A CONVERGENCE THEOREM IN PROCESS ALGEBRA

J.A. Bergstra
University of Amsterdam, Department of Computer Science
P.O. Box 19268, 1000 GG Amsterdam;
State University of Utrecht, Department of Philosophy,
P.O. Box 8810, 3508 TA Utrecht.

J.W. Klop
Centre for Mathematics and Computer Science,
P.O. Box 4079, 1009 AB Amsterdam;
Free University, Department of Mathematics and Computer Science,
De Boelelaan 1081, 1081 HV Amsterdam.

ABSTRACT
We study a convergence phenomenon in the projective limit model $\mathbb{A}^\infty$ for PA, an axiom system in the framework of process algebra for processes built from atomic actions by means of alternative composition (+) and sequential composition (·), and subject to the operations || (merge) and ⌊⌊ (left-merge). The model $\mathbb{A}^\infty$ is also a complete metric space. Specifically, it is shown that for every element $q \in \mathbb{A}^\infty$ the sequence q, s(q), $s^2(q)$, ... , $s^n(q)$, ... converges to a solution of the (possibly unguarded) recursion equation X = s(X) where s(X) is an expression in the signature of PA involving the recursion variable X. As the convergence holds for arbitrary starting points q, this result does not seem readily obtainable by the usual convergence proof techniques. Furthermore, the connection is studied between projective models and models based on process graphs. Also these models are compared with the process model introduced by De Bakker and Zucker.

Key words and phrases: process algebra, projective limit model, merge, left-merge, recursion equations, complete metric space, process graph, Approximation Induction Principle.
1985 Mathematics subject classification: 68Q05, 68Q10, 68Q55, 68Q45.
1987 CR Categories: F.1.2, F.3.1, F.3.2, F.3.3.
Note: Partial support received from the European Communities under ESPRIT contract 432, An Integrated Formal Approach to Industrial Software Development (Meteor).

Introduction

The present paper is a revised and extended version of [BK82], which was written as a response to a question of De Bakker and Zucker [BZ82a,b], namely how to assign a semantics in their process domain to certain fixed point expressions μX.s(X) where s(X) is an expression in the signature specified below. In case μX.s(X) is a 'guarded' fixed point expression it is, as shown in [BZ82a,b], straightforward to define the appropriate semantics, using Banachs fixed point theorem for complete metric spaces, but there is a problem in the unguarded case - at least, if one wishes the semantics to be a solution of the recursion equation X = s(X). In order to tackle the problem, we devised in [BK82] an axiom system called below PA (for Process Algebra), together with a 'projective limit model' $\mathbb{A}^\infty$ for these axioms, and showed that every iteration sequence

Report CS-R8733
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

$$q,\ s(q),\ s^2(q),\ \dots,\ s^n(q),\ \dots$$

where q is a finite process and s(X) is an expression in the signature of PA involving occurrences of the recursion variable X, converges to a solution of the recursion equation X = s(X). Here we can speak about convergence, since the projective limit model is also a complete metric space. In the case of a guarded expression s(X) this solution is unique; in the unguarded case there exists a solution but not necessarily a unique one - as is readily seen by contemplating the recursion equation X = X.

The axiom system PA has been a stepping stone towards more and more expressive axiom systems for processes, covering also process features such as handshaking communication, deadlock, abstraction, priorities between atomic actions etc. Some introductory surveys of this work are given in [BK86a,b]. In the course of pursuing this line of 'process algebra', it was found that there is at least one easier way to prove the existence of solutions of recursion equations X = s(X), even if unguarded. For such a proof, based on a method of Milner [Mi85], we refer to a paper by Van Glabbeek [Gl87]. (See p.344 last line. Some work remains to be done to transpose the result there to the present setting, however.) Yet we find that the proof below is worthwhile, since it gives more information than the mere existence of solutions of X = s(X): as already stated, such solutions can be found by iteration from an *arbitrary* initial process. The interesting point is that the usual convergence proof methods, such as appealing to Banachs fixed point theorem for complete metric spaces or the Tarski-Knaster fixed point theorem for complete partial orders, do not seem to yield this additional information. A challenging question, which for us is open, is to analyze the convergence result below in terms of 'more general' theory. The proof below, which is 'combinatorial' in nature, rests upon the specific algebraic properties of the operators defined by PA. On the other hand, there seems to be a more general convergence principle involved: we expect that analogous convergence theorems can be proved for more extensive process axiomatisations such as ACP, Algebra of Communicating Processes (see [BK86a,b]).

In the second part of the paper we define projective models for arbitrary large alphabets; also here the convergence theorem holds. Furthermore, we compare these projective models with models obtained via process graphs ('graph models'), and with the process model of 'hereditarily closed sets' introduced by De Bakker and Zucker in [BZ82a,b].

Acknowledgements. We are indebted to J.W. de Bakker for introducing us to a challenging and fruitful question, which was the initial motivation for our paper, as well as for valuable comments. We also thank J.C.M. Baeten, E. Kranakis and R.J. van Glabbeek for useful discussions.

Contents

1. The axiom system PA

In this paper we will discuss 'processes' built from atomic actions or events a,b,c,... , by means of the basic operators + (*alternative composition*) and · (*sequential composition*). We will adopt the restriction that the *action alphabet* A = {a,b,c,...} is *finite*. (The case of infinite alphabets will be studied in Section 4.) Intuitively, a process expression as e.g. (a + b)·c will denote a process capable first of choosing between the actions a,b and executing the chosen action, and second performing the action c. Actually this process expression will be considered equivalent (that is, denoting the same process) to a·c+ b·c. On the other hand we do not wish to identify processes c·(a + b) and c·a + c·b as these processes differ in their timing of choices. Apart from the two basic operators we introduce a *merge* or interleaving operator || together with an auxiliary operator ⌊⌊ called *left-merge*. These four operators will be subject to the axioms in Table 1, where a ∈ A and x,y,z are variables denoting general processes.

PA		
	x+y = y+x	A1
	(x+y)+z = x+(y+z)	A2
	x+x = x	A3
	(xy)z = x(yz)	A4
	(x+y)z = xz+yz	A5
	x‖y = x⌊⌊y + y⌊⌊x	M1
	ax⌊⌊y = a(x‖y)	M2
	a⌊⌊y = ay	M3
	(x+y)⌊⌊z = x⌊⌊z + y⌊⌊z	M4

Table 1

Here we have suppressed the product sign ·, as we will do henceforth, and we use the convention that · binds stronger than the other operators; so ax⌊⌊y stands for (a·x)⌊⌊y.

A model of this axiom system PA will be called a *process algebra* (for PA). The elements in a process algebra are processes. The simplest process algebra is the closed term model $\mathbb{A}_\omega$, with as elements the closed terms (or closed expressions) in the signature of PA modulo the equality generated by the axioms of PA. The word 'closed' refers to the absence of variables. One easily establishes the following facts:

1.1. PROPOSITION. *For all process algebras:*

$$\Sigma_{i=1,..,n}\, a_i x_i \parallel \Sigma_{j=1,..,m}\, b_j y_j =$$

$$\Sigma_{i=1,..,n}\, a_i(x_i \parallel \Sigma_{j=1,..,m}\, b_j y_j) + \Sigma_{j=1,..,m}\, b_j(y_j \parallel \Sigma_{i=1,..,n}\, a_i x_i). \quad \square$$

1.2. PROPOSITION. (Representation of elements of $\mathbb{A}_\omega$) *Modulo the equivalence generated by the axioms of* PA, *the term algebra* $\mathbb{A}_\omega$ *is inductively generated as follows:*

$$x_i \in \mathbb{A}_\omega,\ a_i \in A\ (i = 1,...,n),\ b_j \in A\ (j = 1,...,m) \Rightarrow$$

$$(\Sigma_{j=1,..,m}\, b_j + \Sigma_{i=1,..,n}\, a_i x_i) \in \mathbb{A}_\omega. \ \square$$

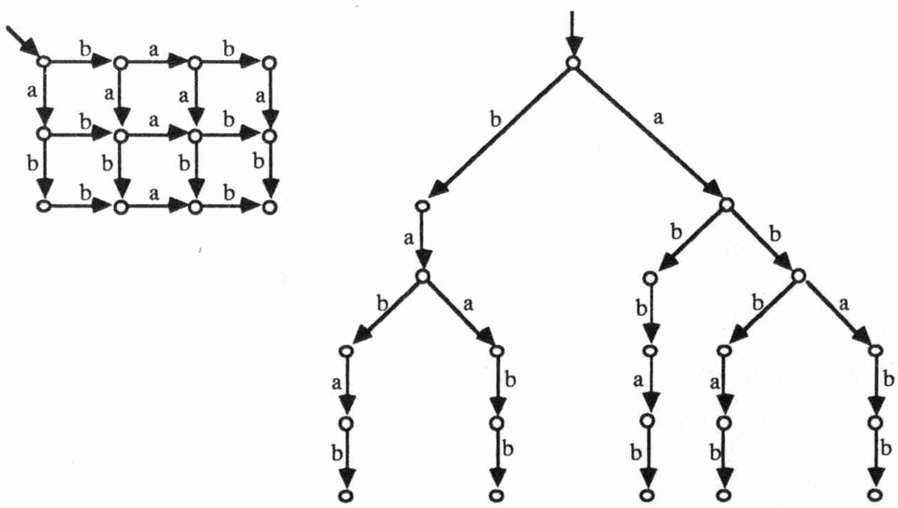
As to Proposition 1.1, we remark that it does not seem possible to avoid the cumbersome explicit sum formula without using an auxiliary operator such as $\lfloor\!\lfloor$. We conjecture that process algebras without $\lfloor\!\lfloor$ and using $\|$, are not finitely axiomatizable.

The elements of $\mathbb{A}_\omega$ can also be pictured as (equivalence classes of) finite trees, or directed acyclic graphs.

1.3. EXAMPLE.

$bab \| ab =$
$bab \lfloor\!\lfloor ab + ab \lfloor\!\lfloor bab = b(ab \| ab) + a(b \| bab) =$
$b(ab \lfloor\!\lfloor ab + ab \lfloor\!\lfloor ab) + a(b \lfloor\!\lfloor bab + bab \lfloor\!\lfloor b) =$
$b(ab \lfloor\!\lfloor ab) + a(bbab + b(ab \| b)) =$
$b(a(b \| ab)) + a(bbab + b(abb + bab)) =$
$b(a(bab + abb)) + a(bbab + b(abb + bab)).$

The first expression, bab $\|$ ab, corresponds to the 'cartesian product' graph as in Figure 1a, the last expression in which the merge and left-merge operators have been eliminated, corresponds to the tree in Figure 1b which is the 'unshared' version of the graph in Figure 1a. Actually one can construct process algebras for PA starting from a domain of process graphs or process trees as in



(a) (b)

Figure 1

Figure 1, and next dividing out a suitable equivalence relation ('bisimulation equivalence', see Definition 4.4). We will explain this construction in Section 4. If the domain of process graphs or trees consists of acyclic and finite trees, the resulting quotient algebra is isomorphic to $\mathbb{A}_\omega$.

On the elements $x \in \mathbb{A}_\omega$ we define the following norm $\nu(x)$, which intuitively is the minimum of the length of the 'branches' of the tree of x (as in Figure 1b).

1.4. DEFINITION. For $x \in \mathbb{A}_\omega$, we define $\nu(x)$ by:

(i) $\nu(a) = 1$ for $a \in A$,
(ii) $\nu(x+y) = \min\{\nu(x), \nu(y)\}$,
(iii) $\nu(ax) = 1 + \nu(x)$.

The following proposition says that merging will certainly not lead to shorter branches. (In fact, the proposition holds with '>' instead of '≥', but we will not need that). The routine proof is left to the reader.

1.5. PROPOSITION. *For all* $x,y \in \mathbb{A}_\omega$: $\nu(x \parallel y) \geq \nu(x), \nu(y)$. □

In the next proposition we establish some useful identities valid in $\mathbb{A}_\omega$ (needed in Section 2), again without the routine proofs. First some more notation:

1.6. NOTATION. (i) $x^1 \equiv x$; $x^{n+1} \equiv xx^n$ $(n \geq 1)$
(ii) $x^{\underline{1}} \equiv x$; $x^{\underline{n+1}} \equiv x \parallel x^{\underline{n}}$ $(n \geq 1)$

1.7. PROPOSITON. *In* $\mathbb{A}_\omega$ *the following identities are valid:*

(i) $(x \parallel y) \parallel z = x \parallel (y \parallel z)$
(ii) $(x ⫡ y) ⫡ z = x ⫡ (y \parallel z)$
(iii) $x \parallel y \parallel z = x ⫡ (y \parallel z) + y ⫡ (x \parallel z) + z ⫡ (x \parallel y)$
(iv) $x_1 \parallel x_2 \parallel \ldots \parallel x_n =$
$x_1 ⫡ (x_2 \parallel \ldots \parallel x_n) + x_2 ⫡ (x_1 \parallel x_3 \parallel \ldots \parallel x_n) + \ldots + x_n ⫡ (x_1 \parallel \ldots \parallel x_{n-1})$ $(n \geq 2)$
(v) $x^{\underline{n+1}} = x ⫡ x^{\underline{n}}$ $(n \geq 1)$.

PROOF. (v) follows directly from (iv), which generalizes (iii); (iv) follows via simple algebraic manipulations from (i) and (ii). Statements (i) and (ii) can be proved simultaneously using induction on the structure of $x,y,z \in \mathbb{A}_\omega$ according to Proposition 1.2. □

2. The projective model for PA

We will now introduce a process algebra for PA that also contains infinite processes. First we need projection operators:

2.1. DEFINITION. (i) On $\mathbb{A}_\omega$ we define for each $n \geq 1$ the *projection* $(\,)_n: \mathbb{A}_\omega \to \mathbb{A}_\omega$ as follows. (Intuitively, $(\,)_n$ cuts of the 'tree' of x at level n.) For each $a \in A$ and $x,y \in \mathbb{A}_\omega$:

$$(a)_n = a,$$
$$(ax)_1 = a,$$
$$(ax)_{n+1} = a(x)_n,$$
$$(x+y)_n = (x)_n + (y)_n.$$

(ii) $\mathbb{A}_n = \{(x)_n \mid x \in \mathbb{A}_\omega\}$.

(iii) Instead of $(x)_n = (y)_n$ we will also say: $x = y$ *modulo* n.

2.2. EXAMPLE. Modulo 3 we have:

$$[\,(a^3 \,\|\, b^3) + a^3] \,\|\, b^3 \;=\; a^3 \,\|\, b^3 \;=\; (a+b)^3.$$

The following proposition is easily established, and we omit the proof. Note especially the occurrences of n - 1 in (iii) and (iv):

2.3. PROPOSITION. *For all* $x,y \in \mathbb{A}_\omega$:

(i)	$((x)_n)_m = (x)_{\min(n,m)}$	$(n,m \geq 1)$
(ii)	$(x+y)_n = ((x)_n + (y)_n)_n$	$(n \geq 1)$
(iii)	$(xy)_n = ((x)_n (y)_{n-1})_n$	$(n \geq 2)$
(iv)	$(x \mathbin{\lfloor\!\lfloor} y)_n = ((x)_n \mathbin{\lfloor\!\lfloor} (y)_{n-1})_n$	$(n \geq 2)$
(v)	$(x \,\|\, y)_n = ((x)_n \,\|\, (y)_n)_n$	$(n \geq 1)$
(vi)	$(xy)_1 = (x)_1$	
(vii)	$(x \mathbin{\lfloor\!\lfloor} y)_1 = (x)_1.$	$\square$

2.4. REMARK. Note that the $\mathbb{A}_n$ are also process algebras with operators $+_n$, $\cdot_n$, $\lfloor\!\lfloor_n$, $\|_n$ for PA; the operators are defined by:

$$x \,\square_n\, y \;=\; (x \,\square\, y)_n,$$

where $\square$ is $+, \cdot, \lfloor\!\lfloor, \|$. In fact, $+_n$ coincides with $+$.

2.5. DEFINITION. Let $q_i \in \mathbb{A}_\omega$ ($i \geq 1$). Then the sequence $q_1, q_2, \ldots$ is called *projective* if for all i:

$$q_i = (q_{i+1})_i.$$

2.6. DEFINITION. $\mathbb{A}^\infty$ is the *projective limit* of the $\mathbb{A}_n$ ($n \geq 1$); the elements of $\mathbb{A}^\infty$ are the projective sequences.

It is not hard to establish that $\mathbb{A}^\infty$ is a process algebra for PA where the operations are defined component-wise. It will be called the *projective limit model* or *projective model.*

2.7. EXAMPLE. (i) $(a, a+a^2, a+a^2+a^3, \ldots) \in \mathbb{A}^\infty$.

(ii) $(a, a^2, a^3, \ldots) \cdot (b, b^2, b^3, \ldots) = ((ab)_1, (a^2b^2)_2, (a^3b^3)_3, \ldots) = (a, a^2, a^3, \ldots)$.

(iii) $(a, a+a^2, a+a^2+a^3, \ldots) \cdot (b, b+b^2, b+b^2+b^3, \ldots) = (a, ab+a^2, a(b+b^2) + a^2b+a^3, \ldots)$.

(iv) $(a, a^2, a^3, \ldots) \| (b, b^2, b^3, \ldots) = ((a\|b)_1, (a^2\|b^2)_2, (a^3\|b^3)_3, \ldots) = (a+b, (a+b)^2, (a+b)^3, \ldots)$.

3. Iteration sequences

In this section we will show that every iteration sequence q, s(q), s(s(q)), ... must eventually be constant modulo n, for every $n \geq 1$. We will also say that the sequence 'stabilizes' modulo n.

3.1. DEFINITION. The set EXP of *(possibly open) process expressions* is defined (in BNF notation) by:

$$s ::= \quad a, b, c, \ldots \quad | \quad X, Y, Z, \ldots \quad | \quad s_1 + s_2 \quad | \quad s_1 \cdot s_2 \quad | \quad s_1 \lfloor\!\lfloor s_2 \quad | \quad s_1 \| s_2 .$$

Here $a, b, c, \ldots \in \mathbb{A}$ and $X, Y, Z, \ldots$ are recursion variables.

3.2. DEFINITION. (i) Let $s(X) \in$ EXP be an expression containing no other variables than X. Let $q \in \mathbb{A}_\omega$. Then the sequence

$$q, s(q), s(s(q)), \ldots, s^k(q), \ldots$$

is called the *iteration sequence generated by* s(X) *from* q.

(ii) The sequence $q_1, q_2, \ldots, q_k, \ldots$ ($q_i \in \mathbb{A}_\omega$, $i \geq 1$) is said to *stabilize modulo* n if the sequence stabilizes in $\mathbb{A}_n$, i.e. if

$$(q_1)_n,\ (q_2)_n,\ \ldots,\ (q_k)_n,\ \ldots$$

is eventually constant.

In order to prove the main theorem of this section, we need some propositions.

3.3. PROPOSITION. *For every* $q \in \mathbb{A}_\omega$ *and* $n \geq 1$, *the iteration sequence*

$$q,\ q\|q,\ q\|q\|q,\ \ldots,\ q^{\underline{k}},\ \ldots$$

stabilizes modulo n.

PROOF. Induction on n. *Basis*: $n = 1$. One easily computes:

$$(q)_1 = \Sigma a_i = (q\|q)_1 = \ldots = (q^{\underline{k}})_1 = \ldots$$

for some sum Σa_i. *Induction step*. Suppose the proposition is proved for n - 1. By Proposition 1.7(v):

$$q^{\underline{k+1}} = q \lfloor\!\lfloor\, q^{\underline{k}}.$$

By Proposition 2.3(iv):

$$(q^{\underline{k+1}})_n = (q \lfloor\!\lfloor q^{\underline{k}})_n = ((q)_n \lfloor\!\lfloor (q^{\underline{k}})_{n-1})_n.$$

By the induction hypothesis, $(q^{\underline{k}})_{n-1} = p$ for some fixed p for all but finitely many k. Hence the sequence stabilizes indeed modulo n, viz. in $((q)_n \lfloor\!\lfloor p)_n$. □

The next two propositions generalize the preceding one considerably.

3.4. PROPOSITON. *Let the action alphabet* A *be finite. Let* $q_1, q_2, \ldots$ *be a sequence in* $\mathbb{A}_\omega$ *such that for all* $i \geq 1$: $q_{i+1} = q_i \| r_i$ *for some* r_i.

Then the sequence $q_1, q_2, \ldots$ *stabilizes modulo* n.

PROOF. By assumption, $q_k = q_1 \| r_1 \| r_2 \| r_3 \| \ldots \| r_{k-1}$ $(k \geq 2)$, hence by Proposition 2.3(v):

$$(q_k)_n = ((q_1)_n \| (r_1)_n \| \ldots \| (r_{k-1})_n)_n.$$

Here all $(r_i)_n$ are elements of the *finite* $\mathbb{A}_n$. (Obviously, since A is finite, every $\mathbb{A}_n$ is finite). Say $\mathbb{A}_n = \{p_1,...,p_N\}$. Then by associativity and commutativity of ||, we can write

$$(q_k)_n = ((q_1)_n \,||\, p_1 \overline{}^{f_1(k)} \,||\, p_2 \overline{}^{f_2(k)} \,||\, ... \,||\, p_N \overline{}^{f_N(k)})_n$$

for some *monotonic* functions f_i (i=1,...,N), with the understanding that if $f_i(k) = 0$, the corresponding 'mergend' vanishes. By Proposition 3.3, every

$$p_i \overline{}^{f_i(k)}$$

(i = 1,...,N) stabilizes modulo n, with growing k; whence the result follows. □

3.5. PROPOSITION. *Let* A *be finite. Let* $q_1, q_2, ...$ *be a sequence in* $\mathbb{A}_\omega$ *such that for all* $i \geq 1$, *either*

(i) $q_{i+1} = q_i || r_i$, *or*

(ii) $q_{i+1} = q_i \cdot r_i$

for some r_i. *Then the sequence* $q_1, q_2, ...$ *stabilizes modulo* n.

PROOF. We may suppose that for infinitely many i we are in case (ii); otherwise we are done at once using Proposition 3.4.

So by Proposition 1.5, $\nu(q_i) \geq n$, and hence $\nu((q_i)_n) = n$, for all but finitely many i. (Here we use also the obvious fact: $\nu(q_i r_i) \geq \nu(q_i)$.) Now if $\nu((q_i)_n) = n$, and $q_{i+1} = q_i r_i$, then evidently $(q_{i+1})_n = (q_i)_n$. That is, modulo n, right multiplication has no effect from some i onwards. But then we are again in the case of the previous proposition. □

3.5.1. REMARK. If in Proposition 3.5, (ii) is replaced by: (ii) $q_{i+1} = q_i + r_i$, then the resulting proposition is no longer true. Cfr. Example 2.2.

3.5.2. REMARK. A corollary of Proposition 3.5 is that in every $\mathbb{A}_n$ as well as in $\mathbb{A}^\infty$, if A is finite:

$$\exists x \ \forall y \ \ x||y = x,$$

i.e. there exists an element which is "saturated" w.r.t. merges.

3.6. PROPOSITION. *Let* A *be finite. Let* $q_1, q_2, ...$ *be a sequence in* $\mathbb{A}_\omega$ *such that for all* $i \geq 1$, *either*

(i) $q_{i+1} = q_i \Lsh r_i$, *or*

(ii) $q_{i+1} = q_i \cdot r_i$

for some r_i. *Then the sequence* $q_1, q_2, \ldots$ *stabilizes modulo* n.

PROOF. By Proposition 1.2, we have $q_1 = \Sigma a_i + \Sigma b_j x_j$ for some $a_i, b_j \in A$ and $x_j \in \mathbb{A}_\omega$. Now if $q_2 = q_1 r_1$, then

$$q_2 = \Sigma a_i r_1 + \Sigma b_j x_j r_1,$$

and if $q_2 = q_1 \Lsh r_1$, then

$$q_2 = \Sigma a_i r_1 + \Sigma b_j (x_j \| r_1).$$

In both cases q_2 has the form, say, $\Sigma c_k p_k$ for some $c_k \in A$, $p_k \in \mathbb{A}_\omega$. Now suppose, e.g.:

$$q_3 = q_2 \Lsh r_2$$
$$q_4 = q_3 r_3$$
$$q_5 = q_4 \Lsh r_4$$
$$q_6 = q_5 \Lsh r_5$$
$$q_7 = q_6 r_6$$
$$\ldots$$

(so for instance $q_7 = ((((q_2 \Lsh r_2) r_3) \Lsh r_4) \Lsh r_5) r_6)$. Then

$$q_3 = (\Sigma c_k p_k) \Lsh r_2 = \Sigma c_k (p_k \| r_2),$$
$$q_4 = (\Sigma c_k (p_k \| r_2)) r_3 = \Sigma c_k (p_k \| r_2) r_3$$
$$\ldots$$

(so $q_7 = \Sigma c_k [((((p_k \| r_2) r_3) \| r_4) \| r_5) r_6]$). Hence an appeal to the previous proposition yields the result. □

3.6.1. REMARK. The generality in Propositions 3.5 and 3.6 w.r.t. the elements r_i, suggests looking at possible stabilization (modulo n) of general sequences of the forms:

(i) $q, s_1(q), s_1(s_2(q)), s_1(s_2(s_3(q))), \ldots$

(ii) $q, s_1(q), s_2(s_1(q)), s_3(s_2(s_1(q))), \ldots$

where $q \in \mathbb{A}_\omega$ and $s_i(X)$ $(i \geq 1)$ are arbitrary expressions $\in$ EXP having only X free.

Both types of sequences do not necessarily stabilize, however. For (i) one may take

$s_{2n+1}(X) = Xa$, $s_{2n+2}(X) = Xb$ ($n \geq 0$) and q = a. This sequence does not stabilize modulo 2. For (ii): take $s_{2n+1}(X) = X+a^3$, $s_{2n+2}(X) = X \| b^3$ ($n \geq 0$) and $q = a^3 \| b^3$. This sequence does not stabilize modulo 3 as already remarked in 3.5.1.

We will now state and prove the main theorem of this paper, saying that every sequence q, s(q), $s^2(q)$, ... must eventually be constant modulo n. For *guarded* expressions like e.g. $s(X) = aX + b(cX \| X^3) + d$ this is clear since iterating s(X) yields a tree which develops itself in such a way that an increasing part of it is fixed. 'Guarded' means that an occurrence of a recursion variable cannot be accessed without passing an atom. But even for simple terms as $s(X) = (X \| X) + ab$ the situation is at first sight not at all clear: in each step of the iteration the whole tree including the top is again in 'motion'.

3.7. THEOREM. *Let* $q \in \mathbb{A}_\omega$ *and let* $s(X) \in$ EXP *have only* X *as free variable. Then the iteration sequence* q, s(q), s(s(q)), ..., $s^k(q)$, ... *stabilizes modulo* n, *for every* $n \geq 1$.

PROOF. The proof is by induction on n. *Basis*: n = 1. By Proposition 2.3, $(s(X))_1 = \Sigma a_i$ or $(s(X))_1 = (X)_1 + \Sigma a_i$. E.g. if $s(X) = X \Join X + a \Join X + bcX$, then

$$(s(X))_1 = (X \Join X)_1 + (a \Join X)_1 + (bcX)_1 = (X)_1 + a + b.$$

In the first case the iteration sequence stabilizes modulo 1 at Σa_i, in the second case at $(q)_1 + \Sigma a_i$.
Induction step. Induction hypothesis: suppose the statement in the theorem is proved for n - 1. Consider s(X). It has the following form, possibly after some rewritings by means of axioms A5, M4:

$$\begin{array}{l}
X\square t_1\square t_2 \ldots \square t_{_} \quad + \\
X\square t_{_}\square t_{_}\square \ldots \square t_{_} \quad + \\
\vdots \\
X\square t_{_}\square \ldots\ldots \square t_{_} \quad + \\
a_1\square t_{_}\square \ldots\ldots \square t_{_} \quad + \\
a_1\square t_{_}\square \ldots\ldots \square t_{_} \quad + \\
\vdots \\
a_1\square t_{_}\square \ldots\ldots \square t_{_} \quad + \\
\vdots \\
a_k\square t_{_}\square \ldots\ldots \square t_{_} \quad + \\
\vdots \\
a_k\square t_{_}\square \ldots\ldots \square t_{_}.
\end{array}$$

Here $\square$ is either $\Join$ or $\cdot$, $a_1,\ldots,a_k \in$ A and $t_1,t_2,t_{\_},t_{\_},\ldots \in$ EXP. (The reader is invited to write the appropriate subscripts for the _ in $t_{\_}$). In each summand brackets associate to the left.

In order to avoid excessive notation, we will give the remainder of the proof using as a typical example

$$s(X) = ((X \mathbin{\|\!\!\_}\, t_1)t_2) \mathbin{\|\!\!\_}\, t_3 + (X \mathbin{\|\!\!\_}\, t_4) \parallel t_5 + a \mathbin{\|\!\!\_}\, t_6.$$

Note that $t_1,\ldots,t_6$ may contain occurrences of X. To denote this, we will write $t_1(X),\ldots,t_6(X)$.

Now from Proposition 2.3 we have (using also the following fact which is easily derived from that Proposition: $(t(X))_n = (t((X)_n))_n$, $t \in EXP$):

$$\begin{aligned}(s(X))_n = \;& ((X_n \mathbin{\|\!\!\_}\, (t_1(X_{n-1}))_{n-1})\, (t_2(X_{n-1}))_{n-1} \mathbin{\|\!\!\_}\, (t_3(X_{n-1}))_{n-1} + \\ & (X_n \mathbin{\|\!\!\_}\, (t_4(X_{n-1}))_{n-1}) \mathbin{\|\!\!\_}\, (t_5(X_{n-1}))_{n-1} + \\ & a \mathbin{\|\!\!\_}\, (t_6(X_{n-1}))_{n-1}.\end{aligned}$$

(Here we saved some brackets by writing X_n instead of $(X)_n$.) By the induction hypothesis, the iteration sequence stabilizes modulo n - 1, say at $Q \in \mathbb{A}_{n-1}$. Hence for k sufficiently large we have, substituting $s^k(q)$ for X and Q for X_{n-1}:

$$\begin{aligned}(s(s^k(q)))_n = \;& (((s^k(q))_n \mathbin{\|\!\!\_}\, (t_1(Q))_{n-1})\, (t_2(Q))_{n-1}) \mathbin{\|\!\!\_}\, (t_3(Q))_{n-1} + \\ & ((s^k(q))_n \mathbin{\|\!\!\_}\, (t_4(Q))_{n-1}) \mathbin{\|\!\!\_}\, (t_5(Q))_{n-1} + \\ & a \mathbin{\|\!\!\_}\, (t_6(Q))_{n-1}.\end{aligned}$$

Let us write t_i' instead of $t_i(Q)$, i=1,...,6. So in order to prove stabilization modulo n of the iteration sequence generated by s(X) with starting value q, it suffices to prove stabilization modulo n of the iteration sequence generated by

$$s'(X) = ((X \mathbin{\|\!\!\_}\, t_1')t_2') \parallel t_3' + (X \mathbin{\|\!\!\_}\, t_4') \parallel t_5' + a \mathbin{\|\!\!\_}\, t_6',$$

with starting value $s^k(q) \equiv P$ for some k. The advantage obtained now is that the t_i' are closed terms, i.e. not containing X anymore. Write

$$\begin{aligned}T_1(X) &\equiv ((X \mathbin{\|\!\!\_}\, t_1')\, t_2') \mathbin{\|\!\!\_}\, t_3' \\ T_2(X) &\equiv (X \mathbin{\|\!\!\_}\, t_4') \mathbin{\|\!\!\_}\, t_5', \\ T_3 \; &\equiv a \mathbin{\|\!\!\_}\, t_6'.\end{aligned}$$

Then $s'(P) = T_1(P) + T_2(P) + T_3$, and

$$\begin{aligned}s'(s'(P)) &= T_1(T_1(P) + T_2(P) + T_3) + T_2(T_1(P) + T_2(P) + T_3) + T_3 \\ &= T_1(T_1(P)) + T_1(T_2(P)) + T_1(T_3) + \\ &\quad\; T_2(T_1(P)) + T_2(T_2(P)) + T_2(T_3) + T_3.\end{aligned}$$

Here the 'linearity' of T_1 and T_2 is due to the distributive laws for $\mathbin{\|\!\!\_}$ and $\cdot$ (A5, M4). Continuing in this way we find

$$s'^k(P) = \alpha_k + \beta_k + \beta_{k-1} + \ldots + \beta_2 + T_3$$

where

$$\alpha_k = \quad \Sigma_{i1,..,ik \in \{1,2\}} T_{i1}(T_{i\,2}(\ldots(T_{ik}(P))\ldots))$$

$$\beta_k = \quad \Sigma_{j1,..,j(k-1) \in \{1,2\}} T_{j1}(T_{j2}(\ldots(T_{j(k-1)}(T_3))\ldots)).$$

Now the summands α_k and β_k stabilize modulo n for growing k. For, consider α_k:

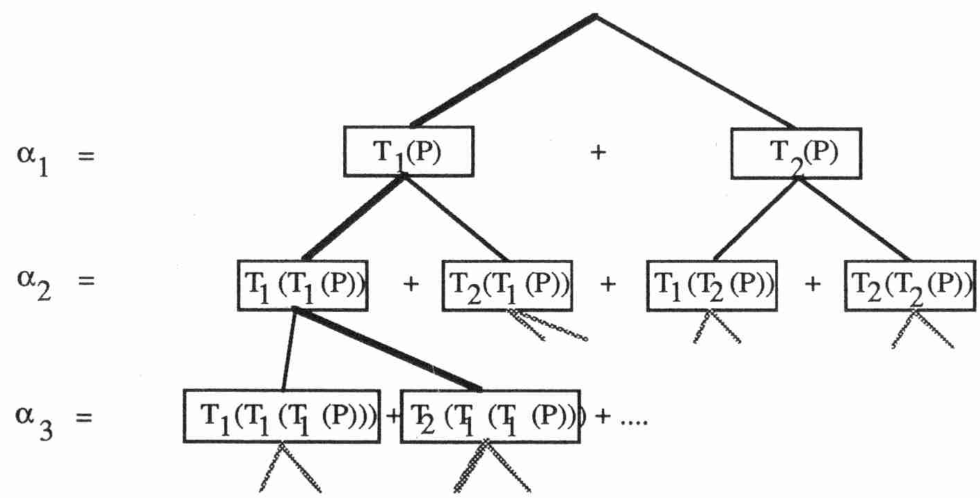


Figure 2

Each 'branch' in the tree thus obtained (see Figure 2), e.g. the indicated branch

$$T_1(P),\ T_1(T_1(P)),\ T_2(T_1(T_1(P))),\ T_1(T_2(T_1(T_1(P)))),\ \ldots$$

stabilizes modulo n, according to Proposition 3.6, since the operations T_1, T_2 consist of some left-merges on the right and some multiplications on the right. Hence, by König's Lemma, there is some k such that all branches are stabilized (modulo n) at that level k, i.e. for all summands $T_{i1}(T_{i2}(\ldots(T_{ik}(P))\ldots))$ in α_k further prefixing of T_1 or T_2 makes no difference modulo n. So from that k onwards, α_k is stable, modulo n. The same argument shows that β_k stabilizes modulo n for growing k. Therefore $s'^k(P)$ stabilizes modulo n for growing k, and this ends the proof.

(Note that the finiteness condition on A, necessary for the application of Proposition 3.6, is satisfied since the only $a \in A$ playing a role here, occur in q and s(X).) □

3.7.1. REMARK. Note that the theorem remains valid for an arbitrary $q \in \mathbb{A}^\infty$ as starting point for the iteration sequence: stabilization modulo n occurs as if the starting point was $(q)_n$.

3.8. COROLLARY. *Let* $s(X) \in$ EXP *contain no other variables than* X. *Then the equation* $X = s(X)$ *has a solution in the projection algebras* $\mathbb{A}_n$, *for every* $n \geq 1$; *and likewise in the projective model* $\mathbb{A}^\infty$.

PROOF. That $X = s(X)$ has a solution in $\mathbb{A}_n$, is an immediate consequence of Theorem 3.7: take an arbitrary atom a and iterate: a, s(a), $s^2(a)$, $s^3(a)$, ... until the sequence stabilizes modulo n:

$$(s^{k(n)}(a))_n = (s^{k(n)+1}(a))_n$$

for some k(n). Then $Q_n = (s^{k(n)}(a))_n$ is a solution in $\mathbb{A}_n$.

A solution in $\mathbb{A}^\infty$ is found by taking Q_n as above such that k(n) is a monotonic sequence; now $(Q_1, Q_2, ...)$ is a solution in $\mathbb{A}^\infty$. It is easy to verify (using the monotonicity of k(n)) that this is indeed a projective sequence. □

3.8.1. REMARK. In [Kr87] Corollary 3.8 has been generalized: in the equation $X = s(X)$ the RHS may contain parameters $p_1,...,p_m \in \mathbb{A}^\infty$. See [Kr87] also for several other generalizations.

3.9. Systems of recursion equations.

A natural question is whether the result in Theorem 3.7 can be generalized to *systems of recursion equations* $\{X_i = s_i(\mathbf{X}) \mid i = 1,...,n\}$ (here $s_i(\mathbf{X})$ is $s_i(X_1,...,X_n)$). The answer is no, if we take *parallel iterations* as in the following example:

	X = Ya	Y = Xb
X_1,Y_1	b	a
X_2,Y_2	aa	bb
X_3,Y_3	bba	aab
X_4,Y_4	aaba	bbab
...	...	...

Here (X_{n+1},Y_{n+1}) is computed by parallel substitution of the previous values (X_n,Y_n) in the RHSs of the recursion equations. Obviously, stabilization does not occur in the example. However, it seems that one can prove that if the iteration is not parallel, but *sequential* in the sense that in each step *only one* of the recursion variables is rewritten on the basis of the previous values, then stabilization occurs, modulo every $n \geq 1$; moreover, if the choice of the single recursion variable which is rewritten in the successive iteration steps is *fair*, then we find a solution of the system of recursion equations. See the following example of a fair, sequential iteration; alternatingly, the X and the Y is rewritten.

	X = Ya	Y = Xb
X_1,Y_1	b	a
X_2,Y_2	aa	a
X_3,Y_3	aa...	aab
X_4,Y_4	aaba	aab
...	...	...

We expect that a proof can be given of these statements along the same lines as above, for the case of a single recursion equation, but we will not attempt to do so here.

We further conjecture that parallel iterations of a system of n recursion equations, even though not 'converging' (in a sense to be made precise in the next section) to a fixed point or rather fixed vector of points, there still is a convergence: namely to a 'fixed cycle' consisting of n vectors. For the parallel iteration example above we have indeed:

$$(X_{2n}, Y_{2n}) \rightarrow (a(ab)^{\omega}, b(ba)^{\omega})$$
$$(X_{2n+1}, Y_{2n+1}) \rightarrow (b(ba)^{\omega}, a(ab)^{\omega}).$$

(Here $a(ab)^{\omega}$ stands for aabababab... , which in turn stands for the projective sequence (a, aa, aab, aaba, aabab, ...).)

4. The projective model as a complete metric space

The results above, stated in terms of 'stabilization modulo n', can be phrased in terms of 'convergence', as follows.

4.1. DEFINITION. Let $x,y \in \mathbb{A}^{\infty}$. Then the *distance* between $x = (x_1,x_2,...)$ and $y = (y_1,y_2,...)$, notation $d(x,y)$, is defined by:

$$d(x,y) = \begin{cases} 2^{-m} & \text{if } \exists n\ x_n \neq y_n;\ m = \min\{n \mid x_n \neq y_n\} \\ 0 & \text{otherwise, i.e. } \forall n\ x_n = y_n. \end{cases}$$

E.g. $d(a,b) = 2^{-1}$; $d(aabc, aa(a+b)c) = 2^{-3}$. (Here 'a' is short for the projective sequence (a,a,a,...) and aabc stands for (a,aa,aab,aabc,aabc,...).)

The following fact is easily established:

4.2. THEOREM. *The projective model* $\mathbb{A}^\infty$ *with the distance function* d *is a complete metric space.*

Now we can reformulate Theorem 3.7 (incorporating also Remark 3.7.1):

4.3. THEOREM. *Let* $q \in \mathbb{A}^\infty$ *and let* s(X) *be an expression in the signature of* PA *containing no other variables than* X. *Then the iteration sequence* q, s(q), $s^2(q)$, ... , $s^n(q)$, ... *converges in the metric space* ($\mathbb{A}^\infty$, d) *to a solution of the recursion equation* X = s(X).

Up to this point we have supposed that the alphabet A is finite. We will now show that this is not essential, and define projective models for arbitrary alphabets; furthermore we will connect these models with models obtained via process graphs and the notion of bisimulation equivalence or bisimilarity. It will be convenient to define first the latter class of models for PA.

4.4. DEFINITION. (i) A *process graph* is a rooted, directed, connected, edge-labeled graph. The edges (or arrows) are labeled with elements from the action alphabet A. The root is a designated node (the 'entrance' node, indicated by a small arrow as in Figure 1). Process graphs may have infinitely many nodes, or infinitely many edges (even between two nodes), and may contain cyclic 'paths'. Process graphs without cycles and without 'shared subgraphs' are process *trees*. (In [Mi80] these are called 'synchronisation trees'.) More precisely: a process graph is a process tree if every node has exactly one incoming arrow where the small root arrow also counts as an arrow. A process graph is finite if it contains finitely many edges and nodes.

(ii) If g is a process graph, and $s \in$ NODES(g) is a node of g, then the *branching degree* of s is the number of arrows leaving s. The branching degree of g is the maximum of the branching degrees of the nodes in g.

(iii) Two process graphs g,h with labels from the same alphabet are *bisimilar* if there is a *bisimulation* from g to h, that is a relation $R \subseteq$ NODES(g) x NODES(h) such that (1) the roots of g,h are related, (2) if $(s,t) \in R$ and $s \rightarrow_a s'$ is an edge in g, then there is an edge $t \rightarrow_a t'$ in h with $(s',t') \in R$, (3) likewise with the role of g,h interchanged. If g,h are bisimilar we write: g $\leftrightarrows$ h. (An example of two bisimilar graphs: the process graphs in Figure 1a and b.)

(iv) $\mathbb{G}_{\alpha,\beta}$ is the set of process graphs 'over' an alphabet of cardinality α and with branching degree $< \beta$. Here $\alpha \geq 1$ and $\beta \geq \aleph_0$. (The bound β on the branching degree must be infinite since otherwise the process graph domains below would not be closed under '+', defined below.) On $\mathbb{G}_{\alpha,\beta}$ we define operations +, $\cdot$, $\|$, $⌊⌊$,$(\,)_n$ ($n \geq 1$). For the precise definitions we refer to [BK86a,b]; for the sake of completeness we will give a short description. The *sum* graph g + h originates by identifying the roots of graphs g',h' obtained by unwinding g,h so far as necessary to make the roots acyclic. The *product* graph g·h is obtained by glueing copies of h at each end node of g. The *merge* g $\|$ h is the cartesian product of g and h (for an example see Figure 1a). The *left-merge* g ⌊⌊ h is like the merge but after removing all initial steps from g || h originating from h. The *projection* $(g)_n$ ($n \geq 1$) is defined for trees g: it is the tree obtained by cutting away all nodes reachable from the root by a path of length > n. The corresponding edges are also left away. If g is not a tree, then $(g)_n$ is defined as $(g')_n$ where g' is the tree obtained by unwinding g.

It turns out that bisimilarity ⇔ is a congruence w.r.t. the operations just defined. (The proof is routine.) Hence we can take the quotient

$$\mathbb{G}_{\alpha,\beta} = \mathbf{G}_{\alpha,\beta}/\Leftrightarrow.$$

The quotient structures are models of PA, i.e. process algebras for PA. Using the distance function analogous to the one in Definition 4.1 (with x_n replaced by the projection $(x)_n$), $\mathbb{G}_{\alpha,\beta}$ is a pseudo-metric space but not yet a metric space. (For instance, in $\mathbb{G}_{1,\aleph_1}$ the elements determined by the process graphs $\Sigma_{n\geq 1}\, a^n$ and $\Sigma_{n\geq 1}\, a^n + a^\omega$ are different but have distance 0.) It becomes a metric space after dividing out the congruence induced by the *Approximation Induction Principle* (AIP):

$$\frac{\forall n\ (x)_n = (y)_n}{x = y}$$

Note that the projective model does satisfy AIP. The result of dividing out AIP is

$$\mathbb{G}^\circ_{\alpha,\beta} = \mathbb{G}_{\alpha,\beta}/\mathrm{AIP}.$$

The $\mathbb{G}^\circ_{\alpha,\beta}$ have been defined as a 'double quotient' by first dividing out ⇔ and next AIP. The same result can be obtained by defining a suitable equivalence relation at once; this is done in [GR83] where 'weak equivalence' is divided out. In [Mi80], p.42 this notion is called 'observation equivalence'. It is defined as follows:

4.5. DEFINITION. (i) If $s \in$ NODES(g), then $(g)_s$ is the subgraph of g with root s, and nodes: all nodes in g reachable from s, and edges as induced by g.
(Warning: the notation $(g)_s$ should not be confused with $(g)_n$ for the n-th projection of g.)

(ii) On a process graph domain $\mathbf{G}_{\alpha,\beta}$ we define transition relations $\twoheadrightarrow_a$ for each atom a: if $s \rightarrow_a t$ is a step (edge) in $g \in \mathbf{G}_{\alpha,\beta}$, then $(g)_s \twoheadrightarrow_a (g)_t$.
(Note the difference in notation: open arrows stand for transitions between process graphs, normal arrows denote steps between nodes in one process graph.)

4.6. DEFINITION. On $\mathbf{G}_{\alpha,\beta}$ we define equivalences $\equiv_n$ for each $n \geq 0$:

(i) $g \equiv_0 h$ for all g,h;

(ii) $g \equiv_{n+1} h$ if

(1) whenever $g \rightarrow_a g'$ there is a transition $h \rightarrow_a h'$ with $g' \equiv_n h'$;

(2) as (1) with the roles of g,h interchanged.

Furthermore, $g \equiv h$ if $g \equiv_n h$ for all $n \geq 0$.

An alternative, equivalent definition is:

4.7. DEFINITION. Let $g,h \in \mathbb{G}_{\alpha,\beta}$ be process graphs. Then $g \equiv_n h$ if $(g)_n \leftrightarrow (h)_n$ $(n \geq 1)$. Furthermore, $g \equiv h$ if $g \equiv_n h$ for all $n \geq 1$.

The proof that these definitions are indeed equivalent is left to the reader. We also omit the routine proof of the next proposition, where $\cong$ denotes *isometry*.

4.8. PROPOSITION. $\mathbb{G}^\circ_{\alpha,\beta} \cong \mathbb{G}_{\alpha,\beta}/\equiv$. □

4.8.1. REMARK. For *finitely branching* graphs (i.e. $\beta = \aleph_0$) and arbitrary alphabet, we have in fact

$$\mathbb{G}_{\alpha,\aleph_0}/\equiv \ = \ \mathbb{G}_{\alpha,\aleph_0}/\leftrightarrow.$$

That is, weak equivalence (or observational equivalence) coincides with bisimulation equivalence. The proof (also in [BK86a]) is as follows: suppose g,h are finitely branching process graphs and suppose $g \equiv h$, or equivalently: $\forall n \ (g)_n \leftrightarrow (h)_n$. Now consider

$$B_n = \{R \mid R \text{ is a bisimulation from } (g)_n \text{ to } (h)_n\},$$
$$B = \cup_{n\geq 1} B_n.$$

This collection of 'partial' bisimulations between g,h is ordered by set-theoretic inclusion ($\subseteq$). In fact, $B' = B \cup \{(s_0,t_0)\}$ where s_0,t_0 are the roots of g,h respectively, is a tree w.r.t. $\subseteq$. Because g,h are finitely branching, this tree is also finitely branching: there are only finitely many extensions of a bisimulation between $(g)_n$, $(h)_n$ to a bisimulation between $(g)_{n+1}$, $(h)_{n+1}$. Moreover, because $\forall n \ (g)_n \leftrightarrow (h)_n$, the tree B' has infinitely many nodes. Therefore, by König's Lemma, B' has an infinite branch. This infinite branch is a chain of partial bisimulations R_i $(i\geq 1)$:

$$R_1 \subseteq R_2 \subseteq \ldots \subseteq R_n \subseteq \ldots$$

such that R_i is a bisimulation from $(g)_i$ to $(h)_i$. Now $R = \cup_{n\geq 1} R_i$ is a bisimulation from g to h.

The structures $\mathbb{G}^\circ_{\alpha,\beta}$ are also process algebras for PA. While all of the $\mathbb{G}^\circ_{\alpha,\beta}$ are metric spaces, they are not all *complete*. An example is given in [GR83]: $\mathbb{G}^\circ_{1,\aleph_0}$ is incomplete. (Consider the approximations of $\sum_{n\geq 1} a^n$.) Another example is as follows.

4.9. EXAMPLE. $\mathbb{G}^\circ_{\aleph_\omega,\aleph_\omega}$ is an incomplete metric space.

PROOF (sketch). The alphabet is $\{a_i \mid i < \aleph_\omega\}$. Define a sequence of process graphs g_n ($n \geq 1$) by

$$g_n = \Sigma_{i1<\aleph 1}\, \Sigma_{i2<\aleph 2} \cdots \Sigma_{in<\aleph n}\, a_{i1}a_{i2}\cdots a_{in}.$$

Let brd(g) be the branching degree of process graph g, defined as follows: if s is a node of g, then brd(s) is the (cardinal) number of arrows leaving s; furthermore, brd(g) is the cardinal sum of the brd(s), $s \in$ NODES(g). We claim:

(i) $\mathrm{brd}(g_n) = \aleph_n$ for g_n as defined above,
(ii) $\mathrm{brd}((g)_n) \leq \mathrm{brd}(g)$ for all $g \in \mathbb{G}_{\alpha,\beta}$,
(iii) $h \leftrightarrows g_n \Rightarrow \mathrm{brd}(h) \geq \mathrm{brd}(g_n)$ for g_n as defined above.

Claim (ii) is trivial; the inductive proofs of the other two claims are left to the reader. Using these claims, one shows immediately that there is no limit $g/\equiv$ for the sequence of elements $g_n/\equiv$ in $\mathbb{G}^\circ_{\aleph_\omega,\aleph_\omega}$ as this would require a process graph g with branching degree at least $\Sigma_{n<\omega}\, \aleph_n = \aleph_\omega$. □

We will now define *projective models* $\mathbb{A}^\infty_{\alpha,\beta}$ of PA for arbitrary $\alpha \geq 1$ and $\beta \geq \aleph_0$. These will all be complete metric spaces. Furthermore, modulo isometry $\mathbb{A}^\infty_{\alpha,\beta}$ is an extension of $\mathbb{G}^\circ_{\alpha,\beta}$, so the projective model can be considered as the metric completion of $\mathbb{G}^\circ_{\alpha,\beta}$. (In case $\mathbb{G}^\circ_{\alpha,\beta}$ is also complete, it is of course isometric to the projective model.) The projective models defined below differ from the ones in [Kr87]; there an element of a projective sequence is a sequence of *terms* (modulo derivable equality), below it is a sequence of finitely deep *process graphs* (modulo bisimilarity).

4.10. DEFINITION. (i) $\mathbb{G}^n_{\alpha,\beta} = \{g \in \mathbb{G}_{\alpha,\beta} \mid g = (g)_n\}$.

(ii) $\mathbb{G}^n_{\alpha,\beta} = \mathbb{G}^n_{\alpha,\beta}/\leftrightarrows$. Note that $\mathbb{G}^n_{\alpha,\beta}$ is a process algebra for PA, with definitions of the operators analogous to the one in Remark 2.4. It is a routine exercise to prove that the process algebras $\mathbb{A}_n$ as in Remark 2.4, are isomorphic to $\mathbb{G}^n_{\alpha,\aleph_0}$ where α is finite.

(iii) Let $g_i \in \mathbb{G}_{\alpha,\beta}$ ($i \geq 1$). Then the sequence $(g_1, g_2, \ldots)$ is *projective* if for all i: $g_i = (g_{i+1})_i$.

(iv) $\mathbb{A}^\infty_{\alpha,\beta}$ is the projective limit of the $\mathbb{G}^n_{\alpha,\beta}$ ($n \geq 1$); the elements of $\mathbb{A}^\infty_{\alpha,\beta}$ are the projective sequences. So the projective model $\mathbb{A}^\infty$ of sections 1-3 is the same as $\mathbb{A}^\infty_{\alpha,\aleph_0}$ (α finite).

4.11. THEOREM. (i) *$\mathbb{A}^\infty_{\alpha,\beta}$ is a complete metric space.*
(ii) *The convergence theorem 4.3 also holds for $\mathbb{A}^\infty_{\alpha,\beta}$.*

PROOF (sketch). (i) Consider a converging sequence $\gamma_i = (g_{i1}, g_{i2}, ...)$, $i \geq 1$. For growing i and fixed k, the sequence g_{ik} will eventually be constant, say after N(k) steps. We may suppose that N is a monotonic function. Now $\gamma = (g_{N(1),1}, g_{N(2),2}, ...)$ is the required limit.

(ii) Directly from the proof in Section 3 (Theorem 3.7). □

Van Glabbeek (personal communication) remarked that for finite α, there is no need to consider uncountably branching process graphs, see statement (i) in Corollary 4.15. His observation can be generalized to infinite α. First some notation.

4.12. NOTATION. Let α be a cardinal number (finite or infinite). Then $\alpha^* = \sum_{n<\omega} \alpha_n$, where $\alpha_0 = \alpha$, $\alpha_{n+1} = 2^{\alpha_n}$. For finite α, we have $\alpha^* = \aleph_0$. For $\alpha = \aleph_0$, the numbers α_n are known as the beth-numbers $\beth_n$ and $\alpha^* = \beth_\omega$. The cardinality of a set X is card(X). If κ is a cardinal, then κ^+ denotes the least cardinal larger than κ.

4.13. PROPOSITION. (i) *For infinite* α: $\mathrm{card}(\mathbb{G}^n_{\alpha,\alpha^*}) = \alpha_n$.

(ii) $\mathrm{card}(\bigcup_{n\geq 1} \mathbb{G}^n_{\alpha,\alpha^*}) = \alpha^*$.

(iii) *For any* α,κ: $\mathbb{G}^n_{\alpha,\alpha^*} \cong \mathbb{G}^n_{\alpha,\alpha^*+\kappa}$.

PROOF. (i) Induction on n. For n = 1 the statement is clear, since the process graphs $g_I = \sum_{a\in I} a$ for arbitrary non-empty $I \subseteq A$ are mutually non-bisimilar, and since every process graph in $\mathbb{G}^1_{\alpha,\alpha^*}$ is bisimilar with some g_I. Suppose the statement has been proved for n. Let $\mathcal{X}^n \subseteq \mathbb{G}^n_{\alpha,\alpha^*}$ be a set of representatives of the α_n bisimulation equivalence classes of $\mathbb{G}^{n+1}_{\alpha,\alpha^*}$, so $\mathrm{card}(\mathcal{X}^n) = \alpha_n$. Now every element of $\mathbb{G}^{n+1}_{\alpha,\alpha^*}$ is bisimilar to one of the process graphs

$$g_{h,I,f} = h + \sum_{a\in I} \sum_{x\in f(I)} ax$$

where $h \in \mathcal{X}^n$, $I \subseteq A$ (possibly empty) and $f: I \to \wp(\mathcal{X}^n)$. Moreover, for different triples h,I,f the corresponding $g_{h,I,f}$ are not bisimilar. Hence $\mathrm{card}(\mathbb{G}^n_{\alpha,\alpha^*}) = \alpha_n \cdot \alpha_1 \cdot \alpha_{n+1} = \alpha_{n+1}$. Here the factor α_n stems from the variation in h, α_1 from the variation in I while for each I the choice of f contributes a factor $(2^{\mathrm{card}(\mathcal{X}n)})^{\mathrm{card}(I)} = 2^{\alpha n} = \alpha_{n+1}$.

Part (ii) is by definition; (iii) is left to the reader. □

4.14. THEOREM. $\mathbb{A}^\infty_{\alpha,\alpha^*} \cong \mathbb{A}^\infty_{\alpha,\alpha^*+\kappa}$ *for any cardinal* κ.

PROOF. The isometry follows at once from Proposition 4.13(iii). □

4.15. COROLLARY.

(i) *For finite* α: $\mathbb{A}^\infty_{\alpha,\aleph 0} \cong \mathbb{A}^\infty_{\alpha,\aleph 0+\kappa}$ *for any cardinal* κ.

(ii) *For countably infinite alphabet:* $\mathbb{A}^\infty_{\aleph 0,\beth\omega} \cong \mathbb{A}^\infty_{\aleph 0,\beth\omega+\kappa}$ *for any cardinal* κ. □

We will now turn our attention to the models $\mathbb{G}^\circ_{\alpha,\beta}$ in order to compare them with the projective models.

4.16. PROPOSITION. *If* β *is sufficiently large,* $\mathbb{G}^\circ_{\alpha,\beta}$ *is complete.*

PROOF. We will try to prove that $\mathbb{G}^\circ_{\alpha,\beta}$ is isometric to $\mathbb{A}^\infty_{\alpha,\beta}$ and deduce from that attempt a requirement on β.

We will drop the subscripts α,β. So let us try to establish an isometry φ from $\mathbb{G}^\circ$ to $\mathbb{A}^\infty$. Let $g \in \mathbb{G}^\circ$. Then $\varphi(g) = ((g)_1,(g)_2,...)$. It is easy to prove that this is a projective sequence. The hard part is to prove that φ is a surjection. Consider an element $(g_1, g_2, ...) \in \mathbb{A}^\infty$. Let g_i be a representing process graph of g_i ($i \geq 1$). We would like to find a graph g such that $(g)_i \leftrightarrow g_i$ for all $i \geq 1$. (Cf. the construction in Theorem 3.5 of [GR83] by 'blowing up' trees; we will use another construction.) For the rest of this proof, we will suppose that all process graphs are trees. Let g_i' be $(g_{i+1})_i$. So $g_i \leftrightarrow g_i'$; say R_i is a bisimulation from g_i to g_i'. Let S_i: NODES$(g_i') \rightarrow$ NODES(g_{i+1}) be the obvious embedding function, obtained by the projection mapping. Now if s is a node of depth k in g_k (so s is 'appearing' for the first time in g_k), we define some sequences starting with s, called *fibres*, as follows. Any sequence

$$s = s_k, s_k', s_{k+1}, s_{k+1}', s_{k+2}, s_{k+2}', ...$$

where $s_i \in$ NODES(g_i), $s_i' \in$ NODES(g_i'), $(s_i,s_i') \in R_i$ and $S_i(s_i') = s_{i+1}$ ($i \geq k$) is a fibre. We will say that this fibre starts in g_k. If σ,τ are fibres, starting in g_k and g_{k+1} respectively, we define transitions $\sigma \rightarrow_a \tau$ if there are a-steps between the elements of these sequences:

$$\begin{array}{llllll}
\sigma: & s_k, s_k', & s_{k+1}, & s_{k+1}', & s_{k+2}, & s_{k+2}', ... \\
\downarrow a & & \downarrow a & \downarrow a & \downarrow a & \downarrow a \\
\tau: & & t_{k+1}, & t_{k+1}', & t_{k+2}, & t_{k+2}', ...
\end{array}$$

Now we construct the process graph γ with as nodes the fibres and transitions as just defined. More precisely: the root of γ is the fibre through the roots of $g_1, g_1', g_2, ...$, and the other nodes of γ are those fibres reachable from the root of γ via transitions between fibres.

(Comment: Not all fibres need to be reachable from the root fibre. However, if the bisimulations R_i are taken *minimal* in the set-theoretic sense, then all fibres are reachable from the root fibre. This can be proved with induction on the depth of the fibres, using the following proposition:

Let g,g' *be process trees, and let* R *be a minimal bisimulation from* g *to* g'. *Let* $s \rightarrow_a s'$ *be a step in* g *and suppose* $(s',t') \in R$. *Then there is a node* t *and a step* $t \rightarrow_a t'$ *in* g' *such that* $(s,t) \in R$.)

We claim that the projection $(\gamma)_n$ is bisimilar to g_n. A bisimulation ρ_n is given as follows: if $s \in$ NODES(g_n) and $\sigma \in$ NODES$((\gamma)_n)$ then $(s,\sigma) \in \rho_n$ iff s is an element of σ. The verification of the claim is easy. An illustration is given in Figure 2 where γ is 'reconstructed' from the sequence of process graphs a, $a+a^2$, $a+a^2+a^3$, ... Interestingly, the result is not $\Sigma_{n\geq1}\ a^n$ but $\Sigma_{n\geq1}\ a^n + a^\omega$. (See the 'black fibers' in Figure 3.)

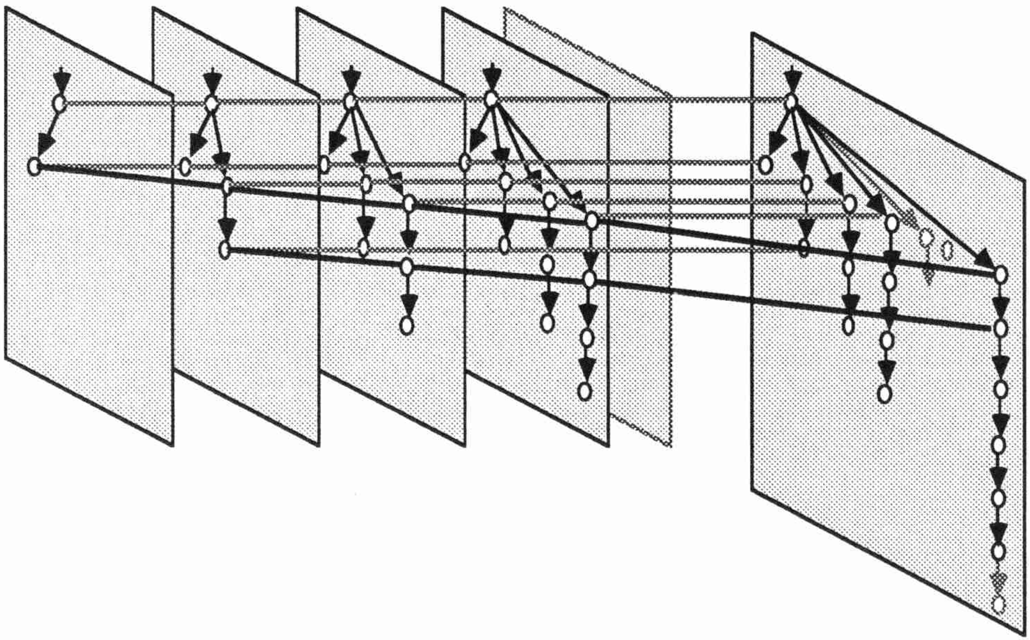


Figure 3

However, the problem is now to prove that the branching degree of γ is strictly bounded by β. We claim that this is so if $\beta > (\alpha^*)^{\aleph_0}$. Proof of the claim: let us take the g_i $(i \geq 1)$ above as small as possible w.r.t. the cardinalities of their node sets. From the proof of Proposition 4.13(i) it is clear that we can take the g_i such that card(NODES(g_i)) $\leq \alpha_i$ (in fact we can even take card(NODES(g_i)) $\leq \alpha_{i-1}$). Hence we may suppose that the union of the node sets of the g_i, g_i' $(i\geq1)$ is bounded by α^*. Now every fibre (a node of the tree γ) is an ω-sequence of nodes of the g_i, g_i'. Hence there are at most $\kappa = (\alpha^*)^{\aleph_0}$ such fibres; so γ has at most κ nodes, so the branching degree of γ is bounded by κ. □

4.17. REMARK. (i) In the example above, in Figure 3, the process graph γ is closed (see Definition 5.1 for the definition of 'closed process graph'). In general, this needs not to be the case: e.g. if in the proof of Proposition 4.16, $g_i = (\Sigma_{n\geq1}\ a^n)_i$ for $i \geq 1$ (so g_1 consists of infinitely many a-steps attached at the root) then $\gamma = \Sigma_{n\geq1}\ a^n$ and this graph is not closed.

(ii) Another way of constructing a process graph g with projections $(g)_n$ bisimilar to g_n as in the proof above, is by taking g as the *canonical process graph* of the projective sequence $(g_1, g_2, ...) \in \mathbb{A}^\infty$. See Definition 5.2. One can prove that this graph is closed indeed, for $\beta > (\alpha^*)^{\aleph_0}$.

4.18. DEFINITION. Let $X, X' \subseteq \mathbb{G}_{\alpha,\beta}$. (i) Then $(X)_n = \{(g)_n \mid g \in X\}$.

(ii) $X \equiv_n X'$ if $\forall g \in X\ \exists g' \in X'\ \ g \equiv_n g'$ and $\forall g' \in X'\ \exists g \in X\ \ g \equiv_n g'$.

(iii) $X \equiv X'$ if $X \equiv_n X'$ for all n.

4.19. DEFINITION. Let $g \in \mathfrak{G}_{\alpha,\beta}$. The a-*derivation* of g is the set of all subgraphs of g reachable by an a-step from the root. Notation: g/a.

4.20. PROPOSITION. *Let* $g,h \in \mathfrak{G}_{\alpha,\beta}$. *Then* g,h *determine the same element in* $\mathbb{G}^\circ_{\alpha,\beta}$ *iff for all* a, $g/a \equiv h/a$.

PROOF. Routine. □

4.21. PROPOSITION. *Let* $X \subseteq \mathfrak{G}_{\alpha,\beta}$. *Then there is an* $X' \subseteq \mathfrak{G}_{\alpha,\beta}$ *such that* $X \equiv X'$ *and* $\text{card}(X') \leq \alpha^*$.

PROOF. Consider the collection $\bigcup_{n\geq1}(X)_n$ of finitely deep process graphs. We will construct a graph (not a process graph) with node set $\bigcup_{n\geq1}(X)_n$, and arrows $g \to h$ for $g \in (X)_n$, $h \in (X)_{n+1}$ whenever $g = (h)_n$. See Figure 4.

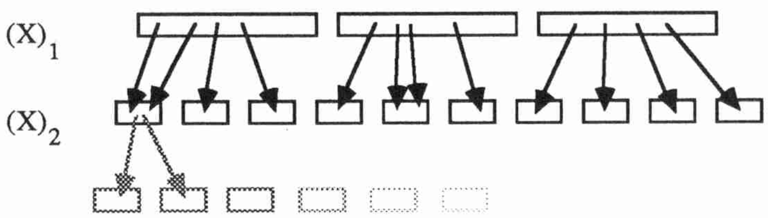


Figure 4

The boxes in Figure 4 are the $\underline{\leftrightarrow}$-equivalence classes. We note (Proposition 4.13(i)) that there are at most α_n boxes at level n, hence at most α^* boxes in total. Now every $g \in X$ corresponds with a path in this huge graph (not necessarily vice versa). We now construct X' as follows. If $g \in X$ is finitely deep (i.e. determines a terminating path in the graph of Figure 2), then $g \in X'$. Furthermore, in each box we select one node (i.e. a process graph $g \in (X)_n$ for some n) and choose an *arbitrary* path through this node. This path (which in fact is a projective sequence of process *graphs*) determines a process graph, call it $g^\sim$. Now we put $g^\sim \in X'$. Obviously, $\text{card}(X') \leq \alpha^*$ and it is not hard to prove that $X \equiv X'$. □

4.22. PROPOSITION. *For all* α,κ: $\mathbb{G}^\circ_{\alpha,(\alpha^*)^+} \cong \mathbb{G}^\circ_{\alpha,(\alpha^*)^+ +\kappa}$

PROOF. Consider a process graph $g \in \mathfrak{G}_{\alpha,(\alpha^*)^+ +\kappa}$. We must show that g can be pruned to a $g' \in \mathfrak{G}_{\alpha,(\alpha^*)^+}$ such that g and g' determine the same element after dividing out $\underline{\leftrightarrow}$ and AIP (or dividing out $\equiv$ at once). This follows directly from the preceding two propositions. □

4.23. COROLLARY. *For all* α,κ,λ: $\mathbb{A}^{\infty}{}_{\alpha,\alpha^*+\kappa} \cong \mathbb{G}^{\circ}{}_{\alpha,(\alpha^*)^+ + \lambda}$

PROOF. This follows from Theorem 4.14 and Propositions 4.16 and 4.22. □

The cardinality of the models constructed above is for infinite alphabets quite large (this was already noticed by Golson and Rounds in [GR83] for the process model of [BZ82a,b]; see our remarks below). In fact:

4.24. PROPOSITION. (i) *For finite* α: card($\mathbb{A}^{\infty}{}_{\alpha,\aleph_0}$) = $2^{\aleph_0}$

(ii) *For countably infinite alphabet*: card($\mathbb{A}^{\infty}{}_{\aleph_0,\beth_\omega}$) = $\beth_{\omega+1}$

(iii) *For general* α: card($\mathbb{A}^{\infty}{}_{\alpha,\alpha^*}$) = $2^{(\alpha^*)} = (\alpha^*)^{\aleph_0}$.

PROOF. (We will assume the Axiom of Choice in our calculations with cardinals.) Statements (i) and (ii) follow from (iii). Proof of (iii): Let λ be card($\mathbb{A}^{\infty}{}_{\alpha,\alpha^*}$) . Using Proposition 4.13 and noting that every element of $\mathbb{A}^{\infty}{}_{\alpha,\alpha^*}$ is a map from ω into the union of the $\mathbb{G}^{n}{}_{\alpha,\alpha^*}$, we have $\lambda \le (\alpha^*)^{\aleph_0}$. In view of the isomorphism with the graph models (Corollary 4.23), we find $\lambda \ge 2^{(\alpha^*)}$. The argument is as follows: there are α^* finitely deep process graphs which are mutually not bisimilar. (This is in fact Proposition 4.13(ii).) Let $\mathcal{F}$ be the set of these process graphs. For every subset $\mathcal{X}$ of $\mathcal{F}$ we define a process graph $g_{\mathcal{X}}$ as $\Sigma_{g\in\mathcal{X}}$ a.g for a fixed atom a. Now $g_{\mathcal{X}} \leftrightarrows g_{\mathcal{Y}}$ iff $\mathcal{X} = \mathcal{Y}$. Moreover, for different $\mathcal{X},\mathcal{Y}$ the corresponding graphs are not identified after dividing out AIP. So we have now:

$$2^{(\alpha^*)} \le \lambda \le (\alpha^*)^{\aleph_0}.$$

We also have: $2^{(\alpha^*)} = (\alpha^*)^{\alpha^*} \ge (\alpha^*)^{\aleph_0}$ (here AC is used, in the equality step). Hence the result follows. □

4.25. QUESTIONS. At present we do not know the answers to the following questions. For what α,β is $\mathbb{G}^{\circ}{}_{\alpha,\beta}$ a complete metric space? What is the cardinality of $\mathbb{G}^{\circ}{}_{\alpha,\beta}$ and $\mathbb{A}^{\infty}{}_{\alpha,\beta}$? If $\mathbb{G}^{\circ}{}_{\alpha,\beta}$ is a complete metric space, is $\mathbb{G}^{\circ}{}_{\alpha,\beta'}$ for $\beta' > \beta$ also complete?

It is interesting to compare the projective model $\mathbb{A}^{\infty}{}_{\alpha,\alpha^*}$ with the process model $\mathbb{P}_\alpha$ as constructed by De Bakker and Zucker [BZ82a,b] as a solution of the domain equation

$$P \cong \{p_0\} \cup \wp_c(A \times P).$$

In $\mathbb{P}_\alpha$, processes can terminate with p_0 or with $\varnothing$ ('successfully' or 'unsuccesfully'). Leaving this double termination possibility aside (one can extend PA to PA_δ and have the same double termination possibility, see [BK86a,b]) or using a variant of the domain equation:

$$P \cong \wp_c(A \cup (A \times P)),$$

we can state that our projective model $\mathbb{A}^{\infty}_{\alpha,\alpha^*}$ is *isometric* to the process domain $\mathbb{P}_{\alpha}$. For finite α, this follows from the proof in [GR83] that $\mathbb{P}_{\alpha}$ is isometric to the graph domain $\mathbb{G}^{\circ}_{\alpha,\aleph_1}$; hence it is also isometric to $\mathbb{A}^{\infty}_{\alpha,\aleph_0}$, by Corollary 4.23. For infinite α the proof is similar. (The proof proceeds by noting that our spaces of finitely deep processes $\mathbb{G}^{n}_{\alpha,\alpha^*}$ are isometric to the P_n in [BZ82a,b] or [GR83]; hence the completions of $\bigcup_{n\geq 1} \mathbb{G}^{n}_{\alpha,\alpha^*}$ and $\bigcup_{n\geq 1} P_n$, respectively, must also be isometric.) So the cardinality statements in Proposition 4.24 apply also to the models in [BZ82a,b], and our convergence theorem is also valid in these models.

For a systematic (category-theoretic) treatment of De Bakker-Zucker domain equations like the two above, showing that they have unique solutions modulo isometry, we refer to [AR87].

5. Closed process graphs

We conclude with some remarks about a trade-off between closure properties of processes and the Approximation Induction Principle used in the construction of $\mathbb{G}^{\circ}_{\alpha,\alpha^*}$. These remarks are suggested by the fact that the model of De Bakker and Zucker is a solution of their domain equation; loosely speaking this means that the elements of that model can be perceived as 'hereditarily closed sets'. (Note, however, that these 'sets' are not well-founded; it would be interesting to give a representation of the solution of the domain equation above in terms of a set theory without the Axiom of Foundation.) One may ask whether the closure property can replace, when constructing a model from process graphs such as $\mathbb{G}^{\circ}_{\alpha,\alpha^*}$, taking the quotient w.r.t. AIP. We will make this question more precise using the definition of 'closed process tree' which was suggested to us by R. van Glabbeek (personal communication).

5.1. DEFINITION. (i) For process trees $g,h \in \mathfrak{G}_{\alpha,\beta}$ we define the distance $\delta(g,h)$ as follows:

$$\delta(g,h) = \begin{cases} 2^{-m} & \text{if } \exists n \;\; g \not\equiv_n h;\;\; m = \min\{n \mid g \not\equiv_n h\} \\ 0 & \text{otherwise, i.e. } g \equiv h. \end{cases}$$

(ii) Let $\mathcal{H} \subseteq \mathfrak{G}_{\alpha,\beta}$ be a set of process trees. Then $\mathcal{H}$ is *closed* if every Cauchy sequence $(g_i)_{i\geq 1}$ w.r.t. δ in $\mathcal{H}$ converges to a limit g in $\mathcal{H}$ (i.e. $\forall k\, \exists N\, \forall n{>}N\;\; g \equiv_k g_n$).

(iii) Let $g \in \mathfrak{G}_{\alpha,\beta}$ be a process tree. Then g is *closed* if all its nodes s are closed; and a node s in g is closed when $(g)_s/a$ is a closed set of trees for every $a \in A$. Here $(g)_s$ is the subtree of g at s. Futhermore, a process graph is closed if its tree unwinding is closed. The set of all closed process graphs is $\mathfrak{G}^{c}_{\alpha,\beta}$.

5.1.1. REMARK. Note that the closure property of process graphs is invariant under bisimulation equivalence: if g $\leftrightarrows$ h and g is closed, then h is closed.

5.2. DEFINITION. Let $\mathbb{M}$ be a a process algebra for PA.

(i) From the elements x,y,z,... of $\mathbb{M}$ we construct a transition diagram (i.e. a 'process graph' without root and not necessarily connected) as follows. Whenever x = ay + z there is a transition $x \rightarrow_a y$. In the case that x = ay we have the same transition. If x = a, then there is a transition $x \rightarrow_a o$ where o is the *termination node*. More concisely, we have $x \rightarrow_a y$ iff x = ay + x and $x \rightarrow_a o$ iff x = a + x. (To see this, use the axiom x + x = x.)

(ii) The *canonical process graph of* x *in* $\mathbb{M}$ is the process graph with root x, and as nodes all the elements of $\mathbb{M}$ reachable from x in zero or more transition steps as just defined, including possibly the termination node. Notation: $can_{\mathbb{M}}(x)$ or just can(x) when it is clear what $\mathbb{M}$ is meant. (See Figure 5 for the canonical process graph of $(\sum_{n\geq 1} a^n)/\equiv$ in $\mathbb{G}^\circ_{\alpha,\beta}$.)

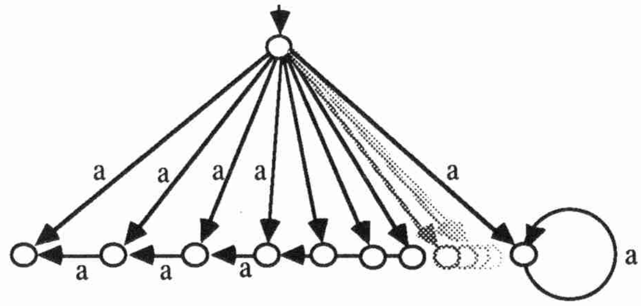


Figure 5

5.3. PROPOSITION. *Let* g/≡ *be an element of* $\mathbb{G}^\circ_{\alpha,\beta}$. *Then:*

(i) can(g/≡) ≡ g.

(ii) can(g/≡) $\equiv_n$ can(h/≡) $\Leftrightarrow$ g $\equiv_n$ h.

(iii) can(g/≡) *is a closed proces graph.*

PROOF. (i) With induction on n we prove that g $\equiv_n$ can(g/≡) for $n \geq 0$ (see Definition 4.6). The basis of the induction, n = 0, is trivial. Suppose (induction hypothesis) that we have proved $\forall g$ g $\equiv_n$ can(g/≡). In order to prove g $\equiv_{n+1}$ can(g/≡), we have to show (1) and (2):

(1) *for every transition* g $\rightarrow_a$ g' *there is an initial step in* can(g/≡): g/≡ $\rightarrow_a$ h/≡ *such that*

$$g' \equiv_n (can(g/\equiv))_{(h/\equiv)} = can(h/\equiv).$$

(Remember that g/≡, h/≡ are nodes in can(g/≡).) Now g/≡ $\rightarrow_a$ h/≡ is (by definition of canonical process graph) the same as: $g/\equiv\ =\ a(h/\equiv) + r/\equiv$ for some graph r. Or, equivalently: $g \equiv ah + r$. So, given the transition $g \twoheadrightarrow_a g'$ we have to find h,r with $g \equiv ah + r$ and $g' \equiv_n\ can(h/\equiv)$. This is simple: take $h = g'$ and r as given by $g \twoheadrightarrow_a g'$ (i.e. $g = a\cdot g' + r$ for some r). Now apply the induction hypothesis.

(2) *For every initial step in* $can(g/\equiv)$: $g/\equiv\ \to_a h/\equiv$ *there is a transition* $g \twoheadrightarrow_a g'$ *such that* $g' \equiv_n\ can(h/\equiv)$.

So, let $g/\equiv\ \to_a h/\equiv$ be given. This means $g \equiv ah + r$ for some r. In particular, $g \equiv_{n+1} ah + r$, i.e.

$$(g)_{n+1} \leftrightarrows (ah + r)_{n+1} = a(h)_n + (r)_{n+1}. \qquad (*)$$

From the induction hypothesis we know that $h \equiv_n can(h/\equiv)$, i.e.

$$(h)_n \leftrightarrows (can(h/\equiv))_n. \qquad (**)$$

Combining (*),(**) we have

$$(g)_{n+1} \leftrightarrows a(can(h/\equiv))_n + (r)_n. \qquad (***)$$

Now we have to find a step $g \to_a g'$ such that $g' \equiv_n can(h/\equiv)$, i.e. $(g')_n \leftrightarrows (can(h/\equiv))_n$. This is easily obtained from (***): consider the a-occurrence displayed in the RHS of (***). By definition of $\leftrightarrows$, this a-step is matched in $(g)_{n+1}$ by an a-step $(g)_{n+1} \to_a (g')_n$ with $(g')_n \leftrightarrows (can(h/\equiv))_n$.

(ii) Write $g^* = can(g/\equiv)$. To prove ($\Leftarrow$), suppose $g \equiv_n h$. Then $g^* \equiv g \equiv_n h \equiv h^*$, using (i). So $g^* \equiv_n h^*$. The proof of ($\Rightarrow$) is similar.

(iii) Consider $can(g/\equiv)$. (See Figure 5.) Let s be a node of this graph (so $s \in \mathbb{G}^\circ_{\alpha,\beta}$). Consider the a-derivation of s, i.e. the set of subgraphs of $can(g/\equiv)$ determined by the a-successors of s. Clearly, this a-derivation is the set of canonical graphs of some elements t_i ($i \in I$) of $\mathbb{G}^\circ_{\alpha,\beta}$. Suppose this set $\{can(t_i) \mid i \in I\}$ contains a Cauchy sequence (w.r.t. δ as in Definition 5.1):

$$can(t_{i0}),\ can(t_{i1}),\ \dots,\ can(t_{in}),\ \dots .$$

We claim that the elements $t_{i0}, t_{i1}, \dots, t_{in}, \dots$ form a Cauchy sequence in $\mathbb{G}^\circ_{\alpha,\beta}$. This follows at once from (ii) of this proposition. So there is a limit $t \in \mathbb{G}^\circ_{\alpha,\beta}$ of the last Cauchy sequence. Now can(t) is easily seen (using again (ii)) to be a limit (in the δ-sense) for the Cauchy sequence $can(t_{i0})$, $can(t_{i1})$, ...

We still have to prove that $s \to_a t$, or equivalently (see Definition 5.2(i)) $s = at + s$ in $\mathbb{G}^\circ_{\alpha,\beta}$. Let $\underline{s}$ denote a representing process graph from the $\equiv$-equivalence class s, and likewise for t etc.

Then we must prove that $\underline{s} \equiv a\underline{t} + \underline{s}$. To this end, take $\underline{t}_{ik}$ such that $\underline{t}_{ik} \equiv_n \underline{t}$. Since $\underline{s} \equiv a\underline{t}_{ik} + \underline{s}$ we have $\underline{s} \equiv_n a\underline{t} + \underline{s}$. Hence $\underline{s} \equiv a\underline{t} + \underline{s}$. □

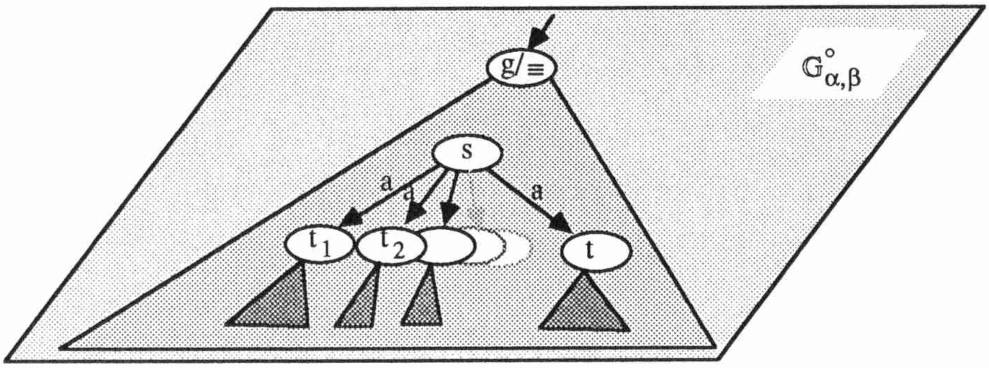


Figure 5

The preceding proposition enables us to define the *closure* of a process graph $g \in \mathbb{G}_{\alpha,\beta}$, notation g^c, as can(g/≡) w.r.t. $\mathbb{G}^{\circ}_{\alpha,\beta}$, such that $g \equiv g^c$. Next, we define operations $+^c$, $\cdot^c$, $||^c$, $\lfloor\!\lfloor^c$ on $\mathbb{G}^c_{\alpha,\beta}$ as follows: $g \,||^c\, h = (g \,||\, h)^c$ and likewise for the other operators. Here $||$ is the merge operation on $\mathbb{G}_{\alpha,\beta}$.

5.4. REMARK. If $\mathbb{G}^c_{\alpha,\beta}$ would have been closed under the operations $+$, $\cdot$, $||$, $\lfloor\!\lfloor$ the preceding closure operation in $(g \,||\, h)^c$ (etc.) would not have been necessary. However, for an *infinite* alphabet $\mathbb{G}^c_{\alpha,\beta}$ is not necessarily closed under $||$, as the following example shows. (We conjecture that for finite alphabets $\mathbb{G}^c_{\alpha,\beta}$ is closed under the operations $||$ etc.)

Let the alphabet be $\{a_i \mid i \geq 1\} \cup \{b,c\}$. We define process graphs H, G, g_n $(n \geq 1)$:

$$H = ||_{i\geq 1}\, a_i$$

$$g_n = a_n \,||\, b^n$$

$$G = \Sigma_{n\geq 1}\, c\cdot g_n.$$

Here H is the infinite merge of all atoms a_i $(i \geq 1)$. Alternatively, H can be defined as having as nodes all finite subsets of $\mathbb{N}^+$ (the set of positive natural numbers), as root Ø, and as edges:

$$V \rightarrow_{ai} V \cup \{i\}$$

for all $V \subseteq \mathbb{N}^+$ and $i \notin V$. Now H is a closed process graph. This can be easily seen, noting that H is a *deterministic* process graph, i.e. a graph where two different edges leaving the same node must have different label, and noting that deterministic graphs are always closed. Also G is closed:

the c-derivation G/c, consisting of the graphs g_n, does not contain a Cauchy sequence since the graphs g_n are already different in their first level, due to the 'spoiling effect' of the a_n in g_n. Now G || H is, we claim, not closed. For, consider the c-derivation

$$(G \parallel H)/c = \{H \parallel g_n \mid n \geq 1\}.$$

Since $H \parallel a_n \leftrightarrow H$, we have

$$(G \parallel H)/c = \{H \parallel b^n \mid n \geq 1\},$$

modulo $\leftrightarrow$ which does not affect the closure properties (as remarked in 5.1.1). The last set is a Cauchy sequence: in general, if $\{q_i \mid i \geq 1\}$ is a Cauchy sequence of process graphs, then $\{p \parallel q_i \mid i \geq 1\}$ is again a Cauchy sequence for arbitrary p. However, there is no limit for this sequence in the set (G || H)/c, and hence it is not closed. So G || H is not closed.

This counterexample may seem somewhat surprising in view of a related result in [BBKM84], where it is stated (Theorem 2.9) that the collection of closed *trace languages* (containing possibly infinite traces) is closed under the merge operation, for arbitrary alphabet. Here a trace language is obtained as the set of all maximal traces of a process (or process graph). Note however that closure of processes does not very well correspond to closure of the corresponding trace sets; cf. also Example 4.4 in [BBKM84] of a closed process graph with a trace set which is not closed.

Next, we define the quotient structure

$$\mathbb{G}^c{}_{\alpha,\beta} = \mathfrak{G}^c{}_{\alpha,\beta} / \leftrightarrow.$$

Here $\mathfrak{G}^c{}_{\alpha,\beta}$ is supposed to be equipped with the operations as just defined. It is left to the reader to show that $\leftrightarrow$ is indeed a congruence w.r.t. these operations. Now there is the following fact, showing that indeed taking the quotient w.r.t. the congruence induced by AIP can be exchanged for the restriction to closed process graphs:

5.5. THEOREM. $\mathbb{G}^c{}_{\alpha,\beta} \cong \mathbb{G}^\circ{}_{\alpha,\beta}$.

PROOF. Remember that $\mathbb{G}^c{}_{\alpha,\beta} = \mathfrak{G}^c{}_{\alpha,\beta} / \leftrightarrow$ and $\mathbb{G}^\circ{}_{\alpha,\beta} = \mathfrak{G}_{\alpha,\beta} / \equiv$. Define the map

$$\varphi: \mathfrak{G}^c{}_{\alpha,\beta} / \leftrightarrow \;\to\; \mathfrak{G}_{\alpha,\beta} / \equiv$$

by $\varphi(g/\leftrightarrow) = (g/\equiv)$. Here $g \in \mathfrak{G}^c{}_{\alpha,\beta}$ and $g/\leftrightarrow$ is the equivalence class modulo $\leftrightarrow$; likewise $g/\equiv$ is the equivalence class of g modulo $\equiv$.

(1) To prove that φ is injective, let $g,h \in \mathbb{G}^c_{\alpha,\beta}$ and suppose $g \equiv h$. We must prove $g \leftrightarrow h$. Define $R \subseteq NODES(g) \times NODES(h)$ as follows: $(s,t) \in R$ iff $(g)_s \equiv (g)_t$. We claim that R is a bisimulation from g to h. Proof of the claim: The roots are related, by the assumption $g \equiv h$. Further, suppose $(s,t) \in R$ and suppose there is a step $s \rightarrow_a s'$ in g. (See Figure 6.)

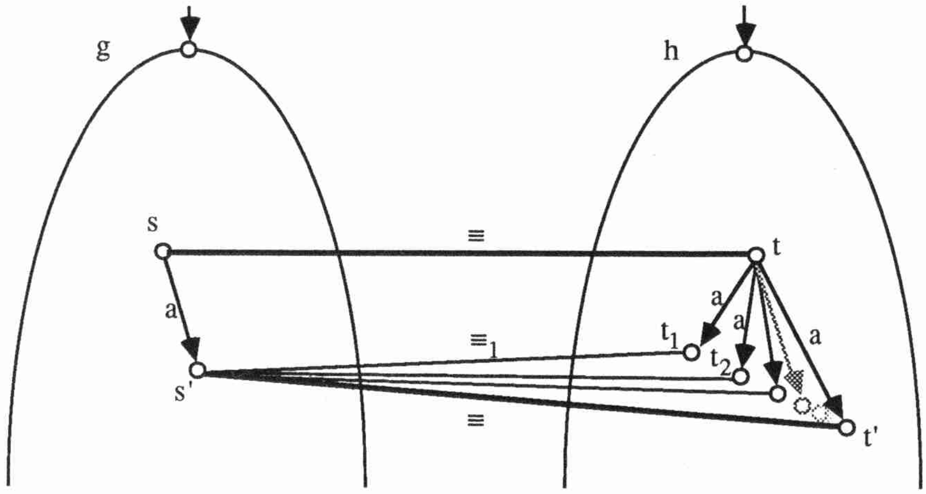


Figure 6

Since $(g)_s \equiv (h)_t$ we have for all $n \geq 1$: $(g)_s \equiv_n (h)_t$. This means that there are t_n such that $(g)_{s'} \equiv_n (h)_{tn}$, for all $n \geq 1$. The t_n (or rather the $(h)_{tn}$) form a Cauchy sequence w.r.t. δ, hence there is, since h is closed, a node t' such that $t \rightarrow_a t'$ and $(h)_{t'}$ is a limit for the Cauchy sequence t_n, $n \geq 1$. So $(h)_{t'} \equiv_n (h)_{tm}$ for some $m \geq n$. Therefore $(h)_{t'} \equiv_n (h)_{tm} \equiv_m (g)_{s'}$, and since $m \geq n$, $(h)_{t'} \equiv_n (g)_{s'}$. This holds for all $n \geq 1$, so $(h)_{t'} \equiv (g)_{s'}$, i.e. $(s',t') \in R$.

The same argument shows that if $(s,t) \in R$ and there is a step $t \rightarrow_a t'$ in h, then there is a step $s \rightarrow_a s'$ with $(s',t') \in R$.

This shows that R is a bisimulation from g to h, and ends the proof of (1).

(2) To prove that φ is surjective, we have to show that

$$\forall g \in \mathbb{G}^c_{\alpha,\beta}\ \exists g' \in \mathbb{G}^c_{\alpha,\beta}\ \ g \equiv g'.$$

This follows by taking $g' = can(g/\equiv)$ and applying Proposition 5.3(iii). □

In the case that β is large enough, so that $\mathbb{G}^c_{\alpha,\beta}$ is isometric to the process model $\mathbb{P}_\alpha$ of De Bakker and Zucker, this isometry leads to an 'explicit representation' of $\mathbb{P}_\alpha$, as follows. First a definition:

5.6. DEFINITION. (i) A process graph g is *minimal* if

$$\forall s,t \in NODES(g)\ \ (g)_s \leftrightarrow (g)_t \ \Rightarrow\ s = t.$$

(ii) A process graph is *normal* if

$$\forall s,t,t' \in NODES(g)\ \forall a \in A\ \ s \rightarrow_a t\ \&\ s \rightarrow_a t'\ \&\ (g)_s \leftrightarrow (g)_t \ \Rightarrow\ s = t.$$

Clearly, normality is implied by minimality. Also note that a process tree can never be minimal, unless it is linear (has only one branch); this is the reason for introducing the concept 'normal'.

It is not hard to prove that if g,h are minimal process graphs and g $\leftrightarrow$ h, then g,h are in fact identical. Moreover, the canonical process graphs (of elements of $\mathbb{G}^\circ_{\alpha,\beta}$) are precisely the closed and minimal process graphs in $\mathfrak{G}_{\alpha,\beta}$. Thus every element in $\mathbb{P}_\alpha$ can be represented by a *closed, minimal process graph with branching degree at most* α^*, and the operations in $\mathbb{P}_\alpha$ can be represented by the corresponding operations in $\mathfrak{G}^c_{\alpha,\beta}$ followed by minimalisation (collapsing all bisimilar subgraphs). Another explicit representation can be given, using trees instead of graphs and observing that normal, bisimilar process trees are identical. Then the elements of $\mathbb{P}_\alpha$ correspond to *closed, normal process trees with branching degree at most* α^*. This representation is closer to the idea of elements of $\mathbb{P}_\alpha$ as 'hereditarily closed and possibly not well-founded sets'.

Summarizing our comparisons with $\mathbb{P}_\alpha$ we have established isometries (for all κ):

$$\mathbb{P}_\alpha \cong \mathbb{A}^\infty{}_{\alpha,\alpha^*+\kappa} \cong \mathbb{G}^\circ{}_{\alpha,(\alpha^*)^+ +\kappa}.$$

Furthermore, writing $\mathfrak{G}^{cm}{}_{\alpha,\beta}$ for the set of closed minimal graphs in $\mathfrak{G}_{\alpha,\beta}$ and $\mathfrak{T}^{cn}{}_{\alpha,\beta}$ for the set of closed normal trees in $\mathfrak{G}_{\alpha,\beta}$, there are the isometries

$$\mathbb{P}_\alpha \cong \mathbb{G}^c{}_{\alpha,(\alpha^*)^+ +\kappa} \cong \mathfrak{G}^{cm}{}_{\alpha,\,(\alpha^*)^+ +\kappa} \cong \mathfrak{T}^{cn}{}_{\alpha,\,(\alpha^*)^+ +\kappa}\,,$$

where the last two complete metric spaces can be seen as 'explicit representations' of $\mathbb{P}_\alpha$.

References

[AR87] P. America & J.J.M.M. Rutten, *Solving reflexive domain equations in a category of complete metric spaces.* Report CS-R8709, Centre for Mathematics and Computer Science, Amsterdam 1987.

[BBKM84] J.W. de Bakker, J.A. Bergstra, J.W. Klop & J.-J.Ch. Meyer, *Linear time and branching time semantics for recursion with merge.* Theoretical Computer Science 34 (1984), p.135-156.

[BZ82a] J.W. de Bakker & J.I. Zucker, *Denotational semantics of concurrency.* Proc. 14th ACM Symp. on Theory of Computing, 153-158, 1982.

[BZ82b] J.W. de Bakker & J.I. Zucker, *Processes and the denotational semantics of concurrency.* Information and Control 54, 1982, 70-120.

[BK82] J.A. Bergstra & J.W. Klop, *Fixed point semantics in process algebras.* Report IW 206/82, Centre for

Mathematics and Computer Science, Amsterdam 1982.

[BK86a] J.A. Bergstra & J.W. Klop, *Algebra of communicating processes.* In: CWI Monographs I, Proceedings of the CWI Symposium Mathematics and Computer Science (eds. J.W. de Bakker, M. Hazewinkel & J.K. Lenstra), North-Holland, Amsterdam, 1986, 89-138.

[BK86b] J.A. Bergstra & J.W. Klop, *Process algebra: specification and verification in bisimulation semantics.* In: CWI Monograph 4, Proceedings of the CWI Symposium Mathematics and Computer Science II (eds. M. Hazewinkel, J.K. Lenstra & L.G.L.T. Meertens), North-Holland, Amsterdam 1986, 61-94.

[Gl87] R.J. van Glabbeek, *Bounded nondeterminism and the approximation principle in process algebra.* In: Proc. of the 4th Annual Symposium on Theoretical Aspects of Computer Science (eds. F.J. Brandenburg, G. Vidal-Naquet and M. Wirsing), Passau (W. Germany) 1987, Springer LNCS 247, 336-347.

[GR83] W.G. Golson & W.C. Rounds, *Connections between two theories of concurrency: metric spaces and synchronization trees.* Information and Control 57 (1983), 102-124.

[Kr86] E. Kranakis, *Fixed point equations with parameters in the projective model.* Report CS-R8606, Centre for Mathematics and Computer Science, Amsterdam 1986. To be published in Information and Computation.

[Mi80] R. Milner, *A Calculus of Communicating Systems,* Springer LCNS 92, 1980.

[Mi85] R. Milner, *Lectures on a calculus for communicating systems.* Seminar on Concurrency, Springer LNCS 197 (1985), 197-220.

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J.A. Bergstra, J.W. Klop

A convergence theorem in process algebra

Computer Science/Department of Software Technology Report CS-R8733 July

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