Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Three Metric Domains of Processes for Bisimulation

Franck van Breugel

Computer Science/Department of Software Technology

CS-R9335 1993

Three Metric Domains of Processes for Bisimulation

Franck van Breugel

CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands
Vrije Universiteit
P.O. Box 7161, 1007 MC Amsterdam, The Netherlands

Abstract

A new metric domain of processes is presented. This domain is located in between two metric process domains introduced by De Bakker and Zucker. The new process domain characterizes the collection of image finite processes. This domain has as advantages over the other process domains that no complications arise in the definitions of operators like sequential composition and parallel composition, and that image finite language constructions like random assignment can be modelled in an elementary way. As in the other domains, bisimilarity and equality coincide in this domain.

The three domains are obtained as unique (up to isometry) solutions of equations in a category of 1-bounded complete metric spaces. In the case the action set is finite, the three domains are shown to be equal (up to isometry). For infinite action sets, e.g., equipollent to the set of natural or real numbers, the process domains are proved not to be isometric.

AMS Subject Classification (1991): 68Q55
CR Subject Classification (1991): D.3.1, F.3.2
Keywords & Phrases: process, complete metric space, bisimulation, finitely branching, image finite, sequential composition
Note: This work was partially supported by the Netherlands *Nationale Faciliteit Informatica* programme, project Research and Education in Concurrent Systems (REX). This paper will appear in Proceedings of the Ninth Conference on the Mathematical Foundations of Programming Semantics, New Orleans, LA, USA, April 7-10, 1993.

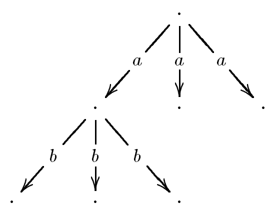
Introduction

In semantics, a *process* is usually understood as a behaviour of a system. *Labelled transition systems* have proved to be suitable for describing the behaviour (or operational semantics) of a system (cf. [Plo81]). A labelled transition system can be viewed as a rooted directed graph of which the edges are labelled by actions (cf. [BK87]), or as a tree of which the edges are labelled by actions, which is obtained by unfolding the graph. The semantic notion of a process is usually defined by means of a suitable behavioural equivalence over the labelled transition systems. *Bisimilarity* (cf. [Par81]) is commonly accepted as the finest behavioural equivalence over labelled transition systems (cf. [Gla90, Gla93]).

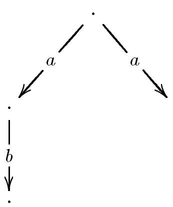
In this paper, processes are studied from the point of view of denotational semantics. In the literature, domains of processes are found for several mathematical structures. For complete partial orders, process domains are presented by Milne and Milner in [MM79], and Abramsky in [Abr91]. Aczel introduces in [Acz88] a process domain for non-well-founded sets. For complete metric spaces, process domains are presented by De Bakker and Zucker in [BZ82, BZ83], and Golson and Rounds in [GR83, Gol84].

Aczel shows in [Acz88] that processes can be viewed as labelled transition systems. Bisimulation relations on these labelled transition systems induce bisimulation relations on the processes. A process domain is called *strongly extensional* (or *internally fully abstract*) if bisimilarity - being the largest bisimulation relation - coincides with equality, i.e. processes are bisimilar if and only if they are equal. Abramsky and Aczel prove that their process domains are strongly extensional. The process domains introduced by De Bakker and Zucker in [BZ82] and [BZ83] are shown to be strongly extensional by Van Glabbeek and Rutten in [GR89] and [Rut92].

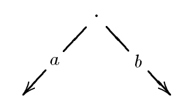
The metric process domains introduced by De Bakker and Zucker in [BZ82] and [BZ83], which will be denoted by P_1 and P_2 in the sequel, and a third new process domain, which will be denoted by P_3, are studied in detail in this paper. Processes can be viewed as trees (both finite and infinite in depth) of which the edges are labelled by actions, and which are absorptive, i.e. for all nodes of a tree the collection of subtrees of that node is a set instead of a multiset, and commutative. For example, the tree



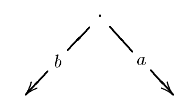
is not a process, and



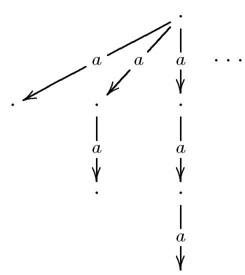
is the process obtained by absorption. Furthermore, the processes



and

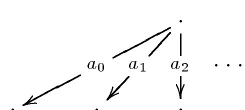


are identified by commutativity. The processes are endowed with a metric such that the distance between processes decreases if the maximal depth at which the truncations of the processes coincide increases. All processes considered in this paper are closed with respect to this metric. For example, the process

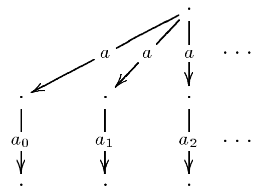


including the infinite branch is closed in contrast with the process not containing this infinite branch.

A process is called *finitely branching* if each node has only finitely many outgoing edges. A process is called *image finite* if, for each action, each node has only finitely many outgoing edges labelled with that action. A finitely branching process is image finite, but an image finite process is in general not finitely branching. For example, the process



is image finite but not finitely branching.



is an example of a general (or unrestricted) process being not finitely branching nor image finite. The process domains P_1, P_2, and P_3 can be shown to correspond to the collections of (finite in depth and)

- general processes,
- finitely branching processes, and
- image finite processes.

For example, the correspondence between the process domain P_3 and the collection of image finite processes of finite depth will be accomplished as follows. First, the space of image finite processes of finite depth is completed. In this way, a complete metric space of (finite and infinite in depth) processes is obtained. Second, the completed space is shown to be isometric to the process domain P_3.

The three process domains can be related in the following way. The process domain P_2 can be isometrically embedded in the process domain P_3 and the process domain P_3 can be isometrically embedded in the process domain P_1. If the action set is finite, then the three process domains can be shown to be isometric. If the action set is infinite, e.g., equipollent ot the set of natural or real numbers, then it can be demonstrated that the three process domains are not isometric.

For P_1-processes, complications arise in the definitions of the following operators:

- sequential composition (cf. [BZ82, BM88]),
- parallel composition (cf. [BZ82, BM88, ABKR89, AR92]),
- trace set as defined by De Bakker et al. in [BBKM84], and
- fairification as defined by Rutten and Zucker in [RZ92].

For example, it is not possible to give a (denotational) definition of the sequential composition of P_1-processes, which coincides with the operational definition of the sequential composition. (Note that processes can be viewed as labelled transition systems.) In [BM88], the sequential composition of P_1-processes is not well-defined. The definition of the sequential composition in [BZ82] is well-defined, but does not coincide to the operational one. It can be shown that these complications do not arise in the definitions of the operators mentioned above on P_2- and P_3-processes.

Unlike the process domain P_2, the process domain P_3 makes an elementary semantic modelling of image finite language constructions like random assignment possible (cf. [Bre94]). (For a detailed overview of metric semantic models the reader is referred to [BR92].)

Novel in the present paper are

- the process domain P_3, which can be shown to correspond to the class of image finite processes and to be strongly extensional,
- the detailed comparison of the process domains P_1, P_2, and P_3, and

- the relation of the process domains P_1, P_2, and P_3 with the classes of general, finitely branching, and image finite processes, extending results concerning the process domains P_1 and P_2 of [BZ82] and [BZ83].

In the first section of this paper, some preliminaries concerning metric spaces can be found. In the second section, the three process domains are introduced. In the third section, the correspondence between P_1-, P_2-, and P_3-processes and general, finitely branching, and image finite processes is studied. The process domains are related as described above in the fourth section. In the fifth section, the process domains are shown to be strongly extensional. In the sixth section, some complications arising in the definition of the sequential composition of P_1-processes are pinpointed. Furthermore, it is shown that these complications do not arise in the definition of this operator on P_3-processes. The other three operators, viz parallel composition, trace set, and fairification, are considered in [Bre94].

In this paper, several definitions from other papers have been modified slightly to stress the correspondence with the other definitions.

1. Metric spaces

Some preliminaries concerning metric spaces are presented. Only some nonstandard notions, i.e. notions which are not found in the main text of [Eng89], are introduced.

Contractive functions, which are called contractions, are introduced in

Definition 1.1 Let (X, d_X) and $(X', d_{X'})$ be metric spaces. A function $f : X \to X'$ is called *contractive* if there exists an ε, with $0 \leq \varepsilon < 1$, such that, for all x and x',

$$d_{X'}(f(x), f(x')) \leq \varepsilon \cdot d_X(x, x').$$

These contractions play a central rôle in

Theorem 1.2 (Banach's theorem) *Let (X, d_X) be a complete metric space. If $f : X \to X$ is a contraction then f has a unique fixed point $\mathit{fix}(f)$. For all x,*

$$\lim_n f^n(x) = \mathit{fix}(f)$$

where

$$f^0(x) = x \text{ and } f^{n+1}(x) = f(f^n(x)).$$

Proof See Theorem II.6 of [Ban22]. □

In this paper, several recursive definitions are presented (cf. Definition 4.1, 4.3, 4.4, 6.1, and 6.3). Banach's theorem can be used to prove the well-definedness of these definitions (cf. [KR90]).

The embeddings to be introduced in Section 4 will be defined by means of nonexpansive functions.

Definition 1.3 Let (X, d_X) and $(X', d_{X'})$ be metric spaces. A function $f : X \to X'$ is called *nonexpansive* if, for all x and x',

$$d_{X'}(f(x), f(x')) \leq d_X(x, x').$$

2. Three process domains

Three process domains are presented. These process domains are defined by means of recursive domain equations.

In [AR89], America and Rutten present a category theoretic technique to solve recursive domain equations. The objects of the category are 1-bounded complete metric spaces. With a domain equation a functor is associated. If this functor satisfies certain conditions, then it has a unique fixed point (up to isometry) which is the intended solution of the domain equation.

The recursive domain equations, by which the process domains are defined, are built from an action set A, which is endowed with the discrete metric, and the constructions described in

Definition 2.1 Let (X, d_X) and $(X', d_{X'})$ be 1-bounded complete metric spaces.
A metric on the Cartesian product of X and X', $X \times X'$, is defined by

$$d_{X \times X'}((x, x'), (\bar{x}, \bar{x}')) = \max\{\, d_X(x, \bar{x}), d_{X'}(x', \bar{x}') \,\}.$$

A metric on the collection of functions from X to X', $X \to X'$, is defined by

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

A new metric on X is defined by

$$d_{id_{\frac{1}{2}}(X)}(x, x') = \tfrac{1}{2} \cdot d_X(x, x').$$

The Hausdorff metric on the set of closed subsets of X, $\mathcal{P}_{cl}(X)$, and on the set of compact subsets of X, $\mathcal{P}_{co}(X)$, is defined by

$$\begin{aligned} d_{\mathcal{P}(X)}(A, B) = \max\{\, & \sup\{\, \inf\{\, d_X(x, x') \mid x' \in B \,\} \mid x \in A \,\}, \\ & \sup\{\, \inf\{\, d_X(x, x') \mid x' \in A \,\} \mid x \in B \,\} \,\} \end{aligned}$$

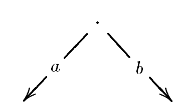
where $\sup \emptyset = 0$ and $\inf \emptyset = 1$.

The three process domains are introduced in

Definition 2.2 The process domains P_1, P_2, and P_3 are defined by the recursive domain equations

$$\begin{aligned} P_1 &\cong \mathcal{P}_{cl}(A \times id_{\frac{1}{2}}(P_1)) \\ P_2 &\cong \mathcal{P}_{co}(A \times id_{\frac{1}{2}}(P_2)) \\ P_3 &\cong A \to \mathcal{P}_{co}(id_{\frac{1}{2}}(P_3)) \end{aligned}$$

Processes as described in the introduction can be represented by elements of these process domains. For example, the process



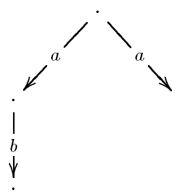
is represented by the P_1- and P_2-process

$$\{(a, \emptyset), (b, \emptyset)\}$$

and by the P_3-process

$$\lambda a' . \begin{cases} \{\lambda a'' . \emptyset\} & \text{if } a' = a \text{ or } a' = b \\ \emptyset & \text{otherwise} \end{cases}$$

The process



is represented by the P_1- and P_2-process

$$\{(a, \{(b, \emptyset)\}), (a, \emptyset)\}$$

and by the P_3-process

$$\lambda a' . \begin{cases} \{p_0, p_1\} & \text{if } a' = a \\ \emptyset & \text{otherwise} \end{cases}$$

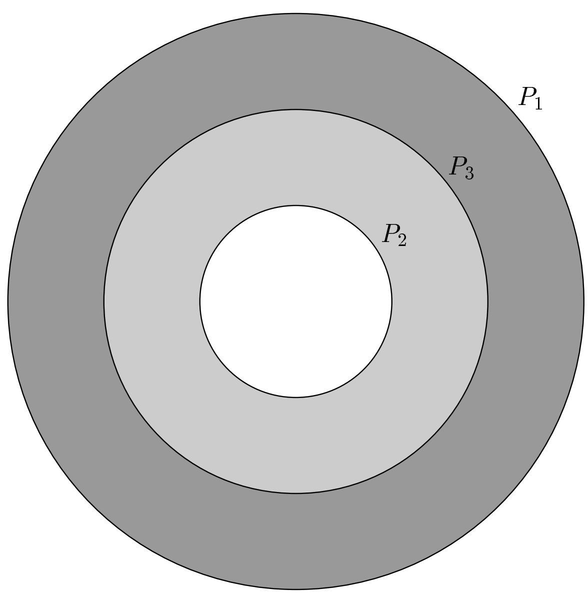
where

$$p_0 = \lambda a'' . \begin{cases} \{\lambda a''' . \emptyset\} & \text{if } a'' = b \\ \emptyset & \text{otherwise} \end{cases}$$

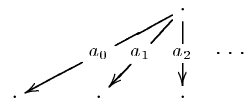
and

$$p_1 = \lambda a'' . \emptyset.$$

Not every process can be represented in all three process domains. In Section 4, we will show that the process domain P_3 is located in between P_1 and P_2, i.e. P_2 can be isometrically embedded in P_3 and P_3 can be isometrically embedded in P_1.



Next, processes in the shaded regions of the above picture are presented. The process



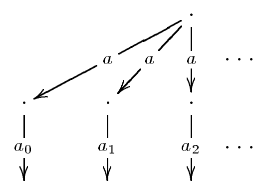
is represented by the P_1-process

$$\{ (a_n, \emptyset) \mid n \in \mathbb{N} \}.$$

However, this is not a P_2-process, because the above set is closed but not compact. The process is also represented by the P_3-process

$$\lambda a' . \begin{cases} \{\lambda a'' . \emptyset\} & \text{if } a' = a_n \text{ for some } n \\ \emptyset & \text{otherwise} \end{cases}$$

The process



is represented by the P_1-process

$$\{ (a, \{(a_n, \emptyset)\}) \mid n \in \mathbb{N} \}.$$

Again, this is not a P_2-process, because the above set is not compact. The process can also not be represented by a P_3-process. The obvious candidate

$$\lambda a' . \begin{cases} \{ p_n \mid n \in \mathbb{N} \} & \text{if } a' = a \\ \emptyset & \text{otherwise} \end{cases}$$

where

$$p_n = \lambda a'' . \begin{cases} \{\lambda a''' . \emptyset\} & \text{if } a'' = a_n \\ \emptyset & \text{otherwise} \end{cases}$$

is not a P_3-process, since the set

$$\{ p_n \mid n \in \mathbb{N} \}$$

is not compact.

3. Finite processes

The three process domains are related to certain collections of *finite* (in depth) processes. It is demonstrated that P_1-, P_2-, and P_3-processes correspond to general, finitely branching, and image finite processes, respectively.

The set of processes of finite depth is introduced in

DEFINITION 3.1 The set P_1^* of *processes of finite depth* is defined by

$$P_1^* = \bigcup \{ P_1^n \mid n \in \mathbb{N} \}$$

where

$$P_1^n = \begin{cases} \{\emptyset\} & \text{if } n = 0 \\ \mathcal{P}(A \times P_1^{n-1}) & \text{otherwise} \end{cases}$$

Obviously, each P_1^*-process is a P_1-process. The P_1^*-processes are endowed with the restriction of the metric on the P_1-processes. The obtained metric space is not complete. For example, the sequence $(p_n)_n$ of P_1^*-processes defined by

$$p_n = \begin{cases} \emptyset & \text{if } n = 0 \\ \{(a, p_{n-1})\} & \text{otherwise} \end{cases}$$

is a Cauchy sequence but does not have a limit in P_1^* (the sequence converges to a process of infinite depth). The metric completion of the metric space of P_1^*-processes, which is denoted by $\widetilde{P_1^*}$, is shown to be isometric to the process domain P_1 in

THEOREM 3.2 $\widetilde{P_1^*} \cong P_1$.

PROOF See Theorem 2.11 of [BZ82]. □

The set of finitely branching processes of finite depth is introduced in the following definition, in which $\mathcal{P}_{fi}$ denotes the set of all finite subsets.

DEFINITION 3.3 The set P_2^* of *finitely branching processes of finite depth* is defined by

$$P_2^* = \bigcup \{ P_2^n \mid n \in \mathbb{N} \}$$

where

$$P_2^n = \begin{cases} \{\emptyset\} & \text{if } n = 0 \\ \mathcal{P}_{fi}(A \times P_2^{n-1}) & \text{otherwise} \end{cases}$$

Similarly, the metric completion of the metric space of P_2^*-processes is proved to be isometric to the complete metric space of P_2-processes in

THEOREM 3.4 $\widetilde{P_2^*} \cong P_2$.

PROOF See Theorem 3.2 of [BZ83]. □

The set of image finite processes of finite depth is introduced in

DEFINITION 3.5 The set P_3^* of *image finite processes of finite depth* is defined by

$$P_3^* = \bigcup \{ P_3^n \mid n \in \mathbb{N} \}$$

where

$$P_3^n = \begin{cases} \{\lambda a \cdot \emptyset\} & \text{if } n = 0 \\ A \to \mathcal{P}_{\mathrm{fi}}\,(P_3^{n-1}) & \text{otherwise} \end{cases}$$

The process domain P_3 can be shown to be isometric to the metric completion of the metric space of P_3^*-processes.

THEOREM 3.6 $\widetilde{P_3^*} \cong P_3$.

PROOF Similar to the proofs of the Theorems 3.2 and 3.4. $\square$

4. Comparison of the process domains

The three process domains are related. It is shown that the process domain P_2 can be isometrically embedded in the process domain P_3 and that the process domain P_3 can be isometrically embedded in the process domain P_1. Furthermore, if the action set A is finite, then the process domain P_1 can be isometrically embedded in the process domain P_2 such that the diagram

$$\begin{tikzcd}
& P_3 \arrow[loop above, "id"] \arrow[dr, "i_2"] & \\
P_2 \arrow[ur, "i_1"] \arrow[loop below, "id"] & & P_1 \arrow[ll, "i_3"] \arrow[loop below, "id"]
\end{tikzcd}$$

commutes. Consequently, if the action set A is finite, then the process domains P_1, P_2, and P_3 are isometric. If the action set A is infinite, then it can be proved that the process domains P_1, P_2, and P_3 are not isometric.

The embedding i_1 from the process domain P_2 to the process domain P_3 is introduced in

DEFINITION 4.1 The embedding $i_1 : P_2 \to P_3$ is defined by

$$i_1\,(p) = \lambda a \cdot \{\, i_1\,(p') \mid (a, p') \in p \,\}.$$

In order to prove the well-definedness of the above recursive definition of the embedding i_1, a so-called higher-order transformation Ψ_{i_1} is introduced in

DEFINITION 4.2 The higher-order transformation $\Psi_{i_1} : (P_2 \to^1 P_3) \to (P_2 \to^1 P_3)$ is defined by

$$\Psi_{i_1}\,(\psi)(p) = \lambda a \cdot \{\, \psi\,(p') \mid (a, p') \in p \,\}.$$

In order to be well-defined, the higher-order transformation Ψ_{i_1} is restricted to nonexpansive functions, i.e.

$$\Psi_{i_1} \in (P_2 \to^1 P_3) \to (P_2 \to^1 P_3).$$

(The collection of nonexpansive functions from P_2 to P_3, $P_2 \to^1 P_3$, endowed with the restriction of the metric on functions from P_2 to P_3 is a complete metric space.) Although only continuity, which is implied by nonexpansiveness, is needed in the well-definedness proof of the higher-order transformation Ψ_{i_1}, the restriction induces half of the proof that the embedding i_1 is isometric (see below). This higher-order transformation Ψ_{i_1} can be shown to be contractive (here the $id_{\frac{1}{2}}$ in the domain equation of process domain P_3 is crucial). According to Banach's theorem (cf. Theorem 1.2), the higher-order transformation Ψ_{i_1} has a unique fixed point which is the intended embedding i_1, i.e.

$$i_1 = \mathit{fix}\,(\Psi_{i_1}).$$

Consequently, $i_1 \in P_2 \to^1 P_3$. To show that the embedding i_1 is isometric it is left to prove that, for all p and p',

$$d\,(i_1\,(p), i_1\,(p')) \geq d\,(p, p').$$

This can be demonstrated by fixed point induction using Banach's theorem.

The embedding i_2 from the process domain P_3 to the process domain P_1 is introduced in

Definition 4.3 The embedding $i_2 : P_3 \to P_1$ is defined by

$$i_2\,(p) = \{\,(a, i_2\,(p')) \mid p' \in p\,(a)\,\}.$$

As the embedding i_1, also the embedding i_2 can be shown to be well-defined and isometric.

Assume the action set A is finite. Then the process domain P_1 can be isometrically embedded in the process domain P_2. The embedding i_3 from the process domain P_1 to the process domain P_2 is introduced in

Definition 4.4 The embedding $i_3 : P_1 \to P_2$ is defined by

$$i_3\,(p) = \{\,(a, i_3\,(p')) \mid (a, p') \in p\,\}.$$

Also this embedding can be shown to be well-defined by means of a higher-order transformation. In the well-definedness proof of the higher-order transformation the compactness of the process domain P_1 is exploited. The process domain P_1 is compact, since the solution of a recursive domain equation built from 1-bounded compact metric spaces (e.g., the finite action set A endowed with the discrete metric), $\mathcal{P}_{cl}$, $\times$, and $id_{\frac{1}{2}}$ is a 1-bounded compact metric space as is proved in [BW93].

The embedding i_3 can also be shown to be isometric. Furthermore, it can be demonstrated that the above diagram commutes. For example, it can be proved that

$$d\,(i_3 \circ i_2 \circ i_1, id) \leq \tfrac{1}{2} \cdot d\,(i_3 \circ i_2 \circ i_1, id)$$

and hence $i_3 \circ i_2 \circ i_1 = id$. As a consequence, the process domains P_1, P_2, and P_3 are isometric.

Theorem 4.5 *If A is finite, then $P_1 \cong P_2$, $P_2 \cong P_3$, and $P_1 \cong P_3$.*

Assume the action set is infinite. More precisely, assume A is equipollent to $2 \uparrow n$, for some n, where $2 \uparrow n$ is defined in

Definition 4.6 The sets $2 \uparrow n$ are defined by

$$2 \uparrow n = \begin{cases} \mathbb{N} & \text{if } n = 0 \\ 2^{2 \uparrow (n-1)} & \text{otherwise} \end{cases}$$

The set $2 \uparrow \omega$ is defined by

$$2 \uparrow \omega = \bigcup_{n \in \mathbb{N}} 2 \uparrow n.$$

The case $n = 0$, i.e. $A \approx \mathbb{N}$, is considered to be the most interesting case. The case $n = 1$, i.e. $A \approx 2^{\mathbb{N}} \approx \mathbb{R}$, is also of interest when one considers real-time processes.

Theorem 4.7 *If $A \approx 2 \uparrow n$, for some n, then $P_1 \ncong P_2$, $P_2 \ncong P_3$, and $P_1 \ncong P_3$.*

The above theorem can be proved as follows. It can be demonstrated that P_1^*, P_2^*, and P_3^* are discrete spaces. Consequently, the *weight* of these spaces is equal to the cardinality of the spaces. Since the weight of the metric completion of a space is equal to the weight of the original space, the weight of $\widetilde{P_1^*}$, $\widetilde{P_2^*}$, and $\widetilde{P_3^*}$ is equal to the cardinality of P_1^*, P_2^*, and P_3^*. The weight of a space being smaller than some cardinal number is a topological property. Because the cardinality of P_2^* $(2 \uparrow n)$ is strictly smaller than the cardinality of P_3^* $(2 \uparrow (n+1))$ and the cardinality of P_3^* is strictly smaller than the cardinality of P_1^* $(2 \uparrow \omega)$, it can be concluded that $\widetilde{P_1^*}$, $\widetilde{P_2^*}$, and $\widetilde{P_3^*}$ are not isometric. From the theorems of the previous section immediately follows that P_1, P_2, and P_3 are not isometric.

5. Bisimulation

The process domains can be viewed as labelled transition systems. The bisimulation relations on these labelled transition systems induce bisimulation relations on the process domains. The process domains are proved to be strongly extensional, i.e. the largest bisimulation relation - bisimilarity - coincides with equality.

The process domain P_1 is turned into a labelled transition system of which the configurations are P_1-processes, the labels are actions, and the transition relation is defined by

$$p \xrightarrow{a} p' \text{ if and only if } (a, p') \in p.$$

Bisimilarity on the process domain P_1 coincides with equality as is shown in

Theorem 5.1 *P_1 is strongly extensional.*

Proof See Theorem 1 of [GR89]. □

A similar result is proved for the process domain P_2 in

Theorem 5.2 *P_2 is strongly extensional.*

Proof See [Rut92]. □

The process domain P_3 is turned into a labelled transition system of which the configurations are P_3-processes, the labels are actions, and the transition relation is defined by

$$p \xrightarrow{a} p' \text{ if and only if } p' \in p\,(a).$$

Also the process domain P_3 can be shown to be strongly extensional.

THEOREM 5.3 *P_3 is strongly extensional.*

PROOF Similar to the proofs of the Theorems 5.1 and 5.2. □

6. SEQUENTIAL COMPOSITION

Some complications arising in the definition of the sequential composition of P_1-processes are pinpointed. Furthermore, it is shown that these complications do not arise in the definition of the sequential composition of P_3-processes.

In Definition 4.4 of [BM88], the sequential composition of P_1-processes is defined by

DEFINITION 6.1 The operator $;: P_1 \times P_1 \to P_1$ is defined by

$$p\,;p' = \begin{cases} p' & \text{if } p = \emptyset \\ \{\,(a, p''\,; p') \mid (a, p'') \in p\,\} & \text{otherwise} \end{cases}$$

This definition coincides with the operational definition of the sequential composition. (Note that processes can be seen as labelled transition systems.) However, the above definition is not well-defined, as Warmerdam ([War90]) showed (cf. Appendix A).

Also in Definition 2.14 of [BZ82], the sequential composition of P_1-processes is defined.

DEFINITION 6.2 For a finite process p, $p;p'$ is defined as in Definition 6.1, and for an infinite process p,

$$p\,; p' = \lim_n (p\,[n]\,; p')$$

where $p\,[n]$ denotes the truncation of process p at depth n.

This definition is well-defined. However, the above definition does not coincide with the operational definition of the sequential composition (cf. Appendix A).

For P_3-processes, the sequential composition is defined in

DEFINITION 6.3 The operator $;: P_3 \times P_3 \to P_3$ is defined by

$$p\,;p' = \begin{cases} p' & \text{if } p = \lambda a \cdot \emptyset \\ \lambda a \cdot \{\,p''\,; p' \mid p'' \in p\,(a)\,\} & \text{otherwise} \end{cases}$$

The well-definedness of the above definition of the sequential composition can be proved along the lines of the well-definedness proof of the embedding i_1 in the fourth section of this paper.

Also in the definitions of the operators parallel composition, trace set, and fairification on P_1-processes similar complications arise (cf. [BK87, BBKM84, Bre94]). These complications do not arise in the definitions of the operators on P_3-processes (cf. [Bre94]). Also process domain P_2 does not give rise to these complications (cf. [KR90]). However, unlike process domain P_3, process domain P_2 does not allow an elementary modelling of image finite language constructions like random assignment (cf. [Bre94]).

Concluding remarks

In this concluding section, some related work is discussed and some points for further research are mentioned.

A fourth process domain P_4 defined by the recursive domain equation $P_4 \cong A \to \mathcal{P}_{cl}(id_{\frac{1}{2}}(P_4))$ is considered in [Bre94]. The process domain P_4 can be shown to be isometric to the process domain P_1 (independent of the size of the action set A).

An alternative metric process domain is introduced by Golson and Rounds in [GR83, Gol84]. The processes are Milner's rigid synchronization trees endowed with a pseudometric. The pseudometric is induced by the (strong) behavioural equivalence relation introduced in [Mil80]. This behavioural equivalence relation and the bisimilarity equivalence relation considered in Section 5 do not coincide (cf. [Mil90]). Golson and Rounds show that their process domain is isometric to the process domain P_1 in case the action set is finite or countably infinite (for the countably infinite case, the power set construction used in the domain equation defining P_1 should be restricted to the collection of countable subsets).

In [Ole87], Oles defines a denotational semantics for a nonuniform language with the so-called angelic choice operator. The mathematical domain of this denotational semantics is defined as the solution of a recursive domain equation over bounded complete directed sets. For a uniform language with the conventional choice operator, the mathematical domain defined by the recursive domain equation $P \cong A \to \mathcal{P}_{fi}(P)$ has been suggested ([Ole92]). This domain equation shows some resemblance with the domain equation for process domain P_3.

Some topics for further research are the study of the process domains P_1, P_2, and P_3 with the action set endowed with an arbitrary complete metric instead of the discrete metric, and process domains corresponding to general, finitely branching, and image finite processes for complete partial orders and non-well-founded sets.

Acknowledgements

The author would like to thank Jaco de Bakker, Jan Rutten, and Fer-Jan de Vries for several discussions and their comments on a preliminary version of this paper. Furthermore, the author is grateful to Marcello Bonsangue, Frank Oles, Daniele Turi, and Erik de Vink for discussion.

References

[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.

[Abr91] S. Abramsky. A Domain Equation for Bisimulation. *Information and Computation*, 92(2):161–218, June 1991.

[Acz88] P. Aczel. *Non-Well-Founded Sets*. Number 14 in CSLI Lecture Notes. Centre for the Study of Languages and Information, Stanford, 1988.

[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.

[AR92] P. America and J.J.M.M. Rutten. A Layered Semantics for a Parallel Object-Oriented Language. *Formal Aspects of Computing*, 4(4):376–408, 1992.

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

[BBKM84] J.W. de Bakker, J.A. Bergstra, J.W. Klop, and J.-J.Ch. Meyer. Linear Time and Branching Time Semantics for Recursion with Merge. *Theoretical Computer Science*, 34(1/2):135–156, 1984.

[BK87] J.A. Bergstra and J.W. Klop. A Convergence Theorem in Process Algebra. Report CS-R8733, CWI, Amsterdam, July 1987. Appeared in [BR92], pages 164–195.

[BM88] J.W. de Bakker and J.-J.Ch. Meyer. Metric Semantics for Concurrency. *BIT*, 28:504–529, 1988.

[BR92] J.W. de Bakker and J.J.M.M. Rutten, editors. *Ten Years of Concurrency Semantics, selected papers of the Amsterdam Concurrency Group*. World Scientific, Singapore, 1992.

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

[BW93] F. van Breugel and J.H.A. Warmerdam. Solving Recursive Domain Equations in a Category of Compact Metric Spaces. CWI, Amsterdam. Preprint, to appear.

[BZ82] 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.

[BZ83] J.W. de Bakker and J.I. Zucker. Compactness in Semantics for Merge and Fair Merge. In E. Clarke and D. Kozen, editors, *Proceedings of 4th Workshop on Logics of Programs*, volume 164 of *Lecture Notes in Computer Science*, pages 18–33, Pittsburgh, June 1983. Springer-Verlag.

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

[Gla90] R.J. van Glabbeek. The Linear Time - Branching Time Spectrum. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings of CONCUR'90*, volume 458 of *Lecture Notes in Computer Science*, pages 278–297, Amsterdam, August 1990. Springer-Verlag.

[Gla93] R.J. van Glabbeek. The Linear Time - Branching Time Spectrum II. To appear in *Proceedings of CONCUR'93*, Hildesheim, August 1993.

[Gol84] W.G. Golson. *Denotational Models based on Synchronous Communicating Processes*. PhD thesis, University of Michigan, Ann Arbor, 1984.

[GR83] W.G. Golson and W.C. Rounds. Connections between Two Theories of Concurrency: Metric Spaces and Synchronization Trees. *Information and Control*, 57(2/3):102–124, May/June 1983.

[GR89] R.J. van Glabbeek and J.J.M.M. Rutten. The Processes of De Bakker and Zucker represent Bisimulation Equivalence Classes. In *J.W. de Bakker, 25 jaar semantiek*, pages 243–246. CWI, Amsterdam, April 1989.

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

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

[Mil90] R. Milner. Operational and Algebraic Semantics of Concurrent Processes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, chapter 19, pages 1201–1242. The MIT Press/Elsevier, Cambridge/Amsterdam, 1990.

[MM79] G. Milne and R. Milner. Concurrent Processes and Their Syntax. *Journal of the ACM*, 26(2):302–321, April 1979.

[Ole87] F.J. Oles. Semantics for Concurrency without Powerdomains. In *Proceedings of the 14th Annual ACM Symposium on Principles of Programming Languages*, pages 211–222, Munich, January 1987.

[Ole92] F.J. Oles, August 1992. Personal communication.

[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.

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

[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.

[RZ92] J.J.M.M. Rutten and J.I. Zucker. A Semantic Approach to Fairness. *Fundamenta Informaticae*, 16(1):1–38, January 1992.

[War90] J.H.A. Warmerdam, November 1990. Personal communication.

A. Warmerdam's counterexample

Warmerdam ([War90]) showed that the sequential composition of P_1-processes as defined in Definition 4.4 of [BM88] (cf. Definition 6.1) is not well-defined by proving that the set

$$\{ (a, p'' \,;\, p') \mid (a, p'') \in p \}$$

is in general not closed. Here, Warmerdam's counterexample is presented. Furthermore, this counterexample is used to illustrate that the sequential composition as defined in Definition 2.14 of [BZ82] (cf. Definition 6.2) does not correspond to the operational definition of the sequential composition.

Let P_1-process p be defined by

$$p = \{ (a, p_n) \mid n \in \mathbb{N} \}$$

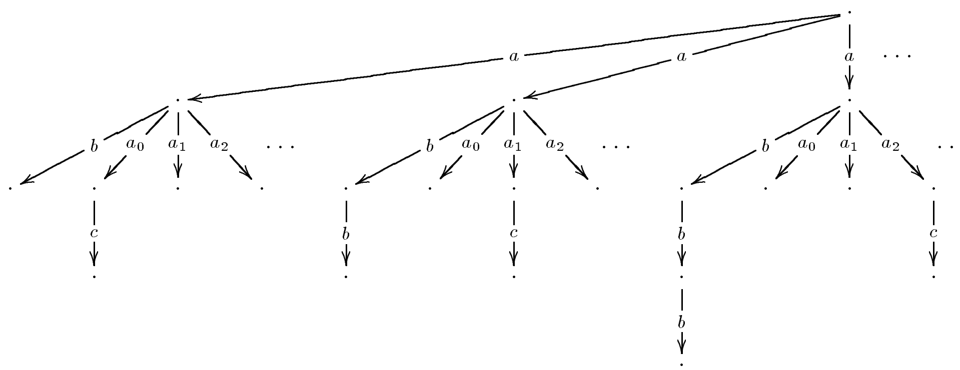
where

$$p_n = \{b^n, (a_0, \emptyset), \ldots, (a_{n-1}, \emptyset), (a_n, \{(c, \emptyset)\}), (a_{n+1}, \emptyset), \ldots\}$$

and

$$b^n = \begin{cases} (b, \emptyset) & \text{if } n = 0 \\ (b, \{b^{n-1}\}) & \text{otherwise} \end{cases}$$

This P_1-process p is depicted by



Let P_1-process p' be defined by

$$p' = \{\lim_n c^n\}.$$

This P_1-process p' is depicted by

According to Definition 4.4 of [BM88] (cf. Definition 6.1), the sequential composition of the P_1-processes p and p' is defined by

$$p\,;p' = \{\,(a, p''_n) \mid n \in \mathbb{N}\,\}$$

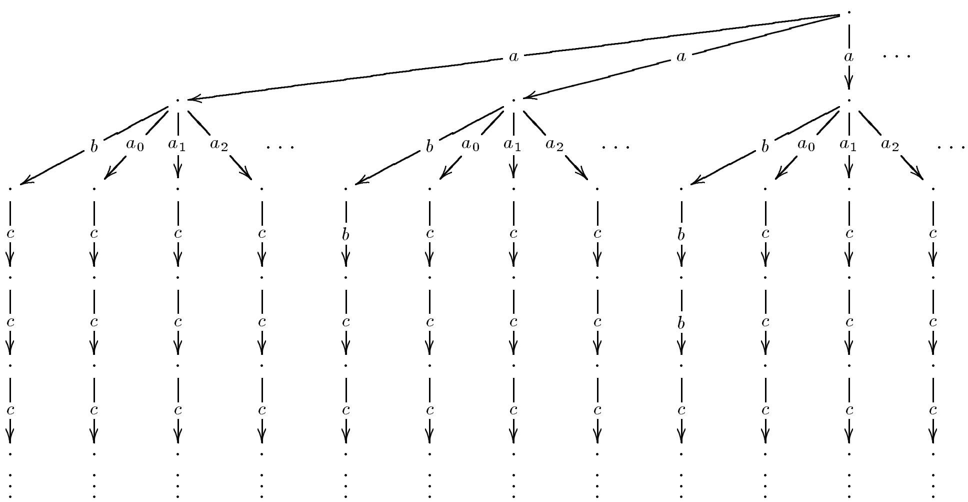
where

$$p''_n = \{b^n\,;p', (a_0, p'), (a_1, p'), \ldots\}$$

and

$$b^n\,;p' = \begin{cases} (b, p') & \text{if } n = 0 \\ (b, \{b^{n-1}\,;p'\}) & \text{otherwise} \end{cases}$$

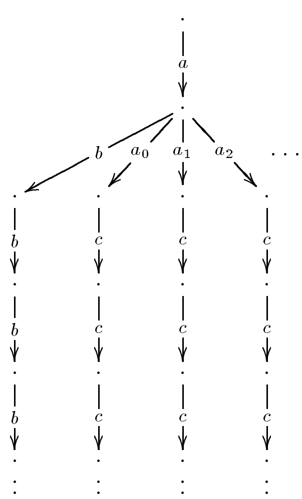
This process $p\,;p'$ is depicted by



However, $p\,;p'$ is not a P_1-process, since the set $p\,;p'$ is not closed. The set $p\,;p'$ contains the Cauchy sequence $((a, p''_n))_n$ but not its limit (a, p'') where

$$p'' = \{\lim_n b^n, (a_0, p'), (a_1, p'), \ldots\}$$

which is depicted by



The above counterexample also shows that the limit construction in the definition of the sequential composition presented in Definition 2.14 of [BZ82] (cf. Definition 6.2) adds unexpected subprocesses; the limit construction $\lim_n (p\,[n]\,;p')$ adds subprocess (a, p'').