Regularity of BPA-Systems is Decidable

Sjouke Mauw and Hans Mulder

Dept. of Mathematics and Computing Science,
Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands.
sjouke@win.tue.nl, hansm@win.tue.nl

Abstract. It is decidable whether a system in Basic Process Algebra (BPA) is regular with respect to bisimulation semantics. Basic operators in BPA are alternative composition, sequential composition and guarded recursion. A system is regular if the interpretations of all process variables defined in the system have finitely many states. We present an effective method to transform a BPA specification into a linear specification whenever possible.

1 Introduction

An important issue in automatic verification of concurrent systems using process algebra is extending the techniques to systems with an infinite state space. The simplest extension of regular specifications is BPA (Basic Process Algebra [3]), which has operators for alternative and sequential composition and allows for the construction of infinite processes by means of guarded recursion. The languages generated by BPA specifications are exactly the context-free languages. However, we will not study language equivalence, but bisimulation equivalence ([9]) which is considered more appropriate for verification purposes.

It has already been shown that bisimulation equivalence is decidable for BPA processes ([1, 4]). An open problem was the question whether it is decidable if a BPA specification is regular (i.e. whether it can be interpreted as a graph which has finitely many states). If so, this would enable the application of the well known algorithms for regular systems to those BPA specifications which are in fact regular. This would help in deciding exactly when to use existing efficient implementations for deciding bisimulation equivalence (for example for the PSF-Toolkit [7]).

In this paper we prove that it is decidable whether a BPA system is regular, that is, all process variables defined by it are regular. Some results for similar specification languages are known. Weakening BPA by replacing the general multiplication by action prefix multiplication will only allow the description of regular systems, while extending BPA with the communication merge to ACP (the algebra of communicating processes, [3]) yields a language in which regularity is not decidable. It is also known that regularity of BPA systems modulo language equivalence is not decidable.

The basic observation in this paper is that if a process is not regular this is caused by stacking a tail of processes: consider

$$X = aXb + c$$

This process can execute a and then enter state Xb. From this state the process can do an a again and enter state Xbb. Executing more a steps leads to infinitely many different states.

In this paper we will formulate the conditions under which this stacking leads to an irregular process. Furthermore, we give a method to generate a linear specification if the BPA specification is known to be regular.

This paper is built up as follows. In Sect. 2 we introduce BPA and its interpretation in the graph model. Normed processes and the reachability relation play an important role in the decision procedure. They are defined in Sect. 3. Section 4 contains the main theorem of this paper and in Sect. 5 we give a linearization procedure.

Thanks are due to Felix Croes who was of great help while developing the ideas in this paper and to Michel Reniers for proofreading.

2 Basic Process Algebra

In this section we will give the signature of the theory BPA and its interpretation in the graph model. For a more detailed treatment we refer to [2].

2.1 Specifications

We consider a finite set of atomic actions A. Typical elements of A are a, b, ... Let V be a countably infinite set of process variables. Typical elements of V are $X, Y, \ldots$. A BPA term (over A and V) is a term constructed from A, V and the operators $\cdot$ and $+$ (sequential and alternative composition). An equation is an expression of the form $X = t$, where $X \in V$ and t is a BPA term. An equation $X = t$ is guarded if every occurrence of a variable in t is in the sub-term q of some sub-term $p \cdot q$ of t (p, q BPA terms). A (BPA) specification is a finite collection of guarded equations, such that all variables occurring in the right-hand side of an equation occur in the left-hand side of some equation and no variable occurs in the left-hand side of two equations. The collection of variables occurring in specification S is denoted by V_S. If X is a variable defined in S, then $Def_S(X)$ is the right-hand side of the equation of which X is the left-hand side.

If V is a collection of variables, then V^* denotes the set of all finite sequences over V. The empty sequence is denoted by λ. The length $|\sigma|$ of a sequence σ is defined in the usual way. Greek letters σ, ρ, ... range over V^*. Every non-empty sequence of variables can be considered as a BPA term by inserting the sequential composition operator. If the meaning of a sequence is clear from the context, we will not explicitly apply conversion functions from sequences to BPA terms and back. Furthermore if p is a BPA term, the expressions $p \cdot \lambda$ and $p\lambda$ are interpreted as the BPA term p. Concatenation of sequences σ and ρ is denoted by $\sigma\rho$. We have $\lambda\sigma = \sigma\lambda = \sigma$.

A specification is in Greibach Normal Form (GNF) if the right-hand sides of all equations have the form $a_0 \cdot \sigma_0 + \ldots + a_n \cdot \sigma_n$ $(n \geq 0)$. Note that σ_i may be λ so $a_i \cdot \sigma_i = a_i$. Given an equation $X = a_0 \cdot \sigma_0 + \ldots + a_n \cdot \sigma_n$, we say that $a_i \cdot \sigma_i$ $(0 \leq i \leq n)$ is a summand of X, notation $a_i \cdot \sigma_i \subset X$.

In [1] it is shown that every BPA specification can be transformed into a specification in GNF. Therefore we can restrict ourselves to specifications in GNF.

A specification is linear if every summand of every equation in the specification has the form a or $a \cdot X$.

2.2 The Graph Model

We will interpret BPA specifications in the so called graph model. This model consists of finitely branching rooted graphs with labeled edges. This means that one node is marked as the root and that every edge has a label from a given set A. A node is a termination node if it has no outgoing edges. A node is also called a state and an edge a transition. If there is an edge with label a from node s to node t, we denote this by $s \xrightarrow{a} t$, or simply $s \rightarrow t$. If there is a sequence of edges $s_0 \xrightarrow{a_0} s_1 \ldots \xrightarrow{a_{n-1}} s_n$ $(n > 0)$ then we write $s_0 \overset{a_0 \ldots a_{n-1}}{\twoheadrightarrow} s_n$, or simply $s_0 \twoheadrightarrow s_n$.

Definition 1. A relation R between the nodes of two graphs g and h is a bisimulation if the following holds.

- If $s \xrightarrow{a} s'$ is an edge in g and $R(s,t)$, then there is an edge $t \xrightarrow{a} t'$ in h such that $R(s',t')$.
- If $t \xrightarrow{a} t'$ is an edge in h and $R(s,t)$, then there is an edge $s \xrightarrow{a} s'$ in g such that $R(s',t')$.

Two graphs g and h are bisimilar, notation $g \leftrightarrow h$, if there is a bisimulation relating the roots of g and h.

The collection of graphs divided out by bisimulation equivalence is denoted by $G/\leftrightarrow$. This is a model of BPA. The notion of bisimulation can easily be extended to nodes from the same graph. For details see [2].

A specification in GNF is interpreted in $G/\leftrightarrow$ in the following way.

Definition 2. Let S be a specification and $\sigma \in V_S^*$, then $gr_S(\sigma)$ is the graph with nodes V_S^*, root node σ and edges $\{X\xi \xrightarrow{a} \rho\xi \mid a \in A,\ X \in V_S,\ \rho, \xi \in V_S^*, a\rho \subset X\}$

From this definition it follows that λ is the only termination node. This construction is equivalent to the standard interpretation of BPA terms in the graph model. The above definition satisfies our needs in the easiest way. For $\sigma, \sigma' \in V_S^*$ we say that σ and σ' are bisimilar, notation $\sigma \leftrightarrow \sigma'$, if $gr_S(\sigma)$ and $gr_S(\sigma')$ are bisimilar.

Proposition 3. *Let S be a specification and $\sigma, \sigma', \rho \in V_S^*$*

1. $\sigma \twoheadrightarrow \sigma' \Rightarrow \sigma\rho \twoheadrightarrow \sigma'\rho$
2. $\sigma \neq \lambda \wedge \rho \neq \lambda \wedge \sigma\rho \twoheadrightarrow \lambda \;\Rightarrow\; \sigma \twoheadrightarrow \lambda \wedge \rho \twoheadrightarrow \lambda$

Proof. 1. From the definition of the edges we infer $\sigma \xrightarrow{a} \sigma' \Rightarrow \sigma\rho \xrightarrow{a} \sigma'\rho$, which can be generalized using induction on the number of transitions.
2. Proof by induction on the number of transitions in the sequence $\sigma\rho \twoheadrightarrow \lambda$. If $\sigma\rho \rightarrow \lambda$ in one step, then either σ or ρ equals λ. If $\sigma\rho \rightarrow \sigma_1 \ldots \rightarrow \lambda$ in $n+1$ steps, either $\sigma = \lambda$, in which case the implication is trivially true, or σ is of the form $X\xi$, where X has a summand $a\eta$ and $\sigma_1 = \eta\xi\rho$. Now there are two cases. The first case is $\eta\xi = \lambda$. Then $\sigma \rightarrow \lambda$ and $\rho = \sigma_1 \twoheadrightarrow \lambda$ in n transitions. The second case is $\eta\xi \neq \lambda$, then we can apply the induction hypothesis to $\eta\xi$ and ρ. □

Definition 4. A graph is regular if it is bisimilar to a graph with a finite set of nodes. Let S be a specification and $X \in V_S$, then X is regular if $gr_S(X)$ is regular. A specification is regular if all variables in V_S are regular.

Two alternative characterizations of regularity follow directly from the definition.

Proposition 5. (i) *A graph is regular if and only if there is no infinite sequence* $s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$ *such that* $s_i \not\leftrightarrow s_j$ *for* $i \neq j$, *where* s_0 *is the root of the graph.* (ii) *A graph is regular if and only if there is no infinite sequence* $s_0 \twoheadrightarrow s_1 \twoheadrightarrow s_2 \twoheadrightarrow \ldots$ *such that* $s_i \not\leftrightarrow s_j$ *for* $i \neq j$, *where* s_0 *is the root of the graph.*

Refer to [8] for a proof of the following proposition, which gives a correspondence between regular and linear specifications.

Proposition 6. *A specification* S *is regular if and only if there is a linear specification* T *such that* $V_S \subset V_T$ *and for all* $X \in V_S$ $gr_S(X) \leftrightarrow gr_T(X)$.

3 Normed Processes and the Reachability Relation

3.1 Normed Processes

A weakly normed process (or normed process for short) is a process which may terminate in a finite number of steps.[1]

Definition 7. A node s in a graph is normed, notation $s{\downarrow}$, if s is a termination node, or there is a termination node t such that $s \twoheadrightarrow t$. A node that is not normed is called (strongly) perpetual[2], notation $s{\Uparrow}$. A graph is normed if its root node is normed. If S is a specification and $\sigma \in V_S^*$ then we say that σ is normed if $gr_S(\sigma)$ is normed.

[1] A strongly normed process is a process which may terminate at any point during its execution. We will not use this notion in this paper.
[2] A process is called weakly perpetual if it is not strongly normed.

Proposition 8. *Let S be a specification and $\sigma, \rho \in V_S^*$ then*

$$\sigma\rho\downarrow \Leftrightarrow \sigma\downarrow \wedge \rho\downarrow$$

Proof. If σ or ρ is a termination node, and thus equal to λ, the proposition is clearly true. Now suppose $\sigma \twoheadrightarrow \lambda$ and $\rho \twoheadrightarrow \lambda$, then use Proposition 3.1 to derive $\sigma\rho \twoheadrightarrow \lambda$. For proving the other implication, suppose $\sigma\rho \twoheadrightarrow \lambda$ then we can use Proposition 3.2 to derive $\sigma \twoheadrightarrow \lambda$ and $\rho \twoheadrightarrow \lambda$. □

Proposition 9. *Let $\sigma, \rho \in V_S^*$ such that $\sigma\Uparrow$, then $gr_S(\sigma\rho) \leftrightarrow gr_S(\sigma)$.*

Proof. Construct a bisimulation by relating $\eta\xi$ to η for all $\eta, \xi \in V_S^*$ for which η is perpetual. □

In a given state, we can count the minimal number of transitions needed to terminate. This is called the norm of the state.

Definition 10. The norm of a node s is inductively defined by

$$norm(s) = \begin{cases} \infty & \text{if } s\Uparrow \\ 0 & \text{if } s\downarrow \text{ and } s \text{ is a termination node} \\ 1 + \min\{norm(t) | s \to t\} & \text{if } s\downarrow \text{ and } s \text{ is not a termination node} \end{cases}$$

Proposition 11. *For all nodes s and t $s \leftrightarrow t$ implies $norm(s) = norm(t)$.*

Proof. If both s and t are perpetual, then it is clear. Because s and t are bisimilar, it is impossible that s is perpetual and t is normed or vice versa. If s and t are normed, use induction on the norm. □

Given a specification S we can calculate the normed variables in the following way. Define a sequence of sets of variables N_i inductively by

$$\begin{aligned} N_0 &= \emptyset \\ N_{i+1} &= N_i \cup \{X \mid \exists_{a\sigma \subset X, \sigma \in V_S^*} \forall_{Y \in \sigma}\ Y \in N_i\} \end{aligned}$$

Now set $N = \cup_{i \geq 0} N_i$ then N can be computed effectively.

Theorem 12. *The set N contains exactly all normed variables of V_S. There is some $i \geq 0$ such that $N_i = N_{i+1}$ and for this value $N = N_i$.*

Proof. It is clear from the construction that N contains only normed variables. In order to see that N contains all normed variables, we suppose that X is the variable such that $X\downarrow$, $X \notin N$ and $X \twoheadrightarrow \lambda$ with a minimal number of transitions. We consider two cases. If $X \to \lambda$, then X has a summand a for some $a \in A$ and thus $X \in N_1$, which is a contradiction. If $X \to X_0 \ldots X_n \twoheadrightarrow \lambda$, then $(X_0 \ldots X_n)\downarrow$ and thus $X_0 \twoheadrightarrow \lambda, \ldots X_n \twoheadrightarrow \lambda$. These variables all need at least one less transition to reach λ than X, so they are elements of N. But by the definition of N this would imply that $X \in N$, which again gives a contradiction.

Finally, since the N_i are an increasing sequence of subsets of V_S and V_S is finite, there are only finitely many different sets N_i and therefore there exists an i such that $N_i = N_{i+1}$, which implies that $N_{i+k} = N_i$ for all k. □

If we define $n + \infty = \infty + n = \infty$ we have the following proposition.

Proposition 13. *For* $\sigma, \rho \in V_S^*$ $norm(\sigma\rho) = norm(\sigma) + norm(\rho)$

Proof. Induction on the length of σ. If $\sigma = \lambda$ or $\rho = \lambda$ then it is clearly true. If $\sigma = X\sigma'$ then $norm(X\sigma'\rho) = norm(X) + norm(\sigma'\rho) = norm(X) + norm(\sigma') + norm(\rho) = norm(X\sigma') + norm(\rho)$. The first equality can be proven with induction on the norm of X. □

Proposition 14. *Let* $X \in V_S$, $\sigma, \rho \in V_S^*$, *then*

$$\sigma\downarrow \wedge X \twoheadrightarrow \sigma\rho \Rightarrow X \twoheadrightarrow \rho$$

Proof. From $\sigma\downarrow$ we derive $\sigma = \lambda$ or $\sigma \twoheadrightarrow \lambda$. In the first case the proposition is trivially true, In the second case we can use Proposition 3.1 to get $\sigma\rho \twoheadrightarrow \lambda\rho$ and by transitivity $X \twoheadrightarrow \rho$. □

3.2 Reachability

The reachability relation $X \overset{\sigma}{\hookrightarrow} Y$ expresses that variable X can become variable Y after executing a number of transitions. The sequence σ is stacked after Y and will be executed upon termination of Y.

Definition 15. Let S be a specification, then we define the reachability relation $\hookrightarrow_S$ on $V_S \times V_S^* \times V_S$ for all $X, Y \in V_S, \sigma \in V_S^*$ by

$$X \overset{\sigma}{\hookrightarrow}_S Y \Leftrightarrow \exists_{\rho \in V_S^*, a \in A}\ a\rho Y\sigma \subset X \wedge \rho\downarrow$$

We will write $\hookrightarrow$ instead of $\hookrightarrow_S$ if S is known from the context.

Definition 16. Consider for $n > 0$ the reachability sequence $X_0 \overset{\sigma_0}{\hookrightarrow} X_1 \overset{\sigma_1}{\hookrightarrow} \ldots \overset{\sigma_{n-1}}{\hookrightarrow} X_n$, then we define the following properties

1. The sequence is normed if $(X_n\sigma_{n-1} \ldots \sigma_0)\downarrow$.
2. The sequence is a cycle if $X_0 = X_n$.
3. A cycle is minimal if $\forall_{0 \le i < j \le n} X_i = X_j \Rightarrow i = 0 \wedge j = n$
4. The sequence is stacking if $\sigma_{n-1} \ldots \sigma_0 \neq \lambda$.

Since V_S is finite, we can consider $(V_S, \hookrightarrow_S)$ as a finite graph and thus we have the following proposition.

Proposition 17. *For a given specification* S, $\hookrightarrow_S$ *has finitely many minimal cycles.*

Reachability implies a transition relation:

Proposition 18. *For all* $X, Y \in V_S, \sigma \in V_S^*$

$$X \overset{\sigma}{\hookrightarrow} Y \Rightarrow X \twoheadrightarrow Y\sigma$$

Proof. Suppose $X \overset{\sigma}{\hookrightarrow} Y$, then $a\rho Y\sigma \subset X$ which gives $X \overset{a}{\rightarrow} \rho Y\sigma$. Since ρ is normed, we can use Proposition 14 and conclude $X \twoheadrightarrow Y\sigma$. □

Corollary 19. *If* $X_0 \overset{\sigma_0}{\hookrightarrow} \ldots \overset{\sigma_{n-1}}{\hookrightarrow} X_n$ $(n > 0)$ *is a reachability sequence, then* $X_0 \twoheadrightarrow X_n\sigma_{n-1} \ldots \sigma_0$.

4 Deciding Regularity

Theorem 20. *A specification S is regular if and only if $\hookrightarrow_S$ has no normed stacking minimal cycles.*

Proof. First we prove the "only if" part by contradiction. Suppose that $\hookrightarrow_S$ has a normed stacking cycle $X \overset{\sigma_0}{\hookrightarrow} \dots \overset{\sigma_{n-1}}{\hookrightarrow} X$ $(n > 0)$, then from Corollary 19 we conclude $X \twoheadrightarrow X\rho$, where $\rho = \sigma_{n-1} \dots \sigma_0$. Since the cycle is stacking, $\rho \neq \lambda$. Using Proposition 3.1 we can construct a sequence

$$X \twoheadrightarrow X\rho \twoheadrightarrow X\rho\rho \twoheadrightarrow X\rho\rho\rho \twoheadrightarrow \dots .$$

We calculate the norm of each state using Proposition 13:

$$norm(X\rho^i) = norm(X) + i \cdot norm(\rho)$$

The cycle under consideration is normed, therefore $(X\rho)\downarrow$, therefore $X\downarrow$ and $\rho\downarrow$ (Proposition 8). In other words, $norm(X) < \infty$ and $norm(\rho) < \infty$. Moreover, the cycle is stacking, hence $norm(\rho) > 0$. Consequently, for $i \neq j$, $norm(X\rho^i) \neq norm(X\rho^j)$. Using the fact that bisimulation respects the norm (Proposition 11) we have $X\rho^i \not\leftrightarrow X\rho^j$ and thus S is not regular (Proposition 5.ii).

The "if" part of the proof is more elaborate. Assuming that some $X \in V_S$ is not regular, we derive a contradiction. By Proposition 5.i there exists an infinite sequence (setting $\sigma_0 = X$)

$$\sigma_0 \rightarrow \sigma_1 \rightarrow \sigma_2 \rightarrow \dots$$

such that $\sigma_i \not\leftrightarrow \sigma_j$ for $i, j \geq 0$, $i \neq j$. From the absence of normed stacking cycles, we will derive the existence of i and j $(i \neq j)$ such that $\sigma_i \leftrightarrow \sigma_j$ and thus we will have a contradiction.

The first step is to make the relation between the individual variables from σ_i and σ_{i+1} explicit. For this purpose, we will consider the infinite sequence as a directed tree with labeled nodes and unlabeled edges. For every variable in σ_i $(i \geq 0)$, we create a node. This node is related to all reachable successors (if any) of this variable in σ_{i+1}. Formally:

Definition 21. For every $i \geq 0$ we have nodes $\langle i, 0\rangle, \dots, \langle i, |\sigma_i| - 1\rangle$. The label $L(\langle i, k\rangle)$ of node $\langle i, k\rangle$ is the kth variable of σ_i (if we start counting at 0). An edge from node $\langle i, p\rangle$ to node $\langle i+1, p'\rangle$ is denoted by $\langle i, p\rangle \rightsquigarrow \langle i+1, p'\rangle$. The edges are defined as follows. Let $i \geq 0$ and $\sigma_i = X_0 \dots X_k$ $(k \geq 0)$ then, following Definition 2, the transition $\sigma_i \rightarrow \sigma_{i+1}$ is due to a summand $a\rho \subset X_0$. Now we consider two cases.

1. $|\rho| = 0$ and thus $\rho = \lambda$. Then $\sigma_{i+1} = X_1 \dots X_k$. For $1 \leq p \leq k$ we define edges from $\langle i, p\rangle$ to $\langle i+1, p-1\rangle$.
2. $|\rho| > 0$ and thus $\rho = Y_0 \dots Y_h$ $(h \geq 0)$. Then $\sigma_{i+1} = Y_0 \dots Y_h X_1 \dots X_k$. There are two sub-cases.
   (a) If $\rho\downarrow$ then we define edges from $\langle i, 0\rangle$ to all nodes $\langle i+1, 0\rangle, \dots, \langle i+1, h\rangle$.

(b) If $\rho\Uparrow$ then there is an $m \geq 0$ such that $Y_m\Uparrow$ and $Y_0 \ldots Y_{m-1}\downarrow$. Then we define edges from $\langle i, 0\rangle$ only to the nodes $\langle i+1, 0\rangle, \ldots, \langle i+1, m\rangle$.

Moreover in both sub-cases we define edges from $\langle i, p\rangle$ to $\langle i+1, h+p\rangle$ for $1 \leq p \leq k$.

This construction implies that the sequence of labels at level i, namely $L(\langle i, 0\rangle) \ldots L(\langle i, |\sigma_i| - 1\rangle)$ is exactly σ_i. Furthermore, a node $\langle i, p\rangle$ has exactly one successor if $p > 0$, while if $p = 0$ then $\langle i, p\rangle$ may have more than one successor or none at all.

Example 1. Figure 1 below shows a specification and a fragment of the graph corresponding to the sequence

$$S \rightarrow TUV \rightarrow WXYUV \rightarrow XYUV \rightarrow \ldots.$$

A node $\langle i, k\rangle$ is represented by its label $L(\langle i, k\rangle)$ appearing as the kth letter on the ith line in the figure (counting from 0); arrows denote $\rightsquigarrow$ relationships. Note that the graph is almost a tree; it would be a tree if there were an edge from $\langle 1, 0\rangle$ (with label T) to $\langle 2, 2\rangle$ (with label Y). This edge is omitted because X is perpetual.

$$\begin{aligned}
S &= aTUV \\
T &= bWXY + b \\
U &= c \\
V &= d \\
W &= e \\
X &= fZX \\
Y &= g \\
Z &= h
\end{aligned}$$

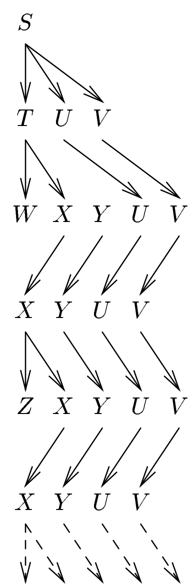

Fig. 1. Sample specification and successor graph

Before completing the proof of Theorem 20, we will formulate a few lemmas relating $\rightsquigarrow$ to $\hookrightarrow$.

Lemma 22. *If there is an edge* $\langle i, p\rangle \rightsquigarrow \langle i+1, q\rangle$ *such that* $p > 0$, *then* $L(\langle i, p\rangle) \ldots L(\langle i, |\sigma_i| - 1\rangle) = L(\langle i+1, q\rangle) \ldots L(\langle i+1, |\sigma_{i+1}| - 1\rangle)$

Proof. This follows directly from the definition. □

Lemma 23. *Let* $n > 0$*,* $i \geq 0$ *and* $\langle i, 0\rangle \rightsquigarrow \langle i+1, p_1\rangle \rightsquigarrow \ldots \rightsquigarrow \langle i+n, p_n\rangle$ *be a sequence of edges such that* $p_1 > 0, \ldots, p_{n-1} > 0$*, then there exists* $\rho \in V_S^*$ *such that for all* $0 < k \leq n$*,* $L(\langle i, 0\rangle) \overset{\rho}{\hookrightarrow} L(\langle i+k, p_k\rangle)$ *and* $L(\langle i+k, p_k+1\rangle)$ $L(\langle i+k, p_k+2\rangle) \ldots L(\langle i+k, |\sigma_{i+k}|-1\rangle)$ *is equal to* $\rho L(\langle i, 1\rangle) \ldots L(\langle i, |\sigma_i|-1\rangle)$*.*

Proof. Induction on n.

If $n = 1$ then we consider the edge $\langle i, 0\rangle \rightsquigarrow \langle i+1, p_1\rangle$. Then we have $\sigma_i = X_0 \ldots X_k$, $\sigma_{i+1} = Y_0 \ldots Y_h X_1 \ldots X_k$ such that $0 \leq p_1 \leq h$ and $aY_0 \ldots Y_h \subset X_0$. Because $Y_0 \ldots Y_{p_1-1}\downarrow$, this gives a reachability step $X_0 \overset{Y_{p_1+1}\ldots Y_h}{\hookrightarrow} Y_{p_1}$, and thus $L(\langle i, 0\rangle) \overset{Y_{p_1+1}\ldots Y_h}{\hookrightarrow} L(\langle i+1, p_1\rangle)$. If we take $\rho = Y_{p_1+1} \ldots Y_h$ then $L(\langle i+1, p_1+1\rangle)$ $L(\langle i+1, p_1+2\rangle) \ldots L(\langle i+1, |\sigma_{i+1}|-1\rangle)$ is equal to $\rho L(\langle i, 1\rangle) \ldots L(\langle i, |\sigma_i|-1\rangle)$, because both sequences are $Y_{p_1+1} \ldots Y_h X_1 \ldots X_k$.

If $n = m+1$, then by the induction hypothesis there is a ρ such that for all $0 < k \leq m$, $L(\langle i, 0\rangle) \overset{\rho}{\hookrightarrow} L(\langle i+k, p_k\rangle)$ and $L(\langle i+k, p_k+1\rangle) L(\langle i+k, p_k+2\rangle)$ $\ldots L(\langle i+k, |\sigma_{i+k}|-1\rangle)$ is equal to $\rho L(\langle i, 1\rangle) \ldots L(\langle i, |\sigma_i|-1\rangle)$. We have an edge $\langle i+m, p_m\rangle \rightsquigarrow \langle i+m+1, p_{m+1}\rangle$, and $p_m \neq 0$, so Lemma 22 applies. This yields

$$\begin{aligned}
&L(\langle i+m+1, p_{m+1}\rangle) \ldots L(\langle i+m+1, |\sigma_{i+m+1}|-1\rangle) = \\
&L(\langle i+m, p_m\rangle) L(\langle i+m, p_m+1\rangle) \ldots L(\langle i+m, |\sigma_{i+m}|-1\rangle) = \\
&L(\langle i+m, p_m\rangle) \rho L(\langle i, 1\rangle) \ldots L(\langle i, |\sigma_i|-1\rangle)
\end{aligned}$$

The second equality follows from the induction hypothesis with k substituted by m.

The first equality implies that $L(\langle i+m+1, p_{m+1}\rangle) = L(\langle i+m, p_m\rangle)$. By the induction hypothesis $L(\langle i, 0\rangle) \overset{\rho}{\hookrightarrow} L(\langle i+m, p_m\rangle)$ and thus $L(\langle i, 0\rangle) \overset{\rho}{\hookrightarrow} L(\langle i+m+1, p_{m+1}\rangle)$. □

Corollary 24. *Let* $n > 0$*,* $i \geq 0$ *and* $\langle i, 0\rangle \rightsquigarrow \langle i+1, p_1\rangle \rightsquigarrow \ldots \rightsquigarrow \langle i+n, p_n\rangle$ *be a sequence of edges such that* $p_v = 0$ *only for the values* $v_0, \ldots v_q$ *of* v*, then there exists a reachability sequence* $L(\langle i, 0\rangle) \overset{\rho_0}{\hookrightarrow} L(\langle i+v_0, 0\rangle) \overset{\rho_1}{\hookrightarrow} \ldots \overset{\rho_q}{\hookrightarrow} L(\langle i+v_q, 0\rangle)$*, such that* $L(\langle i+v_j, 1\rangle) \ldots L(\langle i+v_j, |\sigma_{i+v_j}|-1\rangle)$*. is equal to* $\rho_q \ldots \rho_1 \rho_0 L(\langle i, 1\rangle) \ldots$ $L(\langle i, |\sigma_i|-1\rangle)$

We will say that $\langle j, q\rangle$ is a descendant of $\langle i, p\rangle$ if there is a sequence of $\rightsquigarrow$ edges from $\langle i, p\rangle$ to $\langle j, q\rangle$.

Lemma 25. *All nodes* $\langle n, 0\rangle$ *(*$n > 0$*) are descendents of node* $\langle 0, 0\rangle$*.*

Proof. Suppose $\langle i, p\rangle$ is not a descendent of $\langle 0, 0\rangle$, then let j be the smallest number such that for some q, $\langle i, p\rangle$ is a descendant of $\langle j, q\rangle$. The only sub-case of Definition 21 where a node does not have a predecessor is the last one, so $L(\langle j, m\rangle)\Uparrow$ for some $m < q$. Therefore there is some $r < p$ such that $L(\langle i, r\rangle)$ is a descendant of $L(\langle j, m\rangle)$. Hence, $p > 0$. □

Now we complete the proof of Theorem 20. Let T be the subtree formed by all descendants of node $\langle 0,0\rangle$. T must be infinite because it contains all nodes $\langle n,0\rangle$ (Lemma 25). T is finitely branching, therefore by König's Lemma it contains an infinite branch. Let B be the lowest infinite branch, that is, the infinite branch with nodes $\langle i,p_i\rangle$ such that for all i if $\langle i,q\rangle$ is on an infinite branch, then $q \geq p_i$.

Since for every i there is a unique p_i such that $\langle i,p_i\rangle \in B$, we may consider B as a function mapping i to p_i.

We claim that for infinitely many $i \geq 0$ we have $\langle i,0\rangle \in B$. Suppose that this is not the case, then for all n greater than some value k the nodes $\langle n,0\rangle$ are not in B.

Such a node $\langle n,0\rangle$ is a descendant of a node $\langle k+1,j\rangle$ with $j < |\sigma_j|$. Since there are infinitely many such n and finitely many such j, at least one node $\langle k+1,j\rangle$ must have infinitely many descendants $\langle n,0\rangle$. That node is therefore the root of an infinite subtree and we apply König's Lemma to find an infinite branch B' in this subtree. B' can be extended to an infinite branch in T, which contradicts the fact that B is the lowest infinite branch.

Now find the first j such that there is an $i < j$ with $L(\langle i,0\rangle) = L(\langle j,0\rangle)$ and $\langle i,0\rangle,\langle j,0\rangle \in B$. By Corollary 24 there is a reachability sequence $L(\langle i,0\rangle)\overset{\rho_0}{\hookrightarrow}\ldots\overset{\rho_q}{\hookrightarrow}L(\langle j,0\rangle)$. Since j is minimal, this sequence is a minimal cycle. Moreover, $L(\langle j,0\rangle)L(\langle j,1\rangle)\ldots L(\langle j,|\sigma_j|-1\rangle)$ is equal to $L(\langle i,0\rangle)\rho_q\ldots\rho_0 L(\langle i,1\rangle)\ldots L(\langle i,|\sigma_i|-1\rangle)$.

We can repeat this construction, finding the first $j' > j$ such that there is an i' satisfying $j < i' < j'$ and $L(\langle i',0\rangle) = L(\langle j',0\rangle)$, giving us another occurrence of a minimal cycle $L(\langle i',0\rangle)\overset{\xi_0}{\hookrightarrow}\ldots\overset{\xi_{q'}}{\hookrightarrow}L(\langle j',0\rangle)$, with $L(\langle j',0\rangle)L(\langle j',1\rangle)\ldots L(\langle j',|\sigma_{j'}|-1\rangle)$ is equal to $L(\langle i',0\rangle)\xi_{q'}\ldots\xi_0 L(\langle i',1\rangle)\ldots L(\langle i',|\sigma_{i'}|-1\rangle)$.

Repeating this construction infinitely often produces infinitely many occurrences of minimal cycles. Since there are only finitely many minimal cycles (Proposition 17), some minimal cycle occurs at least twice. Say $X\overset{\rho_0}{\hookrightarrow}\ldots\overset{\rho_q}{\hookrightarrow}X$ with occurrences $L(\langle i,0\rangle)\overset{\rho_0}{\hookrightarrow}\ldots\overset{\rho_q}{\hookrightarrow}L(\langle j,0\rangle)$ and $L(\langle i',0\rangle)\overset{\rho_0}{\hookrightarrow}\ldots\overset{\rho_q}{\hookrightarrow}L(\langle j',0\rangle)$. Setting $\rho = \rho_q\ldots\rho_0$, we know

$$\begin{aligned}
&L(\langle i,0\rangle) = L(\langle j,0\rangle) = L(\langle i',0\rangle) = L(\langle j',0\rangle) = X,\\
&L(\langle j,0\rangle)L(\langle j,1\rangle)\ldots L(\langle j,|\sigma_j|-1\rangle) = L(\langle i,0\rangle)\rho L(\langle i,1\rangle)\ldots L(\langle i,|\sigma_i|-1\rangle), \text{ and}\\
&L(\langle j',0\rangle)L(\langle j',1\rangle)\ldots L(\langle j',|\sigma_{j'}|-1\rangle) = L(\langle i',0\rangle)\rho L(\langle i',1\rangle)\ldots L(\langle i',|\sigma_{i'}|-1\rangle)
\end{aligned}$$

We consider two cases. First let $\rho = \lambda$, then

$$L(\langle j,0\rangle)\ldots L(\langle j,|\sigma_j|-1\rangle) = L(\langle i,0\rangle)\ldots L(\langle i,|\sigma_i|-1\rangle)$$

and thus $\sigma_i = \sigma_j$ which implies $\sigma_i \Leftrightarrow \sigma_j$. Thus we have found i and j as promised at the start of the proof.

The second case is $\rho \neq \lambda$. Since there are no normed stacking cycles and cycle $X\overset{\rho_0}{\hookrightarrow}\ldots\overset{\rho_q}{\hookrightarrow}X$ is stacking, it must be a perpetual cycle. This means that $\rho\Uparrow$.

Consequently (Proposition 9),

$$
\begin{aligned}
&L(\langle j,0\rangle)L(\langle j,1\rangle)\dots L(\langle j,|\sigma_j|-1\rangle) = \\
&L(\langle i,0\rangle)\rho L(\langle i,1\rangle)\dots L(\langle i,|\sigma_i|-1\rangle)\leftrightarrows \\
&L(\langle i,0\rangle)\rho = \\
&L(\langle i',0\rangle)\rho\leftrightarrows \\
&L(\langle i',0\rangle)\rho L(\langle i',1\rangle)\dots L(\langle i',|\sigma_{i'}|-1\rangle) = \\
&L(\langle j',0\rangle)L(\langle j',1\rangle)\dots L(\langle j',|\sigma_{j'}|-1\rangle)
\end{aligned}
$$

and thus $\sigma_j \leftrightarrows \sigma_{j'}$, and again we have found i and j as promised. This concludes the proof of Theorem 20. □

5 Linearization

A specification in GNF can be transformed into a linear specification if the conditions from the main theorem in the previous section are met. In this section we will give an effective linearization method. The idea behind the method is simply to get rid of anything following a perpetual variable and introduce new process variables corresponding to sequences of old ones. If this procedure converges, it yields a linear BPA-specification equivalent to the original one.

First we need some additional definitions.

Definition 26.

1. If σ is a non empty sequence of variables, then $[\sigma]$ denotes a fresh process variable.
2. If S is a specification, then $[S]$ is the collection of equations derived from S by replacing every summand $aXY\sigma$ by $a[XY\sigma]$.
3. The operator $*$ concatenates a sequence of variables to a process definition. It is defined as follows.

$$
(a_0\sigma_0 + \dots + a_n\sigma_n) * X\sigma = a_0\sigma_0 * X\sigma + \dots + a_n\sigma_n * X\sigma
$$

$$
a\rho * X\sigma = \begin{cases} a\rho X\sigma & \text{if } \rho\downarrow \\ a\rho & \text{if } \rho\Uparrow \end{cases}
$$

Definition 27. A specification S is reduced if for every summand $aX_0\dots X_n$ $(n>0)$ $(X_0\dots X_{n-1})\downarrow$.

Definition 28. The reduction $red(S)$ of a specification S is derived from S by replacing all summands $aX_0\dots X_n$ $(n>0)$ for which there exists $0\le i<n$ with $(X_0\dots X_{i-1})\downarrow$ and $X_i\Uparrow$ by $aX_0\dots X_i$.

A specification S can be linearized by calculating a sequence of equivalent specifications S_i $(i\ge 0)$. If S is regular, only a finite number of specifications must be calculated in order to reach a linear S_i. The specifications are defined as follows.

$$
\begin{aligned}
S_0 &= red(S) \\
S_{i+1} &= [S_i] \,\cup\, \{[XY\sigma] = Def_{red(S)}(X) * Y\sigma \mid \\
&\qquad X,Y\in V_S, \sigma\in V_S^*, \exists_{a\in A, Z\in V_{S_i}} aXY\sigma\subset Z, [XY\sigma]\notin V_{S_i}\}
\end{aligned}
$$

We will not present a detailed proof of the correctness of this method. We will only give the main steps of the proof.

It is easy to verify that every S_i is a reduced specification. Furthermore, by constructing a bisimulation, we have for all $X \in V_S$ and $i \geq 0$, $gr_S(X) \leftrightarrow gr_{S_i}(X)$.

Finally we have that S is regular if and only if for some $i \geq 0$ $S_i = S_{i+1}$. We will only sketch the proof. Suppose that $S_i = S_{i+1}$, then $S_i = [S_i]$, so there are no summands $aXY\sigma$ and thus S_i is linear, which implies that S is regular. For the other implication, suppose that all S_i are different, then there is an infinite sequence

$$X \rightarrow_{S_1} [\sigma_1] \rightarrow_{S_2} [\sigma_2] \rightarrow_{S_3} \ldots$$

such that $[\sigma_{i+1}] \in V_{S_{i+1}}$ and $[\sigma_{i+1}] \notin V_{S_i}$ for $i \geq 0$. This sequence can be transformed into an infinite sequence

$$X \rightarrow_S \sigma_1' \rightarrow_S \sigma_2' \rightarrow_S \ldots$$

of which infinitely many sequences σ_i' are not bisimilar. This contradicts regularity of S.

6 Example

We will apply the results from the previous sections to a simple example. Consider the following specification.

$$\begin{aligned} A &= aBCD \\ B &= bB + b \\ C &= cAC + c \\ D &= d \end{aligned}$$

Clearly the variables B, C and D are normed and since $aBCD$ is a summand of A, A is normed too. Next we derive a reachability sequence. Since $B\downarrow$, we have $A \overset{D}{\hookrightarrow} C$ and since $A\downarrow$, we have $C \overset{C}{\hookrightarrow} A$. Thus we have a reachability cycle $A \overset{D}{\hookrightarrow} C \overset{C}{\hookrightarrow} A$. This cycle is clearly stacking, and because $ACD\downarrow$ it is a normed cycle. Now we may conclude that the specification is not regular. Indeed we have an infinite sequence

$$A \twoheadrightarrow ACD \twoheadrightarrow ACDCD \twoheadrightarrow \ldots$$

Now consider a slightly modified system, which is derived from the previous system by deleting summand c of C. This makes C perpetual.

$$\begin{aligned} A &= aBCD \\ B &= bB + b \\ C &= cAC \\ D &= d \end{aligned}$$

The variables B and D are normed, while A and C are perpetual. We can find three minimal cycles

$$\begin{array}{l} B \overset{\lambda}{\hookrightarrow} B \\ A \overset{D}{\hookrightarrow} C \overset{C}{\hookrightarrow} A \\ C \overset{C}{\hookrightarrow} A \overset{D}{\hookrightarrow} C \end{array}$$

The first cycle is not stacking. The second and third cycle (which are in fact equal) are not normed, because $(ACD)\Uparrow$ and $(CDA)\Uparrow$. Following the main theorem, we conclude that the specification is regular. Now we can apply the linearization procedure and get for S_0 the reduction of S:

$$\begin{array}{l} A = aBC \\ B = bB + b \\ C = cA \\ D = d \end{array}$$

For S_1 we obtain:

$$\begin{array}{ll} A & = a[BC] \\ B & = bB + b \\ C & = cA \\ D & = d \\ [BC] & = bBC + bC \end{array}$$

Already S_2 is a linear specification:

$$\begin{array}{ll} A & = a[BC] \\ B & = bB + b \\ C & = cA \\ D & = d \\ [BC] & = b[BC] + bC \end{array}$$

7 Conclusions

We have proved that regularity of BPA systems is decidable. The question whether it is decidable that a single process variable defines a regular process is still open. We conjecture that it is decidable. A simple example shows that this question is more complicated than regularity of a complete BPA system. Consider the specification

$$\begin{array}{l} X = aYZ \\ Y = bYc + d \\ Z = eZ \end{array}$$

Then it is easy to show that X and Y are irregular, so the specification as a whole is irregular. If we would change the definition of Z into

$$Z = cZ$$

then the complete specification is still irregular (since Y is still irregular), but now X is regular. The reason is clearly that the normed stacking tail c^n of Y is reduced to a regular perpetual process c^∞ by appending Z.

From this example we conclude that it is necessary to take the actual values of the atomic actions into account when deciding regularity of a single process variable. This probably leads to a more complex decision procedure than the one presented in this paper. Since the reachability relation and normedness are completely independent of the actual atomic actions, only the presence of any atomic action plays a role in the decision prodedure presented here.

We do not think that the restriction to complete systems is a problem in practical applications. In most cases one is interested in the linearization of a complete system. Specifications in languages such as PSF [6] only consider complete systems, without singling out a specific variable.

We claim that the techniques described in this paper easily extend to BPA_δ (which results from BPA by adding the special process constant δ for unsuccessful termination). A more interesting topic for future research is the question whether there are extensions of BPA with some operator for parallelism, on which regularity is also decidable.

References

1. J.C.M. Baeten, J.A. Bergstra & J.W. Klop, Decidability of bisimulation equivalence for processes generating context-free languages, Proc. PARLE 87 (J.W. de Bakker, A.J. Nijman, P.C. Treleaven, eds.), LNCS 259, pp. 93-114, 1987.
2. J.C.M. Baeten & W.P. Weijland, Process algebra, Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, 1990.
3. J.A. Bergstra & J.W. Klop, Process theory based on bisimulation semantics, Linear time, branching time and partial order in logics and models for concurrency (J.W. de Bakker, W.P. de Roever, G. Rozenberg, eds.), LNCS 354, pp. 50-122, 1989.
4. S. Christensen, H. Hüttel & C. Stirling, Bisimulation equivalence is decidable for all context-free processes, Proc. CONCUR'92 (W.R. Cleaveland, ed.), LNCS 630, pp. 138-147, 1992.
5. S. Christensen, Y. Hirschfeld & F. Moller, Bisimulation equivalence is decidable for basic parallel processes, Proc. CONCUR'93 (E. Best, ed.), LNCS 715, pp. 143-157, 1993.
6. S. Mauw & G.J. Veltink, A process specification formalism, Fundamenta Informaticæ XIII, pp. 85-139, 1990.
7. S. Mauw & G.J. Veltink, Algebraic specification of communication protocols, Cambridge Tracts in Theoretical Computer Science 36, Cambridge University Press, 1993.
8. R. Milner, A complete inference system for a class of regular behaviours, JCSS 28, pp. 439–466, 1984.
9. D.M.R. Park, Concurrency and automata on infinite sequences, Proc. 5th GI Conf. (P. Duessen, ed.), LNCS 104, pp. 167-183, 1981.

This article was processed using the LaTeX macro package with LLNCS style