Unguardedness Mostly Means Many Solutions

Jos C. M. Baeten[a,b], Bas Luttik[b,c]

[a]Dept. of Mechanical Engineering, Eindhoven University of Technology, The Netherlands
[b]Dept. of Math. and Comput. Sci., Eindhoven University of Technology, The Netherlands
[c]Dept. of Comput. Sci., Vrije Universiteit Amsterdam, The Netherlands

Abstract

A widely accepted method to specify (possibly infinite) behaviour is to define it as *the* solution, in some process algebra, of a recursive specification, i.e., a system of recursive equations over the fundamental operations of the process algebra. The method only works if the recursive specification has a unique solution in the process algebra; it is well-known that *guardedness* is a sufficient requirement on a recursive specification to guarantee a unique solution in any of the standard process algebras.

In this paper we investigate to what extent guardedness is also a necessary requirement to ensure unique solutions. We prove a theorem to the effect that all unguarded recursive specifications over BPA have infinitely many solutions in the standard models for BPA. In contrast, we observe that there exist recursive specifications over PA, necessarily involving parallel composition, that have a unique solution, or finitely many solutions in the standard models for PA.

Keywords: Process algebra, recursive specification, unguarded equation.

1. Introduction

In the beginning of the 1980s, concurrency theory was developing rapidly, and many different methods and techniques were applied to this fledgling field. It was found that a concurrent process can be defined as the solution, in some process algebra, of a recursive equation, or a system of recursive equations, provided that this solution is unique. It was established in different ways that *guarded* equations admitted unique solutions in different models, and so, could be used to define a process (as some fixed point). The situation with respect to *unguarded* equations was much less clear. In their seminal paper [3] (published much later as [4]), Bergstra and Klop established that in their projective limit model of concurrency, also every unguarded equation has a solution. Moreover, a solution can be found by starting an iteration from an arbitrary process. This led to the firmly established belief in concurrency theory, that unguarded equations are unsuitable to define processes, as they admit not just one, but many solutions.

In this paper we study unguardedness in more detail. In particular, we shall ask under what conditions, in which process algebras, do unguarded equations have an infinite number of solutions?

Section 2 defines the process algebras and the sets of recursive equations we will consider. Next, in Section 3, we establish that in a *minimal process algebra* (a process algebra just having action prefix and alternative composition, no sequential composition or parallel composition) every unguarded equation has infinitely many different solutions in the standard models, and also every unguarded specification (i.e., finite or countably infinite set of recursion equations) has infinitely many different solutions. In Section 4 we extend this

Table 1: Axioms for the theory $\mathrm{PA}_{\mathbf{01}}$.

A1	$p+q$	$=$	$q+p$	S1	$(p+q)\cdot r$	$=$	$p\cdot r+q\cdot r$
A2	$(p+q)+r$	$=$	$p+(q+r)$	S2	$(p\cdot q)\cdot r$	$=$	$p\cdot(q\cdot r)$
A3	$p+p$	$=$	p	S3	$(a.p)\cdot q$	$=$	$a.(p\cdot q)$
A4	$p+\mathbf{0}$	$=$	p	S4	$\mathbf{0}\cdot p$	$=$	$\mathbf{0}$
				S5	$\mathbf{1}\cdot p$	$=$	p
P	a	$=$	$a.\mathbf{1}$	S6	$p\cdot\mathbf{1}$	$=$	p
M	$p \parallel q$	$=$	$p \mathbin{\|\!\!\lfloor} q + q \mathbin{\|\!\!\lfloor} p$	L4	$\mathbf{0} \mathbin{\|\!\!\lfloor} p$	$=$	$\mathbf{0}$
				L5	$\mathbf{1} \mathbin{\|\!\!\lfloor} \mathbf{0}$	$=$	$\mathbf{0}$
L1	$a \mathbin{\|\!\!\lfloor} p$	$=$	$a.p$	L6	$\mathbf{1} \mathbin{\|\!\!\lfloor} \mathbf{1}$	$=$	$\mathbf{1}$
L2	$a.p \mathbin{\|\!\!\lfloor} q$	$=$	$a.(p \mathbin{\|\!\!\lfloor} q)$	L7	$\mathbf{1} \mathbin{\|\!\!\lfloor} a.p$	$=$	$\mathbf{0}$
L3	$(p+q) \mathbin{\|\!\!\lfloor} r$	$=$	$p \mathbin{\|\!\!\lfloor} r + q \mathbin{\|\!\!\lfloor} r$	L8	$\mathbf{1} \mathbin{\|\!\!\lfloor} (p+q)$	$=$	$\mathbf{1} \mathbin{\|\!\!\lfloor} p + \mathbf{1} \mathbin{\|\!\!\lfloor} q$

result to include sequential composition, but then, in Section 5 we find these results fail as soon as we include a form of parallel composition.

2. Process Algebras and Recursive Specifications

2.1. Process algebras

In [3], Bergstra and Klop put forward the term *process algebra* to denote models of the axiomatic process theory PA. In this paper it is convenient to assign a slightly more general meaning to the term process algebra: we use it to denote an arbitrary nonempty set $\mathbf{P}$ (the elements of which are thought of as processes) together with some special distinguished processes (atomic actions, deadlock, termination) and a family of fundamental process-theoretic operations (action prefix, sequential composition, alternative composition, parallel composition), satisfying some collection of equational axioms. So, we allow different signatures and different axiomatizations.

Let $\mathcal{A}$ be a non-empty finite set of *actions.* (As in [3], we assume that $\mathcal{A}$ is finite; the results in Sections 3 and 4 remain valid for an infinite alphabet, but this is less clear for the remarks in Section 5.) The process theory PA as presented in [3, 4] considers the actions in $\mathcal{A}$ as constants in the theory, denoting distinguished atomic processes, and includes binary operations for *sequential composition* (denoted by $\cdot$), *alternative composition* (denoted by $+$), *left merge* (denoted by $\mathbin{\|\!\!\lfloor}$) and *parallel composition* (denoted by $\parallel$). Here, we want to consider a slight reformulation of PA. In the original formulation, the action constants denote both action execution and termination. When different forms of termination are considered (especially in extensions with timing) it is useful to separate the two, and to replace action constants by action prefix in combination with termination constants. Thus we declare, in addition, the existence of a *deadlocked* process (denoted by the constant $\mathbf{0}$) and a *successfully terminated* process (denoted by the constant $\mathbf{1}$), and have unary *action prefixes* $a.$ ($a \in \mathcal{A}$). Assuming the presence of the constants $\mathbf{0}$ and $\mathbf{1}$ and the action prefixes will considerably simplify the presentation in the remainder of the paper, but their presence is not essential for the results that we are going to obtain.

We denote the process theory that consists of all the aforementioned constants and operations, together with all the axioms listed in Table 1, by $\mathrm{PA}_{\mathbf{01}}$. We shall also consider in this paper the following two subtheories of $\mathrm{PA}_{\mathbf{01}}$: the subtheory $\mathrm{MPA}_{\mathbf{01}}$ (for *minimal process algebra*) consists of the constants $\mathbf{0}$ and $\mathbf{1}$, the unary action prefixes $a.$ ($a \in \mathcal{A}$) and the binary operation $+$, together with the axioms A1–A4; the subtheory $\mathrm{BPA}_{\mathbf{01}}$ (for *basic process algebra*) extends $\mathrm{MPA}_{\mathbf{01}}$ with the binary operation $\cdot$ for sequential composition and the axioms S1–S6. We note that the theory PA of [4] can also be obtained as a subtheory of $\mathrm{PA}_{\mathbf{01}}$, by excluding the action prefixes and the constants $\mathbf{0}$ and $\mathbf{1}$, and by replacing in L1 and L2 $a.p$ by $a \cdot p$ and $a.(p \mathbin{|\!\lfloor} q)$ by $a \cdot (p \mathbin{|\!\lfloor} q)$.

We proceed to briefly describe two models of the theory $\mathrm{PA}_{\mathbf{01}}$: a projective limit model and a model of countably branching process graphs modulo bisimilarity. The first was considered in [3, 4], the second has since then become the standard model. (The model of countably branching process graphs modulo bisimilarity is isomorphic to the so-called *term model* that consists of transition systems —associated with recursive specifications via structural operational semantics [6]— modulo bisimilarity; see, e.g., [1]. The phrase 'term model' is really a misnomer since it not just considers finite terms but also terms built using infinite recursive specifications.)

Projective limit model. Denote by $\mathbf{I}$ the initial algebra associated with the process theory $\mathrm{PA}_{\mathbf{01}}$. That is, $\mathbf{I}$ is the quotient of the set of all (ground) terms that can be built from the constants and operations of $\mathrm{PA}_{\mathbf{01}}$ modulo the congruence induced on them by derivability from the axioms in Table 1 using the rules of equational logic.

For $p_1, \ldots, p_k \in \mathbf{I}$ ($k \in \omega$), the indexed sum is inductively defined as follows: if $k = 0$, then $\sum_{1 \le i \le k} p_i = \mathbf{0}$, and if $k > 0$, then $\sum_{1 \le i \le k} p_i = (\sum_{1 \le i \le k-1} p_i) + p_k$.

Proposition 1. *Modulo the equivalence on (ground)* $\mathrm{PA}_{\mathbf{01}}$*-terms generated by the axioms of* $\mathrm{PA}_{\mathbf{01}}$*, the initial algebra* $\mathbf{I}$ *is inductively generated by the following rule: if* $a_i \in \mathcal{A}$ *and* $p_i \in \mathbf{I}$ *(*$1 \le i \le k$*), then*

$$\sum_{1 \le i \le k} a_i.p_i[\,+\,\mathbf{1}] \in \mathbf{I}\ .$$

(By putting $+\,\mathbf{1}$ *between square brackets we mean that it is optional.)*

According to the above proposition, occurrences of the constants $a \in \mathcal{A}$, superfluous occurrences of the constants $\mathbf{0}$ and $\mathbf{1}$, and occurrences of the operations $\cdot$, $\|$ and $\mathbin{|\!\lfloor}$ can be eliminated from ground $\mathrm{PA}_{\mathbf{01}}$-terms by means of the axioms of $\mathrm{PA}_{\mathbf{01}}$. The elements of the subset of ground $\mathrm{PA}_{\mathbf{01}}$-terms generated by the rule in the above proposition are often referred to as *basic terms.*

On $\mathbf{I}$ we now define, for $n \ge 0$, projection operations $\pi_n : \mathbf{I} \to \mathbf{I}$ inductively as follows:

$$\pi_0(\sum_{1 \le i \le k} a_i.p_i[\,+\,\mathbf{1}]) = \mathbf{0}[\,+\,\mathbf{1}]\ \text{, and}$$
$$\pi_{n+1}(\sum_{1 \le i \le n} a_i.p_i[\,+\,\mathbf{1}]) = \sum_{1 \le i \le k} a_i.\pi_n(p_i)[\,+\,\mathbf{1}]\ .$$

A *projective sequence* over $\mathbf{I}$ is a sequence $(p_i)_{i<\omega}$ of elements of $\mathbf{I}$ such that, for all $i < \omega$,

$$p_i = \pi_i(p_{i+1})\ .$$

We denote by $\mathbf{I}^\infty$ the process algebra that consists of all projective sequences over $\mathbf{I}$, with the operations of $\mathrm{PA}_{\mathbf{01}}$ induced on them component-wise. The process algebra $\mathbf{I}^\infty$ satisfies the axioms of $\mathrm{PA}_{\mathbf{01}}$ and is called the *projective limit model* for $\mathrm{PA}_{\mathbf{01}}$ (see, e.g., [3, 5].

Graph model. An alternative model for $\mathrm{PA}_{\mathbf{01}}$ is obtained by considering the set G of all countably branching *process graphs* (i.e., rooted, directed graphs with edges labelled with actions in $\mathcal{A}$, and vertices possibly labelled with a termination label $\downarrow$). A *bisimulation* between process graphs g and h is a binary relation between the vertices of g and the vertices of h such that if v is a vertex of g and w is a vertex of h and v and w are related, then for all $a \in \mathcal{A}$;

(i) if $v \xrightarrow{a} v'$ is an edge in g, then there exists an edge $w \xrightarrow{a} w'$ in h such that v' and w' are related;
(ii) if $w \xrightarrow{a} w'$ is an edge in h, then there exists an edge $v \xrightarrow{a} v'$ in g such that v' and w' are related; and
(iii) if a vertex v of g is related to a vertex w of h, then v has the termination label $\downarrow$ if, and only if, w has the termination label.

Process graphs g and h are *bisimilar* (notation: $g \leftrightarrows h$) if there exists a bisimulation relation between them relating their roots.

It is well-known how the operations of $\mathrm{PA}_{\mathbf{01}}$ can be defined on G in such a way that bisimilarity is a congruence and the quotient $\mathbf{G} = G/{\leftrightarrows}$ is a model of $\mathrm{PA}_{\mathbf{01}}$ (see, e.g., [2, 1]).

Let φ be the homomorphism from $\mathbf{G}$ into $\mathbf{I}^{\infty}$ that maps every process graph to the projective sequence consisting of all its finite projections (see, e.g., [4] for details). The kernel of this homomorphism identifies all process graphs that have bisimilar finite projections. (Thus, e.g., the process graph that is the alternative composition of all sequences $a^n.\mathbf{0}$ for some $a \in \mathcal{A}$ is identified with the graph having as additional branch an infinite sequence of a's.) As we will see further on, this homomorphism is not surjective: there exist projective sequences that can only be the images of uncountably branching process graphs. Denote by $\mathbf{I}^{\infty}_{\aleph_1}$ the subalgebra of $\mathbf{I}^{\infty}$ consisting of the homomorphic image of $\mathbf{G}$. As the example above shows, $\mathbf{G}$ and $\mathbf{I}^{\infty}_{\aleph_1}$ are not isomorphic.

Let $\mathbf{P}$ be a process algebra, and let p be an element of $\mathbf{P}$. A *summand* of p in $\mathbf{P}$ is an element q in $\mathbf{P}$ such that $p + q = p$. We call a summand q of p *simple* if $q = a.q'$ for some q' in $\mathbf{P}$.

Lemma 2. *In the process algebras* $\mathbf{G}$ *and* $\mathbf{I}^{\infty}_{\aleph_1}$ *all processes have at most countably many simple summands.*

Proof. In $\mathbf{G}$, if $p + a.q = p$, then every process graph in the bisimulation equivalence class $a.q$ is bisimilar to a branch of every graph in the bisimulation equivalence class p. Since every process graph in p has a countable branching degree, it follows that, up to bisimilarity, there are at most countably many distinct q such that $p + a.q = p$.

To prove that in $\mathbf{I}^{\infty}_{\aleph_1}$ all processes have countably many simple summands, consider elements p and q of $\mathbf{G}$, and suppose that, for some $a \in \mathcal{A}$, $\varphi(p + a.q) = \varphi(p)$. Then all finite projections of a graph in $p + a.q$ are bisimilar to the finite projections of a graph g in p, and therefore all finite projections of a graph h in $a.q$ are bisimilar to the finite projections of a branch of g. This branch gives rise to a summand $a.q'$ of p and clearly $\varphi(a.q') = \varphi(a.q)$. We have established that for every simple summand $\varphi(a.q)$ of $\varphi(p)$ in $\mathbf{I}^{\infty}_{\aleph_1}$ there is a simple summand $a.q'$ of p in $\mathbf{G}$ such that $\varphi(a.q') = \varphi(a.q)$. Since in $\mathbf{G}$ every process has at most countably many summands, it follows that in $\mathbf{I}^{\infty}_{\aleph_1}$ every process has countably many summands too. □

2.2. Recursive specifications

Let $\mathcal{V}$ be a countably infinite set of *variables*, and let α be a countable ordinal. Furthermore, let $\mathbf{P}$ be a process algebra. A *recursive specification* (of dimension α) with respect to $\mathbf{P}$ is a sequence of equations

$$(x_\kappa \overset{\mathrm{def}}{=} t_\kappa)_{\kappa<\alpha} \ ,$$

with $(x_\kappa)_{\kappa<\alpha}$ a sequence of distinct variables and $(t_\kappa)_{\kappa<\alpha}$ a sequence of process terms, i.e., terms built from (symbols denoting) the operations of $\mathbf{P}$ and the variables in $(x_\kappa)_{\kappa<\alpha}$.

In this paper we shall limit our attention to recursive specifications of at most countable dimension (cf. Remark 6 below).

Let $\mathbf{P}$ be a process algebra, and let t be a process term built from the operations of $\mathbf{P}$ and variables in the sequence $\vec{x} = (x_\kappa)_{\kappa<\alpha}$. Furthermore, let, for some $\beta \geq \alpha$, $\vec{p} = (p_\kappa)_{\kappa<\beta}$ be a sequence of elements of $\mathbf{P}$. We denote by $[\![t]\!]_{\vec{p}}$ the element of $\mathbf{P}$ that results from evaluating t with, for all $\kappa < \alpha$, the element p_κ assigned to the variable x_κ.

A *solution* in $\mathbf{P}$ of a recursive specification $(x_\kappa \overset{\text{def}}{=} t_\kappa)_{\kappa<\alpha}$ is a sequence of processes

$$\vec{p} = (p_\kappa)_{\kappa<\alpha}$$

such that $[\![t_\kappa]\!]_{\vec{p}} = p_\kappa$ for all $\kappa < \alpha$.

A process algebra $\mathbf{P}$ satisfies the *recursive definition principle* (RDP) if every recursive specification with respect to $\mathbf{P}$ has at least one solution in $\mathbf{P}$.

Proposition 3. *The process algebras* $\mathbf{G}$, $\mathbf{I}^\infty$, *and* $\mathbf{I}^\infty_{\aleph_1}$ *satisfy RDP.*

Proof. That $\mathbf{G}$ satisfies RDP is proved in [2]. Since $\mathbf{I}^\infty_{\aleph_1}$ is a homomorphic image of $\mathbf{G}$, it follows that it also satisfies RDP. (Indeed, if $\vec{p} = (p_\kappa)_{\kappa<\alpha}$ is a solution of a recursive specification $\mathcal{S} = (x_\kappa \overset{\text{def}}{=} t_\kappa)_{\kappa<\alpha}$ in $\mathbf{G}$, then clearly $\varphi(\vec{p}) = (\varphi(t_\kappa))_{\kappa<\alpha}$ is a solution of $\mathcal{S}$ in $\mathbf{I}^\infty_{\aleph_1}$.) Clearly, since $\mathbf{I}^\infty_{\aleph_1}$ is a subalgebra of $\mathbf{I}^\infty$, a solution in $\mathbf{I}^\infty_{\aleph_1}$ of some recursive specification is also in $\mathbf{I}^\infty$, and hence $\mathbf{I}^\infty$ also satisfies RDP. □

Remark 4. Many process algebras do not satisfy RDP. E.g., in the initial algebra $\mathbf{I}$, there is no solution for the equation $x = a.x$.

Suppose that the process algebra $\mathbf{P}$ is endowed with a sequence of unary operations $a.$ associated with the actions $a \in \mathcal{A}$, let x be a variable and let t be a term built from variables and the fundamental operations of $\mathbf{P}$. An occurrence of a variable x in a process term t is *guarded* if it is within the scope of an action prefix, i.e., if there exists an action a and a term t' such that $a.t'$ is a subterm of t and the occurrence of x is within t'. A term t is *completely guarded* if it has no unguarded occurrences of variables. Let $\mathcal{S} = (x_\kappa \overset{\text{def}}{=} t_\kappa)_{\kappa<\alpha}$ be a recursive specification. We say that $\mathcal{S}$ is *completely guarded* if t_κ is completely guarded for all $\kappa < \alpha$.

The notion of complete guardedness introduced above is syntactic, and it may happen that a specification with unguarded occurrences of variables can be transformed into a completely guarded specification with exactly the same solutions. We are usually interested in a more robust notion of guardedness that is invariant under solution-preserving transformations. To define such a notion of guardedness, we need a notion of equivalence on specifications. Let $\mathcal{S}$ and $\mathcal{S}'$ be recursive specifications with respect to $\mathbf{P}$. We say that $\mathcal{S}$ and $\mathcal{S}'$ are *equivalent* (notation: $\mathcal{S} \approx_{\mathbf{P}} \mathcal{S}'$) if they have the same solutions in $\mathbf{P}$. We call $\mathcal{S}$ *guarded* if there exists an equivalent completely guarded recursive specification, and *unguarded* otherwise.

A process algebra $\mathbf{P}$ satisfies the *recursive specification principle* (RSP) if every guarded recursive specification with respect to $\mathbf{P}$ has a unique solution in $\mathbf{P}$.

Proposition 5. *The process algebras* $\mathbf{G}$, $\mathbf{I}^\infty$ *and* $\mathbf{I}^\infty_{\aleph_1}$ *satisfy RSP.*

2.3. Definability

An element p of a process algebra $\mathbf{P}$ is called *definable* if there exists a recursive specification $(x_\kappa \overset{\text{def}}{=} t_\kappa)_{\kappa<\alpha}$ with a unique solution $(p_\kappa)_{\kappa<\alpha}$ in $\mathbf{P}$ such that $p_0 = p$.

Remark 6. If we would allow recursive specifications of uncountable size, then the set of definable processes remains the same. For, the right-hand side of the equation for the start variable contains finitely many variables of the specification; for each of these, finitely many more variables can be reached. Thus, the reachability tree of variables reachable from the initial variable is a finitely branching tree of countable depth. This means the equation of only countably many variables are relevant for the process defined by the initial variable, and uncountable definability is the same as countable definability.

Proposition 7. *The process algebras* $\mathbf{G}$*,* $\mathbf{I}^\infty$ *and* $\mathbf{I}^\infty_{\aleph_1}$ *have uncountably many definable elements.*

Proof. Let $\{0,1\}^\infty$ be the set of infinite sequences over $\{0,1\}$. We first associate with every $\sigma \in \{0,1\}^\infty$ a guarded recursive specification $\mathcal{S}_\sigma$. Let $\sigma = (\sigma_i)_{i<\omega}$ with $\sigma_i \in \{0,1\}$; we define $\mathcal{S}_\sigma = (x_i \overset{\text{def}}{=} t_i)_{i<\omega}$ by

$$t_i = \begin{cases} a.x_{i+1} & \text{if } \sigma_i = 0 \\ a.x_{i+1} + a.\mathbf{0} & \text{if } \sigma_i = 1 \end{cases}$$

Since $\mathcal{S}_\sigma$ is guarded, it has a unique solution in $\mathbf{G}$, $\mathbf{I}^\infty$ and $\mathbf{I}^\infty_{\aleph_1}$. Now, since $\{0,1\}^\infty$ has uncountably many elements, in order to prove that $\mathbf{G}$, $\mathbf{I}^\infty$ and $\mathbf{I}^\infty_{\aleph_1}$ have uncountably many elements, it suffices to prove that if $\sigma \neq \tau$, then the solutions of $\mathcal{S}_\sigma$ and $\mathcal{S}_\tau$ in $\mathbf{G}$, $\mathbf{I}^\infty$ and $\mathbf{I}^\infty_{\aleph_1}$ are distinct.

Recall that φ denotes the homomorphism from $\mathbf{G}$ into $\mathbf{I}^\infty$ that maps process graphs to projective sequences consisting of their finite projections. Note that if p and q are the unique solutions in $\mathbf{G}$ of $\mathcal{S}_\sigma$ and $\mathcal{S}_\tau$, respectively, then $\varphi(p)$ and $\varphi(q)$ are the unique solutions of $\mathcal{S}_\sigma$ and $\mathcal{S}_\tau$ both in $\mathbf{I}^\infty$ and $\mathbf{I}^\infty_{\aleph_1}$.

Therefore, it remains to establish that the unique solutions of $\mathcal{S}_\sigma$ and $\mathcal{S}_\tau$ in $\mathbf{I}^\infty_{\aleph_1}$ are distinct. To this end, suppose that p and q are the unique solutions of $\mathcal{S}_\sigma$ and $\mathcal{S}_\tau$ in $\mathbf{I}^\infty_{\aleph_1}$. Then, since $\sigma \neq \tau$, there exists $i < \omega$ such that $\sigma_i \neq \tau_i$, and hence $\pi_{i+2}(p) \neq \pi_{i+2}(q)$. It follows that $p \neq q$. □

Having defined the projective sequences p_σ that are the solutions of specifications $\mathcal{S}_\sigma$, we can describe an element of $\mathbf{I}^\infty$ that is not in $\mathbf{I}^\infty_{\aleph_1}$. It is found by considering the finite projections of the process $\sum_{\sigma\in\{0,1\}^\infty} a.p_\sigma$ for some $a \in \mathcal{A}$. The process graph of this process is uncountably branching, and the first five elements of the projective sequence are

$$\begin{aligned}
&\mathbf{0}\ ,\\
&a.\mathbf{0}\ ,\\
&a.a.\mathbf{0}\ ,\\
&a.a.a.\mathbf{0} + a.(a.\mathbf{0} + a.a.\mathbf{0})\ ,\ \text{and}\\
&a.a.a.a.\mathbf{0} + a.(a.\mathbf{0} + a.a.a.\mathbf{0}) + a.a.(a.\mathbf{0} + a.a.\mathbf{0}) + a.(a.\mathbf{0} + a.(a.\mathbf{0} + a.a.\mathbf{0}))\ ,
\end{aligned}$$

doubling the number of summands at every step thereafter. As this process is uncountably branching, it cannot be definable: there is no recursive specification (of any size) having this process as solution of the initial variable.

3. Minimal process algebra

A *minimal process algebra* is a process algebra $\mathbf{P}$ that is a model for the process theory $\text{MPA}_{\mathbf{01}}$, i.e., $\mathbf{P}$ has constants $\mathbf{0}$ and $\mathbf{1}$, unary action prefixes $a.$ ($a \in \mathcal{A}$), and a binary operation $+$, and satisfies the axioms A1–A4 of Table 1. Henceforth, we shall refer to process terms built from variables and the constants and operations of the process theory $\text{MPA}_{\mathbf{01}}$ as $\text{MPA}_{\mathbf{01}}$-process terms.

An occurrence of a variable x in an $\text{MPA}_{\mathbf{01}}$-process term t is *unguarded* if it is not within the scope of an action prefix, i.e., if there does not exists an action a and a term t' such that $a.t'$ is a subterm of t and the occurrence of x is within t'.

Lemma 8. *Let* $\mathbf{P}$ *be a minimal process algebra and let* $\mathcal{S} = (x_\kappa \overset{\text{def}}{=} t_\kappa)_{\kappa<\alpha}$ *be a recursive specification with respect to* $\mathbf{P}$*. If* t_κ *has an unguarded occurrence of the variable* x_λ*, then* $[\![t_\kappa]\!]_{\vec{p}} = [\![t_\kappa]\!]_{\vec{p}} + [\![x_\lambda]\!]_{\vec{p}}$ *for all solutions* $\vec{p}$ *of* $\mathcal{S}$*.*

Proof. By straightforward application of the axioms A1–A3. □

Let $\mathcal{S}$ be a recursive specification with respect to a minimal process algebra. On the variables $(x_\kappa)_{\kappa<\alpha}$ of $\mathcal{S}$ we define a binary relation $\rightarrow_{\mathcal{S}}$ by $x_\kappa \rightarrow_{\mathcal{S}} x_\lambda$ if x_λ has an unguarded occurrence in t_κ.

Lemma 9. *Let* $\mathbf{P}$ *be a minimal process algebra, and let* $\mathcal{S}$ *be a recursive specification with respect to* $\mathbf{P}$*. If* $\rightarrow_{\mathcal{S}}$ *is terminating, then* $\mathcal{S}$ *is guarded.*

Proof. Note that replacing an occurrence of a variable x_κ in a term t_λ by its definition t_κ preserves solutions. If $\rightarrow_{\mathcal{S}}$ is terminating, then also the procedure that repeatedly replaces an unguarded occurrence of a variable by its definition terminates. Clearly, this procedure transforms $\mathcal{S}$ into an equivalent completely guarded recursive specification. □

We now prove that every recursive specification over $\text{MPA}_{\mathbf{01}}$has infinitely many solutions. In the next section we shall extend this result to a larger theory. Here, the method is basically by showing that we can add an arbitrary summand to the solutions of the unguarded variables. Thus, to give an example, if we have the equation $x = a.x + x$, then the process that keeps on doing a is a solution, but adding an arbitrary definable element p to the equation of the unguarded variable x we obtain another solution $a.(a.(a.(\ldots)+p)+p)+p$.

Theorem 10. *Let* $\mathbf{P}$ *be a minimal process algebra satisfying* RDP*. If for every countable subset* P *of* $\mathbf{P}$ *there exists a definable element* q *in* $\mathbf{P}$ *that is not a summand of any of the elements of* $\mathbf{P}$*, then every unguarded recursive specification with respect to* $\mathbf{P}$ *has infinitely many solutions in* $\mathbf{P}$*.*

Proof. Consider an arbitrary unguarded recursive specification $\mathcal{S}$. Then, by Lemma 9, there exists an infinite sequence $\mathcal{U} = (\kappa_i)_{i<\omega}$ such that $x_{\kappa_i} \rightarrow_{\mathcal{S}} x_{\kappa_{i+1}}$; such a sequence $\mathcal{U}$ we shall call an *unguardedness sequence* in $\mathcal{S}$. We proceed to inductively define an infinite sequence of specifications $(\mathcal{S}_j)_{j<\omega}$ such that $\mathcal{S}_0 = \mathcal{S}$ and, for all $j < \omega$,

(i) $\mathcal{U}$ is an unguardedness sequence in $\mathcal{S}_j$;
(ii) $\mathcal{S}_j$ has a solution that differs at κ_i $(i < \omega)$ from every solution of $\mathcal{S}_{j+1}$; and
(iii) an initial segment of any solution of $\mathcal{S}_{j+1}$ is a solution of $\mathcal{S}_j$.

Suppose that $\mathcal{S}_j = (x_\kappa \overset{\text{def}}{=} t_\kappa)_{\kappa<\alpha}$. Since $\mathbf{P}$ satisfies RDP, $\mathcal{S}_j$ has a solution, say $\vec{p} = (p_\kappa)_{\kappa<\alpha}$, in $\mathbf{P}$. Let $P = \{p_{\kappa_i} \mid i < \omega\}$. By the assumption of the theorem, there exists a definable element q in $\mathbf{P}$ such that $p + q \neq p$ for all $p \in P$. Since q is definable, there exists a recursive specification, say $(y_\lambda \overset{\text{def}}{=} u_\lambda)_{\lambda<\beta}$, with a unique solution $(q_\lambda)_{\lambda<\beta}$ such that $q_0 = q$. We may assume without loss of generality that $x_\kappa \neq y_\lambda$ for all $\kappa < \alpha$ and $\lambda < \beta$.

For $\mu \geq \alpha$, we denote by $\mu - \alpha$ the unique ordinal λ such that $\mu = \alpha + \lambda$. We prove that the specification $\mathcal{S}_{j+1} = (z_\mu \overset{\text{def}}{=} v_\mu)_{\mu<\alpha+\beta}$, with z_μ and v_μ defined by

$$z_\mu = \begin{cases} x_\mu & \text{if } \mu < \alpha \\ y_{\mu-\alpha} & \text{if } \alpha \leq \mu < \alpha+\beta \end{cases} \quad \text{and} \quad v_\mu = \begin{cases} t_\mu & \text{if } \mu < \alpha \text{ and } \mu \notin \mathcal{U} \\ t_\mu + z_\alpha & \text{if } \mu < \alpha \text{ and } \mu \in \mathcal{U} \\ u_{\mu-\alpha} & \text{if } \alpha \leq \mu < \alpha+\beta \end{cases},$$

satisfies the three requirements above.

(i) By construction, $\mathcal{U}$ is an unguardedness sequence in $\mathcal{S}_{j+1}$.

(ii) Consider a solution $\vec{r} = (r_\mu)_{\mu<\alpha+\beta}$ of $\mathcal{S}_{j+1}$. Clearly, $(r_\mu)_{\alpha\leq\mu<\alpha+\beta} = (q_\lambda)_{\lambda<\beta}$, so, in particular, it follows that $[\![z_\alpha]\!]_{\vec{r}} = r_\alpha = q$. Hence, for all $\kappa_i \in \mathcal{U}$,

$$\begin{aligned} r_{\kappa_i} &= [\![v_{\kappa_i}]\!]_{\vec{r}} && (\vec{r} \text{ is a solution of } \mathcal{S}_{j+1}) \\ &= [\![t_{\kappa_i}]\!]_{\vec{r}} + [\![z_\alpha]\!]_{\vec{r}} && (\text{definition of } v_{\kappa_i}) \\ &= [\![t_{\kappa_i}]\!]_{\vec{r}} + q \ \ , \end{aligned}$$

from which it follows by A2 and A3 that $r_{\kappa_i} = r_{\kappa_i} + q$. Now, since $p_{\kappa_i} \neq p_{\kappa_i} + q$, it follows that $\vec{p}$ is not an initial segment of $\vec{r}$.

(iii) To prove that the initial segment $\vec{r} = (r_\kappa)_{\kappa<\alpha}$ of a solution of $\mathcal{S}_{j+1}$ is a solution of $\mathcal{S}_j$, we need to show that $[\![t_\kappa]\!]_{\vec{r}} = r_\kappa$ for all $\kappa < \alpha$. We distinguish cases according to whether $\kappa \in \mathcal{U}$ or not. The case when $\kappa \notin \mathcal{U}$ is straightforward, since then $v_\kappa = t_\kappa$. So suppose that $\kappa \in \mathcal{U}$, and let λ be the $\mathcal{U}$-successor of κ; then

$$\begin{aligned} [\![t_\kappa]\!]_{\vec{r}} &= [\![t_\kappa]\!]_{\vec{r}} + r_\lambda && (\text{Lemma 8}) \\ &= [\![t_\kappa]\!]_{\vec{r}} + (r_\lambda + r_\alpha) && (\text{see the proof of item (ii)}) \\ &= ([\![t_\kappa]\!]_{\vec{r}} + r_\lambda) + r_\alpha && (\text{A2}) \\ &= [\![t_\kappa]\!]_{\vec{r}} + r_\alpha && (\text{Lemma 8}) \\ &= [\![v_\kappa]\!]_{\vec{r}} && (\text{definition of } v_\kappa) \\ &= r_\kappa \ \ . \end{aligned}$$

With the infinite sequence of specifications $(\mathcal{S}_j)_{j<\omega}$ defined above we can associate an infinite sequence $(\vec{p}_j)_{j<\omega}$ such that, for all $i < \omega$, $\vec{p}_i$ is a solution of $\mathcal{S}_j$, but $\vec{p}_j$ is *not* an initial segment of any of the solutions of the specifications $\mathcal{S}_k$ $(j < k < \omega)$. Now, since for all $j < \omega$, an initial segment of $\vec{p}_j$ is a solution of $\mathcal{S}$ and these initial segments are all pairwise distinct, we conclude that $\mathcal{S}$ has infinitely many solutions. □

Remark 11. From the proof of the above theorem it is clear that for all *finite* unguarded recursive specifications with respect to $\mathbf{P}$ to have infinitely many solutions it suffices that there exist for all *finite* subsets P of $\mathbf{P}$ a definable element q in $\mathbf{P}$ such that $p + q \neq p$ for all p in $\mathbf{P}$.

By the MPA$_{\mathbf{01}}$-*reduct* of a process algebra $\mathbf{P}$ we mean the algebra obtained from $\mathbf{P}$ by forgetting all process-theoretic operations except $\mathbf{0}$, $\mathbf{1}$, $a.$ $(a \in \mathcal{A})$ and $+$.

Corollary 12. *With respect to the* MPA$_{\mathbf{01}}$*-reducts of* $\mathbf{G}$ *and* $\mathbf{I}^\infty$*, all unguarded recursive specifications have infinitely many solutions.*

Proof. First, we establish that all unguarded recursive specifications with respect to $\mathbf{G}$ have infinitely many solutions in $\mathbf{G}$. By Proposition 3, $\mathbf{G}$ satisfies RDP. Let P be a countable subset of $\mathbf{G}$. Define Q as the subset of $\mathbf{G}$ consisting of all processes q such that $a.q$ is a simple summand of an element of P. By Lemma 2, Q is countable, and since, by Proposition 7, $\mathbf{G}$ has uncountably many definable elements, there exists a definable element q' in $\mathbf{G}$ such that $a.q'$ is not a summand of any of the elements of $\mathbf{P}$. So, $\mathbf{G}$ satisfies the requirements of Theorem 10, and hence every recursive specification with respect to $\mathbf{G}$ has infinitely many solutions.

By similar arguments it can now be established that also $\mathbf{I}^\infty_{\aleph_1}$ satisfies the requirements of Theorem 10, and hence every recursive specification with respect to $\mathbf{I}^\infty_{\aleph_1}$ has infinitely many solutions. Since $\mathbf{I}^{\infty}_{\aleph_1}$ is a subalgebra of $\mathbf{I}^{\infty}$, a recursive specification $\mathcal{S}$ with respect to (the $\mathrm{MPA}_{\mathbf{01}}$-reduct of) $\mathbf{I}^{\infty}$ is also a recursive specification with respect to (the $\mathrm{MPA}_{\mathbf{01}}$-reduct of) $\mathbf{I}^{\infty}_{\aleph_1}$, and, clearly, a solution of $\mathcal{S}$ in $\mathbf{I}^{\infty}_{\aleph_1}$ is also a solution of $\mathcal{S}$ in $\mathbf{I}^{\infty}$. It follows that every unguarded recursive specification $\mathcal{S}$ with respect to $\mathbf{I}^{\infty}$ has infinitely many solutions. □

4. Basic process algebra

A *basic process algebra* is a process algebra $\mathbf{P}$ that is a model for the process theory $\mathrm{BPA}_{\mathbf{01}}$, i.e., it has constants $\mathbf{0}$ and $\mathbf{1}$, unary action prefixes $a.$ ($a \in \mathcal{A}$), and two binary operations $+$ and $\cdot$, and satisfies the axioms A1–A4 and S1–S6 in Table 1. Henceforth, we shall refer to process terms built from variables and the constants and operations of the process theory $\mathrm{BPA}_{\mathbf{01}}$ as $\mathrm{BPA}_{\mathbf{01}}$-process terms.

We shall prove in this section that, under certain mild conditions on a basic process algebra $\mathbf{P}$, every unguarded recursive specification with respect to $\mathbf{P}$ has infinitely many solutions. Our proof is along the same lines as the proof of the main result in the previous section. There are, however, some subtleties that we need to take into account, as illustrated in the next example.

Example 13. 1. Let $\mathcal{S}_1$ be the recursive specification with respect to $\mathbf{G}$ consisting of the equations

$$x_1 \overset{\text{def}}{=} (a.x_1)\cdot x_2 \ \text{ , and}$$
$$x_2 \overset{\text{def}}{=} (b.x_2)\cdot x_1 \ \text{ .}$$

Then, on the one hand, y has an unguarded occurrence in $(a.x)\cdot y$ in the sense of the previous section. On the other hand, $\mathcal{S}_1$ is guarded, for, by axiom S3 it is equivalent to the recursive specification consisting of the equations

$$x_1 \overset{\text{def}}{=} a.(x_1\cdot x_2) \ \text{ , and}$$
$$x_2 \overset{\text{def}}{=} b.(x_2\cdot x_1) \ \text{ .}$$

So $\mathcal{S}_1$ has a unique solution, say $\vec{p} = (p_1, p_2)$, in $\mathbf{G}$, and it is easy to see that if a and b are distinct, then $\vec{p}$ satisfies $p_1 \neq p_1 + p_2$. Hence, $[\![x_1]\!]_{\vec{p}} \neq [\![x_1]\!]_{\vec{p}} + [\![x_2]\!]_{\vec{p}}$ for all solutions $\vec{p}$ of $\mathcal{S}$ (cf. Lemma 8).

2. Let $\mathcal{S}_2$ be the recursive specification with respect to $\mathbf{G}$ consisting of the equations

$$x_1 \overset{\text{def}}{=} x_2\cdot x_1 \ \text{ , and}$$
$$x_2 \overset{\text{def}}{=} a.x_2 \ \text{ .}$$

Then $\mathcal{S}_2$ is guarded, for it is equivalent to the recursive specification consisting of the equations

$$x_1 \overset{\text{def}}{=} a.(x_2\cdot x_1) \ \text{ , and}$$
$$x_2 \overset{\text{def}}{=} a.x_2 \ \text{ .}$$

On the other hand, consider the recursive specification $\mathcal{S}_2{}'$ consisting of the equations

$$x_1 \overset{\text{def}}{=} x_2\cdot x_1 \ \text{ , and}$$
$$x_2 \overset{\text{def}}{=} a.x_2 + \mathbf{1} \ \text{ .}$$

Note that ${\mathcal{S}_2}'$ is equivalent to the recursive specification consisting of the equations

$$x_1 \overset{\text{def}}{=} a.(x_2 \cdot x_1) + x_1 \ , \text{ and}$$

$$x_2 \overset{\text{def}}{=} a.x_2 + \mathbf{1} \ .$$

The recursive specification ${\mathcal{S}_2}'$ is not guarded with respect to **G**.

3. Let $\mathcal{S}_3$ be the recursive specification with respect to **G** that consists of the equations

$$x_1 \overset{\text{def}}{=} x_2 \cdot x_1 + a.\mathbf{1} \ , \text{ and}$$

$$x_2 \overset{\text{def}}{=} x_2 + b.\mathbf{1} \ .$$

Let $\vec{p} = (p_1, p_2)$ be the unique solution of the guarded recursive specification ${\mathcal{S}_3}'$ consisting of the equations

$$y_1 = b.y_1 + a.\mathbf{1} \ , \text{ and}$$

$$y_2 = b.\mathbf{1} \ .$$

Then it is easy to see that $\vec{p}$ is also a solution of $\mathcal{S}$, and, moreover, if $a \neq b$, then $p_1 \neq p_1 + p_2$. Hence, $[\![x_1]\!]_{\vec{p}} \neq [\![x_1]\!]_{\vec{p}} + [\![x_2]\!]_{\vec{p}}$ for all solutions $\vec{p}$ of $\mathcal{S}$ (cf. Lemma 8).

The preceding example illustrates three complications when discussing unguardedness in a recursive specification containing sequential composition. The first complication is that unguarded occurrences of variables may become guarded after applying axioms for sequential composition. The second complication is that variables of which the defining equations have completely guarded right-hand sides may nevertheless fail to be guards themselves. The third complication is that unguarded occurrences of variables have, in general, trailing sequential components.

To circumvent the complication illustrated in Example 13.1, we shall first prove that every recursive specification $\mathcal{S}$ with respect to a basic process algebra can be transformed into a specification $\mathcal{S}'$ in Greibach Normal Form. For recursive specifications in Greibach Normal Form we can straightforwardly define when a variable has an unguarded occurrence in a term. Then, to deal with the complication illustrated in Example 13.2, we introduce the notion of transparent variable, which allows us to express when a variable fails to act as a guard. Finally, the complication illustrated in Example 13.3 is directly taken into account in the formulation of Lemma 17 below.

Let α be a finite sequence of variables. We define the *generalised sequential composition* $\underline{\alpha}$ associated with α inductively by $\underline{\alpha} = \mathbf{1}$ if α is the empty sequence, and $\underline{\alpha} = x \cdot \underline{\alpha'}$ if $\alpha = x\alpha'$.

Definition 14. A process term is in *Greibach Normal Form* if there exist $m, n \in \omega$, actions $a_1, \ldots, a_m$, and sequences of variables $\alpha_1, \ldots, \alpha_m$ and $\beta_1, \ldots, \beta_n$ such that

$$t = \sum_{1 \leq i \leq m} a_i.\underline{\alpha_i} + \sum_{1 \leq j \leq n} \underline{\beta_j}$$

(Here the empty summation denotes $\mathbf{0}$.) A recursive specification is in Greibach Normal Form if all right-hand sides of its equations are in Greibach Normal Form.

The transformation of an arbitrary specification $\mathcal{S}$ with respect to a basic process algebra into Greibach Normal Form introduces new variables, and therefore the resulting specification $\mathcal{S}'$ cannot be equivalent to $\mathcal{S}$. It will be the case, however, that every solution of $\mathcal{S}$ is a prefix of a solution of $\mathcal{S}'$, and, vice versa, a solution of $\mathcal{S}'$ has a prefix that is a solution of $\mathcal{S}$. Suppose that $\alpha \leq \beta$; we write $\mathcal{S} \preccurlyeq_{\mathbf{P}} \mathcal{S}'$ if every solution of $\mathcal{S}$ is a prefix of a solution of $\mathcal{S}'$, and every solution of $\mathcal{S}'$ has a solution of $\mathcal{S}$ as a prefix.

Proposition 15. *For every recursive specification $\mathcal{S}$ with respect to a basic process algebra* $\mathbf{P}$ *there exists a recursive specification $\mathcal{S}'$ with respect to* $\mathbf{P}$ *such that such that $\mathcal{S} \preccurlyeq_{\mathbf{P}} \mathcal{S}'$*

Proof. Consider an equation $x = t$ in $\mathcal{S}$. The following recursive procedure transforms t into Greibach Normal Form:

1. If t is a variable, $\mathbf{0}$, or $\mathbf{1}$, then the procedure terminates: t is in Greibach Normal Form.
2. If $t = a.t'$, for some process term t', then first recursively transform t' to Greibach Normal Form t'', then add an equation $y \overset{\text{def}}{=} t''$ (for y some fresh variable), and finally replace t by $a.y$.
3. If $t = t_1 \cdot t_2$, for some process terms t_1 and t_2, then first recursively transform t_1 and t_2 to Greibach Normal Forms t_1' and t_2', respectively. Then, add an equation $y \overset{\text{def}}{=} t''$ (for y some fresh variable). Clearly, $t = t_1' \cdot y$, and, assuming that $t_1' = \sum_{1\leq i\leq m} a_i.\underline{\alpha_i} + \sum_{1\leq j\leq n} \underline{\beta_j}$, by axiom S2, $t_1' \cdot y$ is equal to the Greibach Normal Form $\sum_{1\leq i\leq m} a_i.\underline{\alpha_i y} + \sum_{1\leq j\leq n} \underline{\beta_j y}$.
4. If $t = t_1 + t_2$, then first recursively transform t_1 and t_2 into Greibach Normal Forms t_1' and t_2', respectively. Then, by axioms A1–A4, the summands of $t_1' + t_2'$ can be rearranged in such a way that the result is in Greibach Normal Form.

Each of the steps described above preserves solutions, and that the procedure terminates can be established by well-founded induction. □

Let $\mathcal{S} = (x_\kappa \overset{\text{def}}{=} t_\kappa)_{\kappa<\alpha}$ be a recursive specification in Greibach Normal Form. We inductively define the sequence $(\mathcal{T}_i)_{i<\omega}$ of sets of variables by

(i) $\mathcal{T}_0 = \emptyset$;

(ii) if $x \overset{\text{def}}{=} \sum_{1\leq i\leq m} a_i.\underline{\alpha_i} + \sum_{1\leq j\leq n} \underline{\beta_j}$ is an equation in $\mathcal{S}$, and for some $1 \leq j \leq n$ the sequence β_j consists entirely of variables in $\mathcal{T}_i$, then $x \in \mathcal{T}_{i+1}$.

The set of *transparent* variables in $\mathcal{S}$ is the union $\bigcup_{i\in\omega} \mathcal{T}_i$.

Lemma 16. *Let $\mathcal{S}$ be a recursive specification with respect to a basic process algebra* $\mathbf{P}$*, and let α be a finite sequence of transparent variables in $\mathcal{S}$. Then, for all solutions $\vec{p}$ of $\mathcal{S}$ in* $\mathbf{P}$*, it holds that $[\![\underline{\alpha}]\!]_{\vec{p}} = [\![\underline{\alpha}]\!]_{\vec{p}} + \mathbf{1}$.*

Proof. It suffices to prove, for all $i \in \omega$, that if α is a sequence of variables in $\mathcal{T}_i$, then

$$[\![\underline{\alpha}]\!]_{\vec{p}} = [\![\underline{\alpha}]\!]_{\vec{p}} + \mathbf{1} \ . \tag{1}$$

We proceed by induction on i.

Clearly, if $i = 0$, then $\mathcal{T}_i = \emptyset$, so α is the empty sequence. Hence, $\underline{\alpha} = \mathbf{1}$, so (1) follows by axiom A3.

Suppose that, for all sequences β of variables in $\mathcal{T}_i$ it holds that $[\![\underline{\beta}]\!]_{\vec{p}} = [\![\underline{\beta}]\!]_{\vec{p}} + \mathbf{1}$ (the induction hypothesis). We consider an arbitrary sequence α of variables in $\mathcal{T}_{i+1}$, and do a subinduction on the length of α. Clearly, if α is the empty sequence, then, again, $\underline{\alpha} = \mathbf{1}$, so (1) follows by axiom A3. It remains to consider the case that α is non-empty, say $\alpha = x\alpha'$, assuming that $x \in \mathcal{T}_{i+1}$, and that α' is a sequence of variables in $\mathcal{T}_{i+1}$ for which (1) holds, i.e., $[\![\underline{\alpha'}]\!]_{\vec{p}} = [\![\underline{\alpha'}]\!]_{\vec{p}} + \mathbf{1}$ (the subinduction hypothesis). Let $x \overset{\text{def}}{=} \sum_{1\leq i\leq m} a_i.\underline{\alpha_i} + \sum_{1\leq j\leq n} \underline{\beta_j}$ in $\mathcal{S}$; that $x \in \mathcal{T}_{i+1}$ means that, for some $1 \leq j \leq n$, β_j is a sequence of variables over $\mathcal{T}_i$.

Hence, by the induction hypothesis, $[\![\underline{\beta_j}]\!]_{\vec{p}} = [\![\underline{\beta_j}]\!]_{\vec{p}} + \mathbf{1}$, so, by axioms A1–A3 it follows that $[\![x]\!]_{\vec{p}} = [\![x]\!]_{\vec{p}} + \mathbf{1}$. Hence, we now get, by axioms S1, S5,A3, and A2,

$$\begin{aligned}
[\![\underline{\alpha}]\!]_{\vec{p}} &= ([\![x]\!]_{\vec{p}} + \mathbf{1}) \cdot ([\![\underline{\alpha'}]\!]_{\vec{p}} + \mathbf{1}) \\
&= [\![x]\!]_{\vec{p}} \cdot ([\![\underline{\alpha'}]\!]_{\vec{p}} + \mathbf{1}) + \mathbf{1} \cdot ([\![\underline{\alpha'}]\!]_{\vec{p}} + \mathbf{1}) \\
&= ([\![x]\!]_{\vec{p}} \cdot ([\![\underline{\alpha'}]\!]_{\vec{p}} + \mathbf{1}) + \mathbf{1} \cdot ([\![\underline{\alpha'}]\!]_{\vec{p}} + \mathbf{1})) + \mathbf{1} \\
&= [\![\underline{\alpha}]\!]_{\vec{p}} + \mathbf{1}
\end{aligned}$$

□

Now, let $t = \sum_{1 \leq i \leq m} a_i \cdot \underline{\alpha_i} + \sum_{1 \leq j \leq n} \underline{\beta_j}$ be a right-hand side of $\mathcal{S}$. We say that x has an *unguarded* occurrence in t if, for some $1 \leq j \leq n$, there exist a sequence of transparent variables γ and a sequence of variables δ such that $\beta_j = \gamma x \delta$.

Lemma 17. *Let* $\mathbf{P}$ *be a basic process algebra and let* $\mathcal{S} = (x_\kappa \overset{\text{def}}{=} t_\kappa)_{\kappa<\alpha}$ *be a recursive specification in Greibach Normal Form with respect to* $\mathbf{P}$*. If* t_κ *has an unguarded occurrence of the variable* x_λ*, then there exists a term* t *such that* $[\![t_\kappa]\!]_{\vec{p}} = [\![t_\kappa]\!]_{\vec{p}} + [\![x_\lambda]\!]_{\vec{p}} \cdot [\![t]\!]_{\vec{p}}$ *for all solutions* $\vec{p}$ *of* $\mathcal{S}$*.*

Proof. Let $t_\kappa = \sum_{1 \leq i \leq m} a_i \cdot \underline{\alpha_i} + \sum_{1 \leq j \leq n} \underline{\beta_j}$, and suppose that x_λ has an unguarded occurrence in t_κ. Then there exists a sequence of transparent variables γ and a sequence of variables δ such that, for some $1 \leq j \leq n$, $\beta_j = \gamma x_\lambda \delta$. Since γ is a sequence of transparent variables, by Lemma 16 $[\![\underline{\gamma}]\!]_{\vec{p}} = [\![\underline{\gamma}]\!]_{\vec{p}} + \mathbf{1}$. Hence, by axioms S1 and S5,

$$\begin{aligned}
[\![\underline{\beta_j}]\!]_{\vec{p}} &= [\![\underline{\gamma x_\lambda \delta}]\!]_{\vec{p}} \\
&= ([\![\underline{\gamma}]\!]_{\vec{p}} + \mathbf{1}) \cdot [\![x_\lambda]\!]_{\vec{p}} \cdot [\![\underline{\delta}]\!]_{\vec{p}} \\
&= [\![\underline{\gamma x_\lambda \delta}]\!]_{\vec{p}} + \mathbf{1} \cdot [\![x_\lambda]\!]_{\vec{p}} \cdot [\![\underline{\delta}]\!]_{\vec{p}} \\
&= [\![\underline{\gamma x_\lambda \delta}]\!]_{\vec{p}} + [\![x_\lambda]\!]_{\vec{p}} \cdot [\![\underline{\delta}]\!]_{\vec{p}}
\end{aligned}$$

□

On the variables $(x_\kappa)_{\kappa<\alpha}$ of $\mathcal{S}$ we define a binary relation $\to$ by $x_\kappa \to x_\lambda$ if x_λ has an unguarded occurrence in t_κ.

Lemma 18. *Let* $\mathbf{P}$ *be a basic process algebra, and let* $\mathcal{S}$ *be a recursive specification with respect to* $\mathbf{P}$*. If* $\to_{\mathcal{S}}$ *is terminating, then* $\mathcal{S}$ *is guarded.*

Proof. Note that replacing an occurrence of a variable x_κ in a term t_λ by its definition t_κ, and then rewriting the result using axioms S1–S3, and S5 preserves solutions. Moreover, if $\to_{\mathcal{S}}$ is terminating and the axioms are only applied from left to right, then also the procedure that repeatedly replaces an unguarded occurrence of a variable by its definition terminates. It transforms $\mathcal{S}$ into an equivalent completely guarded recursive specification. □

In basic process algebra, an unguarded occurrence of a variable in an equation can be the first argument of a sequential composition. We cannot add an arbitrary summand to such a variable, but if we add a summand that mever terminates, then for such a summand, the second argument of the sequential composition can never be reached, and so can be ignored, so that the previous proof can be used. The crucial property we need is right-absorptiveness: an element p of a basic process algebra $\mathbf{P}$ is *right-absorptive* if $p \cdot q = p$ for all $q \in \mathbf{P}$.

Theorem 19. *Let* $\mathbf{P}$ *be a basic process algebra satisfying RDP. If for every countable subset* P *of* $\mathbf{P}$ *there exists a definable right-absorptive element* q *in* $\mathbf{P}$ *that is not a summand of any of the elements of* $\mathbf{P}$*, then every unguarded recursive specification with respect to* $\mathbf{P}$ *has infinitely many solutions.*

Proof. The proof is along the same lines as the proof of Theorem 10. Consider an arbitrary unguarded recursive specification $\mathcal{S}$, Then, by Lemma 18, there exists an infinite sequence $\mathcal{U} = (\kappa_i)_{i<\omega}$ such that $x_{\kappa_i} \to_{\mathcal{S}} x_{\kappa_{i+1}}$; such a sequence $\mathcal{U}$ we shall call an *unguardedness sequence* in $\mathcal{S}$.

and let $\mathcal{U} = (\kappa_i)_{i<\omega}$ be an unguardedness sequence in $\mathcal{S}$. We inductively define an infinite sequence of specifications $(\mathcal{S}_j)_{j<\omega}$ such that $\mathcal{S}_0 = \mathcal{S}$ and, for all $j < \omega$,

(i) $\mathcal{U}$ is an unguardedness sequence in $\mathcal{S}_j$;
(ii) $\mathcal{S}_j$ has a solution that differs at κ_i $(i < \omega)$ from every solution of $\mathcal{S}_{j+1}$; and
(iii) an initial segment of any solution of $\mathcal{S}_{j+1}$ is a solution of $\mathcal{S}_j$.

Suppose that $\mathcal{S}_j = (x_\kappa \overset{\text{def}}{=} t_\kappa)_{\kappa<\alpha}$. Since $\mathbf{P}$ satisfies RDP, $\mathcal{S}_j$ has a solution, say $\vec{p} = (p_\kappa)_{\kappa<\alpha}$. Let $P = \{p_{\kappa_i} \mid i < \omega\}$. By the assumption of the theorem, there exists a definable right-absorptive element q in $\mathbf{P}$ such that $p + q \neq p$ for all $p \in P$. Since q is definable, there exists a recursive specification, say $(y_\lambda \overset{\text{def}}{=} u_\lambda)_{\lambda<\beta}$, with a unique solution $(q_\lambda)_{\lambda<\beta}$ such that $q_0 = q$. We may assume without loss of generality that $x_\kappa \neq y_\lambda$ for all $\kappa < \alpha$ and $\lambda < \beta$.

For $\mu \geq \alpha$, we again denote by $\mu - \alpha$ the unique ordinal λ such that $\mu = \alpha + \lambda$. We prove that the specification $\mathcal{S}_{j+1} = (z_\mu \overset{\text{def}}{=} v_\mu)_{\mu<\alpha+\beta}$, with z_μ and v_μ defined by

$$z_\mu = \begin{cases} x_\mu & \text{if } \mu < \alpha \\ y_{\mu-\alpha} & \text{if } \mu \geq \alpha \end{cases} \quad \text{and} \quad v_\mu = \begin{cases} t_\mu & \text{if } \mu < \alpha \text{ and } \mu \notin \mathcal{U} \\ t_\mu + z_\alpha & \text{if } \mu < \alpha \text{ and } \mu \in \mathcal{U} \\ u_{\mu-\alpha} & \text{if } \mu \geq \alpha \end{cases} ,$$

satisfies the three requirements above. The arguments for the first two requirements are entirely analogous to the arguments for the first two requirements in the proof of Theorem 10, so we only consider the third requirement:

(iii) To prove that the initial segment $\vec{r} = (r_\kappa)_{\kappa<\alpha}$ of an arbitrary solution of $\mathcal{S}_{j+1}$ is a solution of $\mathcal{S}_j$, we need to show that $[\![t_\kappa]\!]_{\vec{r}} = r_\kappa$ for all $\kappa < \alpha$. We distinguish cases according to whether $\kappa \in \mathcal{U}$ or not. The case when $\kappa \notin \mathcal{U}$ is straightforward, since then $v_\kappa = t_\kappa$. So suppose that $\kappa \in \mathcal{U}$, and let λ be the $\mathcal{U}$ successor of κ; then

$$\begin{aligned}
[\![t_\kappa]\!]_{\vec{r}} &= [\![t_\kappa]\!]_{\vec{r}} + r_\lambda \cdot [\![t]\!]_{\vec{r}} && \text{(Lemma 17)} \\
&= [\![t_\kappa]\!]_{\vec{r}} + (r_\lambda + r_\alpha) \cdot [\![t]\!]_{\vec{r}} \\
&= [\![t_\kappa]\!]_{\vec{r}} + (p_\lambda \cdot [\![t]\!]_{\vec{r}} + r_\alpha \cdot [\![t]\!]_{\vec{r}}) && \text{(S1)} \\
&= [\![t_\kappa]\!]_{\vec{r}} + (r_\lambda \cdot [\![t]\!]_{\vec{r}} + r_\alpha) && (r_\alpha \text{ is right-absorptive)} \\
&= ([\![t_\kappa]\!]_{\vec{r}} + p_\lambda \cdot [\![t]\!]_{\vec{r}}) + r_\alpha && \text{(A2)} \\
&= [\![t_\kappa]\!]_{\vec{r}} + r_\alpha && \text{(Lemma 17)} \\
&= [\![v_\kappa]\!]_{\vec{r}} && (\text{definition of } v_\kappa) \\
&= r_\kappa \ .
\end{aligned}$$

Similarly as in the proof of Theorem 10 we can now associate with $(\mathcal{S}_j)_{j<\omega}$ an infinite sequence of solutions of $\mathcal{S}$ that are all pairwise distinct. □

By the BPA$_{\mathbf{01}}$-*reduct* of a process algebra $\mathbf{P}$ we mean the algebra obtained from $\mathbf{P}$ by forgetting all process-theoretic operations except $\mathbf{0}$, $\mathbf{1}$, $a.$ $(a \in \mathcal{A})$, $+$ and $\cdot$. We apply the above theorem to the BPA$_{\mathbf{01}}$-reducts of $\mathbf{I}^\infty_{\aleph_1}$ and $\mathbf{G}$ in order to obtain the following corollary.

Corollary 20. *With respect to the* BPA$_{\mathbf{01}}$*-reducts of* $\mathbf{G}$ *and* $\mathbf{I}^\infty$ *all unguarded recursive specifications have infinitely many solutions.*

Proof. The proof of this corollary to Theorem 19 is analogous to the proof of Corollary 12. If suffices to note that all the processes in the uncountable set of definable processes in $\mathbf{G}$ and $\mathbf{I}^\infty_{\aleph_1}$ given by Proposition 7 are actually right-absorptive. □

5. Process algebra PA

We call an element p of a process algebra $\mathbf{P}$ *merge-saturated* iff for all processes $q \in \mathbf{P}$ it holds that $p = q \parallel p$. In the process algebras $\mathbf{G}$, $\mathbf{I}^\infty$ and $\mathbf{I}^\infty_{\aleph_1}$, a merge-saturated element χ can be defined by the following guarded equation

$$x \stackrel{\text{def}}{=} \sum_{a \in \mathcal{A}} a.x \ \ .$$

(Note that, in case the set of actions is infinite, such a *guarded* specification cannot be given, although we can still give an *unguarded* countable specification having equations $x_n \stackrel{\text{def}}{=} a_n.x_0 + x_{n+1}$ for each natural number n.) This process χ is merge-saturated, see Remark 2.5.2 of [3].

This means that the unguarded specification consisting of the defining specification for a merge-saturated process together with the equation $y \stackrel{\text{def}}{=} y \parallel x$, has the solution consisting of this merge-saturated process for y. Moreover, this is the unique solution of this specification, as the process is merge-saturated. Thus, we have the following theorem.

Theorem 21. *If* $\mathbf{P}$ *is a model of* PA *or* PA$_{\mathbf{01}}$ *with a merge-saturated definable element, and* $\mathbf{P}$ *satisfies RSP, then the unguarded recursive specification consisting of the defining specification of this element together with*

$$y \stackrel{\text{def}}{=} y \parallel x$$

has a unique solution.

Next, we consider the unguarded recursive specification consisting of the defining equations for the merge-saturated element together with the unguarded equation $y \stackrel{\text{def}}{=} y \mathbin{⫗} x$. Obviously, the merge-saturated element is again a solution of this specification. In PA$_{\mathbf{01}}$, however, also $\mathbf{0}$ is a solution. In fact, since the equation $y \stackrel{\text{def}}{=} y \mathbin{⫗} x$ leaves the first step of y undetermined, we can take any subset of actions B for this first step. Hence, for every $B \subseteq \mathcal{A}$, the two-element sequence

$$(\chi, \sum_{a \in B} a.\chi)$$

is a solution (here, χ is the merge-saturated element). Since, after the first step, the mergend χ again saturates all following steps, the only solutions are the abovementioned. This means this specification has $2^{|\mathcal{A}|}$ solutions in PA$_{\mathbf{01}}$, and one less in PA (not for the empty set of initial actions). So we see that with a form of parallel composition, an unguarded specification can have a different number of solutions. Of course, it will still be the case that most unguarded specifications have infinitely many solutions. For example, every process is a solution of

$$x \stackrel{\text{def}}{=} x \mathbin{⫗} \mathbf{1} \ \ ,$$

and every right-absorptive process is a solution of

$$x \stackrel{\text{def}}{=} x \mathbin{⫗} \mathbf{0} \ \ .$$

6. Conclusion

Usually, unguarded recursive specifications have infinitely many solutions. This is the case in minimal process algebra and basic process algebra (in models where every recursive specification has a solution). However, when parallel composition is added, we can give unguarded specifications with a finite number of solutions, and even an unguarded specification with a unique solution.

At each point, we tried to state our theorems in the most general setting. We believe the results cannot be generalized further, to obtain e.g. theorems in universal algebra.

References

[1] J. C. M. Baeten, T. Basten, and M. A. Reniers. *Process Algebra: Equational Theories of Communicating Processes.* Number 50 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2010.

[2] J. C. M. Baeten, J. A. Bergstra, and J. W. Klop. On the Consistency of Koomen's Fair Abstraction Rule. *Theoretical Computer Science*, 51(1/2):129–176, 1987.

[3] J. A. Bergstra and J. W. Klop. Fixed Point Semantics in Process Algebra. Technical Report IW 206, Mathematical Centre, Amsterdam, the Netherlands, 1982.

[4] J. A. Bergstra and J. W. Klop. A Convergence Theorem in Process Algebra. In J. W. de Bakker and J. J. M. M. Rutten, editors, *Ten Years of Concurrency Semantics*, pages 164–195. World Scientific, Singapore, 1992.

[5] N. Bourbaki. *Elements of Mathematics: Algebra 1.* Springer Verlag, 1989.

[6] Gordon D. Plotkin. A structural approach to operational semantics. *J. Log. Algebr. Program.*, 60-61:17–139, 2004.