)
- $P|Q \in \mathbf{UP}$ $P, Q \in \mathbf{UP}$ (parallel construct)
- $fix(X = P) \in \mathbf{UP}$
  where $X \in Idf, P \in \mathbf{UP}$ such that X is guarded in P.

Example 7. The specification of the coffee dispenser could be rewritten in the above setting by

$$Cofmach = cof \;\Box\; cof.cof \;\Box\; cof.cof.cof \;\Box\; \ldots$$

Example 8. The expression $(a.P + b.Q) + (a.P' + c.R)$ of **BP** is expressed in **UP** by $(a.P \Box a.P') + b.Q + c.R$.

Lemma 15. *Let* $T_1, T_2, T_1', T_2' \in \mathcal{P}_{nco}(\mathbf{D})$, *then*

a)

$$d_H(T_1 \cup T_2, T_1' \cup T_2') \leq max(d_H(T_1, T_1'), d_H(T_2, T_2'))$$

b)

$$d_H(T_1 \hat{+} T_2, T_1' \hat{+} T_2') \leq max(d_H(T_1, T_1'), d_H(T_2, T_2'))$$

where $T_1 \hat{+} T_2 = \{t_1 + t_2 \mid t_i \in T_i\}$

c)

$$d_H(T_1 \mid T_2,\ T_1' \mid T_2') \leq max(d_H(T_1, T_1'), d_H(T_2, T_2'))$$

where $T_1 \mid T_2 = \{t_1 | t_2 \mid t_i \in T_i\}$ *and* $|$ *defined as in [dBZ82]*

Proof. By theorem 2.

Definition 8. *The meaning function* $[[\cdot]] : \mathbf{UP} \to \mathbf{ENV} \to \mathcal{P}_{nco}(\mathbf{D})$ *for underspecified processes* $\in \mathbf{UP}$ *is given by*

$[[0]](\sigma) = \{\emptyset\}$
$[[X]](\sigma) = \sigma(X)$
$[[a.P]](\sigma) = \{\{(a,t)\} \mid t \in [[P]](\sigma)\}$
$[[P + Q]](\sigma) = [[P]](\sigma) \hat{+} [[Q]](\sigma)$
$[[P|Q]](\sigma) = [[P]](\sigma) \mid [[Q]](\sigma)$
$[[P \Box Q]](\sigma) = [[P]](\sigma) \cup [[Q]](\sigma)$
$[[fix(X = P)]](\sigma) = fix\ \Psi_{P,X}(\sigma)$

where $fix\ \Psi_{P,X}(\sigma)$ *is the unique fixed point of the contractive mapping*

$$\Psi_{P,X}(\sigma) : \mathcal{P}_{nco}(\mathbf{D}) \to \mathcal{P}_{nco}(\mathbf{D})$$

given by $\Psi_{P,X}(\sigma)(T) = [[P]]\sigma[X/T]$.

Example 9. $[[a.(b.d + b.0) \Box c.0]]$ consists of one deterministic tree and one nondeterministic tree.

Example 10. $[[a.b.0 | a.c.0]]$ consists of one nondeterministic tree.

A **RBP** term is strictly guarded if in every subterm of the form $fixX = Q$ the variable X only occurs immediately prefixed, i.e. in the form $\alpha.X$. There is a simple translation tr from strictly guarded **RBP** terms to **UP** such that for each closed process $P \in \mathbf{RBP}$

$$\langle\langle P \rangle\rangle = [[tr(P)]].$$

tr is given by $tr(0) = 0, tr(X) = X, tr(a.S) = a.tr(S), (tr(fix(X = Q)) = fix(X = tr(Q))$.

For additive terms, i.e. terms of the form $P + Q$ we proceed as follows: determine all minimal summands of P and Q, i.e. all summands P_i of P and Q_j of Q that are no summands themselves. Substitute each minimal summand of the form $fix(X = R)$ by $R[X/fix(X = R)]$. This gives rise to a modified set of minimal summands each of which is either 0 or X or of the form $a.S$. For each $a \in \mathbf{Act}$ we collect all terms of the form $a.S$ and combine their translations $tr(a.S)$ with the operator $\Box$ resulting in a term t_a. The terms t_a, $t_b, \ldots$ are combined with the $+$ operator yielding a term $Z \in \mathbf{UP}$. We define $tr(P+Q) = Z$.

The semantics $\langle\langle P \rangle\rangle$ and $[[tr(P)]]$ coincide.

Example 11. The expression

$$(a.d.0 + b.0 + c.0) + (fix(X = a.0 + a.X) + c.e.0)$$

of **RBP** is translated into

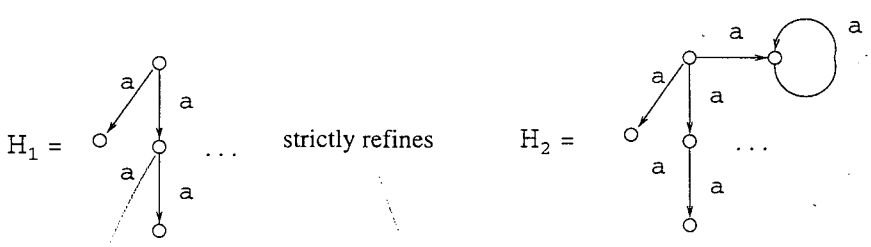
$$(a.d.0) \Box a.0 \Box a.fix(X = a.0 \Box a.X)) + b.0 + (c.0 \Box c.e.0).$$

7 Operational characterization

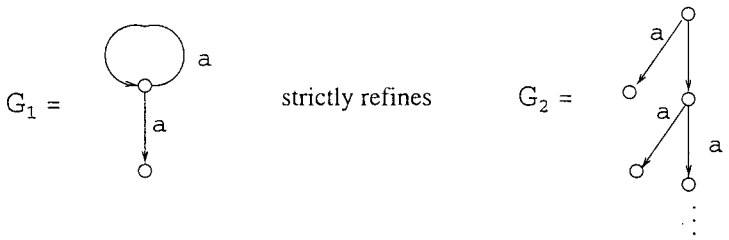
One of the advantages of the use of $+$ in [VD98] is that one may use classical transition systems to obtain an operational meaning for processes in **BP**. [VD98] associate with $P \in \mathbf{BP}$ an operational meaning $PW(P)$ consisting of all process graphs H that are isomorphic to a minimal deterministic graph G satisfying $R(G) = P$ and $(Q \xrightarrow{a} Q',\ Q \in N(G) \Rightarrow \exists Q'' \in N(G) : (Q, a, Q'') \in E(G)$ and $Q \xrightarrow{a} Q'')$.

[VD98] show that two processes $P, Q \in \mathbf{BP}$ are denotational possible worlds equivalent $(P =_D Q)$ iff $PW(Q) = PW(P)$.

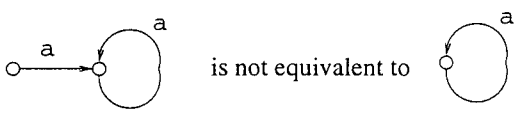
For infinite processes [VD98] remark:*...that infinite processes are already considered in the operational characterization, in fact it is not restricted to finite transitions systems. For example we have that*



We also have that:

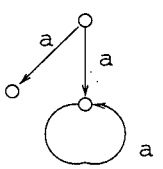


though they are bisimilar. Notice, in fact, that the process on the left admits only two possible worlds... . Unfortunately definition 3 (Definition of PW) *cannot be directly used for infinite processes; it is not sufficiently abstract for loops. For example:*



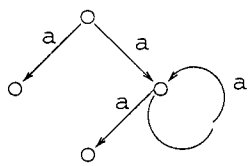
However, this can be easily resolved by choosing a graph equivalence weaker than isomorphism...
Let us assume that we choose such a weak notion of equivalence that identifies the above graphs. Then the resulting operational semantics will be incomparable with the denotational semantics in the sense that
i) there are processes P and Q for which $PW(P) = PW(Q)$ but $P \neq_D Q$
ii) there are processes P and Q for which $P =_D Q$ but $PW(P) \neq PW(Q)$

Example 12. Let $P \equiv fix(X = a.0 + a.X)$ and $Q \equiv a.0 + fix(X = a.X)$. The process graph of P is G_1 whereas the process graph of Q is

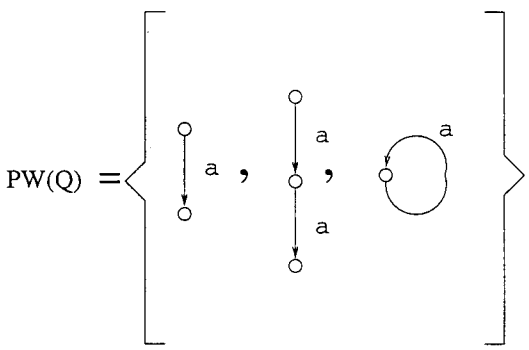


Hence in the view of [VD98] $PW(P)$ and $PW(Q)$ coincide under the assumed weaker notion of graph equivalence, but $Q \neq_D P$, see example 5.

Example 13. Let $P = fix(X = R) = fix(X = a.0 + a.X)$ and $Q = R[X/fix(X = R)] = a.0 + a.fix(X = a.0 + a.X)$, By axiom 6 $P =_D Q$.The transition system for Q is



If G_1 admits only two possible worlds then analogously



hence $PW(P) \neq PW(Q)$ under the assumed notion of graph equivalences.

The question of an appropriate operational semantics for **RBP** and **UP** remains open.

8 Extensions and related work

8.1 Concatenation

It is not difficult to see that concatenation can be easily incorporated into our setting. We omit the 0 process, consider instead each $a \in \mathbf{Act}$ as a basic process and substitute prefixing by concatenation, thus obtaining a language **RCP**. The corresponding semantic operator on $\bigcup_{i\geq 0} \mathbf{D}_i$ is non-distance-increasing and hence all constructions carry over to this case.

8.2 Infinite sums

For simplicity we considered in **RBP** a standard binary 'choice'. It should be noted that the approach can be extended to an infinite summation operator Σ.

The language thus obtained would then include the coffee machine example : $CofMach = cof + cof.cof + \ldots = \Sigma_{i \geq 1} cof^i$. However, we then no longer choose $\mathbf{D}$ as our basic domain, but the complete pseudometric space $\mathbf{T}/_{\sim}$, as $\mathbf{D}$ is too coarse to distinguish between $P_1 = \Sigma_{i \geq 1} a^i$ and $P_2 = \Sigma_{i \geq 1} a^i + fix(X = a.X)$.

8.3 Other models of computation

As already suggested in [VD98] the possible worlds concept can be used to obtain a whole spectrum of possible worlds notion, as e.g. trace possible worlds equivalence and so forth. Another track of transfer of the possible worlds idea leads to other models of computation as e.g. true concurrency models provided we consider a language with concurrency features.

8.4 Related work

The idea of two different types of nondeterminism and their modelling by branching respectively sets is not new. In the special case of a *finite* alphabet **Act** it can be found in [Rou85], where a CSP-type language with the choice-operator of Dijkstra and the $\sqcap$-operator of [BHR84] is considered and interpreted in terms of (sets of) trees.
For a finite alphabet **Act** [Rou85] establishes the relation between the Hennessy-Milner logic HML and $\mathcal{P}_c(\mathbf{T})$ where $\mathcal{P}_c(\mathbf{T})$ denotes the closed subsets of the pseudometric space $\mathbf{T}$.

References

[Bai94] C. Baier. *Transitionssystem- und Baumsemantiken fuer CCS.* PhD thesis, Universitaet Mannheim, 1994.

[BHR84] S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, July 1984.

[BMC94] Christel Baier and Mila E. Majster-Cederbaum. The connection between an event structure semantics and an operational semantics for TCSP. *Acta Informatica*, 31(1):81–104, 1994.

[dBZ82] J. W. de Bakker and J. I. Zucker. Processes and the denotational semantics of concurrency. *Information and Control*, 54(1/2):70–120, July/August 1982.

[LT91] K. G. Larsen and B. Thomsen. Partial specifications and compositional verification. *Theoretical Computer Science*, 88(1):15–32, September 1991.

[MCZ91] Mila E. Majster-Cederbaum and F. Zetzsche. Towards a foundation for semantics in complete metric spaces. *Information and Computation*, 90(2):217–243, February 1991.

[Niv79] M. Nivat. Infinite words, infinite trees, infinite computations. *Math. Cent. Tracts*, 109:1–52, 1979.

[Rou85] William C. Rounds. On the relationship between Scott domains, synchronization trees, and metric spaces. *Information and Control*, 66(1/2):6–28, July/August 1985.

[VD98] S. Veglioni and R. De Nicola. Possible worlds for process algebras. *Lecture Notes in Computer Science*, 1466:179–193, 1998.

[vG90] R. J. van Glabbeek. The linear time – branching time spectrum. In J. C. M. Baeten and J. W. Klop, editors, *Proceedings CONCUR 90,* Amsterdam, volume 458 of *Lecture Notes in Computer Science*, pages 278–297. Springer-Verlag, 1990.