

# Regularity is Decidable for Normed BPA and Normed BPP Processes in Polynomial Time

Antonín Kučera  
e-mail: `tony@fi.muni.cz`

Faculty of Informatics, Masaryk University  
Botanická 68a, 60200 Brno  
Czech Republic

**Abstract.** We consider the problem of deciding regularity of normed BPP and normed BPA processes. A process is regular if it is bisimilar to a process with finitely many states. We show that regularity of normed BPP and normed BPA processes is decidable in polynomial time and we present constructive regularity tests. Combining these two results we obtain a rich subclass of normed PA processes (called sPA) where the regularity is also decidable. Moreover, constructiveness of this result implies decidability of bisimilarity for pairs of processes such that one process of this pair is sPA and the other has finitely many states.

## 1 Introduction

One of the most popular models for concurrency are process algebras like CCS, CSP or ACP. Various properties of these models have been studied in the last decades. This paper belongs to the bunch which could be labelled “decidability results”.

We consider the problem of deciding regularity in several process algebras. A process is regular if it is bisimilar to a process with finitely many states. Almost all interesting properties are decidable for finite-state processes and designed algorithms are practically usable.

This is no more true if one moves to process classes which contain also processes with infinitely many states (up to bisimilarity). Some problems can remain decidable—for example, bisimilarity is known to be decidable for BPA (see [1, 3, 9, 7, 6]) and BPP (see [5, 4]) processes. The same problem becomes undecidable for labelled Petri nets (see [10]). But even if a given property is decidable, the algorithm is usually not interesting from the practical point of view due to its complexity. Before running a complex algorithm, it is a good idea to ask whether the process we are dealing with can be replaced with some equivalent (bisimilar) process with finitely many states. If so, we can usually run a much more efficient algorithm. Natural questions are, whether the regularity is decidable for a given class of processes and whether the equivalent finite-state process can be effectively constructed.

This natural problem is generally undecidable (see [17]), but Mauw and Mulder showed in [14], that “regularity” is decidable in the class of BPA processes.

The quotes are important here because Mauw and Mulder used the word regularity in a different sense.

A recent result of Esparza and Jančar [11] says that regularity is decidable for labelled Petri nets. The algorithm is obtained by a combination of two semi-decidability results and hence there are no complexity estimations.

In this paper we prove that regularity is decidable in classes of normed BPA and normed BPP processes. Presented algorithms are polynomial (and practically usable). As normed BPP processes form a proper subclass of labelled Petri nets, this part of our result can be seen as a refinement of [11].

Combining decision algorithms for normed BPA and normed BPP we can even prove that regularity is decidable in quite a large subclass of normed PA processes (PA processes have both sequential and parallel compositions and were introduced in [2]). We denote this subclass sPA (“s” stands for simple). The class sPA is strictly greater than the union of normed BPA and normed BPP processes. Moreover, if the tested sPA process  $\Delta$  is regular then a process  $\Delta'$  with finitely many states such that  $\Delta \sim \Delta'$  can be effectively constructed.

An interesting related problem is decidability of various behavioural equivalences and preorders for pairs of processes such that one process of this pair has finitely many states. For example, Jančar and Moller proved in [12] that bisimilarity is decidable for a pair of labelled Petri nets provided one net of this pair is bounded (a net is bounded iff it has finitely many states). The same result holds for trace equivalence and simulation equivalence.

The constructiveness of our decidability result for sPA allows us to conclude that bisimilarity is decidable for pairs of processes, such that one process is sPA and the other has finitely many states (decidability of bisimilarity for PA and even for sPA processes are open questions).

## 2 Basic definitions

### 2.1 BPA and BPP processes

Let  $Act = \{a, b, c, \dots\}$  be a countably infinite set of *atomic actions*. Let  $Var = \{X, Y, Z, \dots\}$  be a countably infinite set of *variables* such that  $Var \cap Act = \emptyset$ . The classes of recursive BPA and BPP expressions are defined by the following abstract syntax equations:

$$\begin{aligned} E_{BPA} &::= a \mid X \mid E_{BPA}.E_{BPA} \mid E_{BPA} + E_{BPA} \\ E_{BPP} &::= a \mid X \mid aE_{BPP} \mid E_{BPP} \parallel E_{BPP} \mid E_{BPP} + E_{BPP} \end{aligned}$$

Here  $a$  ranges over  $Act$  and  $X$  ranges over  $Var$ . The symbol  $Act^*$  denotes the set of all finite strings over  $Act$ .

As usual, we restrict our attention to guarded expressions. A BPA or BPP expression  $E$  is *guarded* if every variable occurrence in  $E$  is within the scope of an atomic action.

A *guarded BPA (or BPP) process* is defined by a finite family  $\Delta$  of recursive process equations

$$\Delta = \{X_i \stackrel{def}{=} E_i \mid 1 \leq i \leq n\}$$

where  $X_i$  are distinct elements of  $Var$  and  $E_i$  are guarded BPA (or BPP) expressions, containing variables from  $\{X_1, \dots, X_n\}$ . The set of variables which appear in  $\Delta$  is denoted by  $Var(\Delta)$ .

The variable  $X_1$  plays a special role ( $X_1$  is sometimes called *the leading variable*)—it is a root of a labelled transition system, defined by the process  $\Delta$  and following rules:

$$\begin{array}{cccc} \frac{}{a \xrightarrow{a} \epsilon} & \frac{E \xrightarrow{a} E'}{E.F \xrightarrow{a} E'.F} & \frac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'} & \frac{F \xrightarrow{a} F'}{E + F \xrightarrow{a} F'} \\ \frac{E \xrightarrow{a} E'}{E\|F \xrightarrow{a} E'\|F} & \frac{F \xrightarrow{a} F'}{E\|F \xrightarrow{a} E\|F'} & \frac{E \xrightarrow{a} E'}{X \xrightarrow{a} E'} & (X \stackrel{def}{=} E \in \Delta) \end{array}$$

The symbol  $\epsilon$  denotes the empty expression with usual conventions:  $\epsilon\|E = E$ ,  $E\|\epsilon = E$  and  $\epsilon.E = E$ . Nodes of the transition system generated by  $\Delta$  are BPA (or BPP) expressions, which are often called *states of  $\Delta$* , or just “states” when  $\Delta$  is understood from the context. We also define the relation  $\xrightarrow{w,*}$ , where  $w \in Act^*$ , as the reflexive and transitive closure of  $\xrightarrow{a}$  (we often write  $E \rightarrow^* F$  instead of  $E \xrightarrow{w,*} F$  if  $w$  is irrelevant). Given two states  $E, F$ , we say that  $F$  is *reachable from  $E$* , if  $E \rightarrow^* F$ . States of  $\Delta$  which are reachable from  $X_1$  are said to be *reachable*.

*Remark 1.* Processes are often identified with their leading variables. Furthermore, if we assume a fixed process  $\Delta$ , we can view any process expression  $E$  (not necessarily guarded) whose variables are defined in  $\Delta$  as a process too; we simply add a new leading equation  $X \stackrel{def}{=} E'$  to  $\Delta$ , where  $X$  is a variable from  $Var$  such that  $X \notin Var(\Delta)$  and  $E'$  is a process expression which is obtained from  $E$  by substituting each variable in  $E$  with the right-hand side of its corresponding defining equation in  $\Delta$  ( $E'$  must be guarded now). All notions originally defined for processes can be used for process expressions in this sense too.

**Bisimulation** The equivalence between process expressions (states) we are interested in here is *bisimilarity* [16], defined as follows:

**Definition 1.** A binary relation  $R$  over process expressions is a bisimulation if whenever  $(E, F) \in R$  then for each  $a \in Act$

- if  $E \xrightarrow{a} E'$ , then  $F \xrightarrow{a} F'$  for some  $F'$  such that  $(E', F') \in R$
- if  $F \xrightarrow{a} F'$ , then  $E \xrightarrow{a} E'$  for some  $E'$  such that  $(E', F') \in R$

Processes  $\Delta$  and  $\Delta'$  are bisimilar, written  $\Delta \sim \Delta'$ , if their leading variables are related by some bisimulation.

**Normed processes** Important subclasses of BPA and BPP processes can be obtained by an extra restriction of *normedness*. A variable  $X \in Var(\Delta)$  is *normed* if there is  $w \in Act^*$  such that  $X \xrightarrow{w,*} \epsilon$ . In that case we define the *norm* of  $X$ , written  $|X|$ , to be the length of the shortest such  $w$ . Thus  $|X| = \min\{Length(w) \mid X \xrightarrow{w,*} \epsilon\}$ . A process  $\Delta$  is *normed* if all variables of  $Var(\Delta)$  are normed. The norm of  $\Delta$  is then defined to be the norm of  $X_1$ .

**Greibach normal form** Any BPA or BPP process  $\Delta$  can be effectively presented in so-called 3-Greibach normal form (see [1] and [4]). Before the definition we need to introduce the set  $Var(\Delta)^*$  of all finite sequences of variables from  $Var(\Delta)$ , and the set  $Var(\Delta)^\otimes$  of all finite multisets over  $Var(\Delta)$ . Each multiset of  $Var(\Delta)^\otimes$  denotes a BPP expression by combining its elements in parallel using the ‘ $\parallel$ ’ operator.

**Definition 2.** A BPA (resp. BPP) process  $\Delta$  is said to be in Greibach normal form (GNF) if all its equations are of the form

$$X \stackrel{def}{=} \sum_{j=1}^n a_j \alpha_j$$

where  $n \in \mathbb{N}$ ,  $a_j \in Act$  and  $\alpha_j \in Var(\Delta)^*$  (resp.  $\alpha_j \in Var(\Delta)^\otimes$ ). We also require that for each  $X \in Var(\Delta)$  there is reachable state  $\alpha$  of  $\Delta$  such that  $X \in \alpha$ . If  $Length(\alpha_j) \leq 2$  (resp.  $card(\alpha_j) \leq 2$ ) for each  $j$ ,  $1 \leq j \leq n$ , then  $\Delta$  is said to be in 3-GNF.

From now on we assume that all BPA and BPP processes we are working with are presented in GNF. This justifies also the assumption that all reachable states of a BPA process  $\Delta$  are elements of  $Var(\Delta)^*$  and all reachable states of a BPP process  $\Delta'$  are elements of  $Var(\Delta')^\otimes$ .

**sPA processes** To be able to define the class of sPA processes we first need the notion of VPA expression. VPA expressions are defined as follows:

$$E_{VPA} ::= X \mid E_{VPA} \parallel E_{VPA} \mid E_{VPA} \cdot E_{VPA}$$

The set of variables, which appear in a VPA expression  $\alpha$  is denoted  $Var(\alpha)$ . The class of sPA processes is composed of VPA expressions whose variables are normed BPA or normed BPP processes (see remark in Section 1). An sPA process  $\alpha$  generates a labelled transition system using SOS rules mentioned above. The root of this transition system is  $\alpha$ .

sPA processes form a subclass of PA processes (defined in [2]). The main restriction is the form of recursion which is allowed in sPA (the variables of  $Var(\alpha)$  can be defined recursively, but using only BPA or only BPP operators). In spite of this restriction, sPA class is strictly greater than the union of normed BPA and normed BPP processes. This is demonstrated by the following example:

*Example 1.* Let  $\Delta_1, \Delta_2$  be normed BPA processes given by the following equations:

$$\begin{array}{ll} \Delta_1: X \stackrel{def}{=} aY + bZ & \Delta_2: A \stackrel{def}{=} cB + dC \\ Y \stackrel{def}{=} a(Y.Y) + bX + b & B \stackrel{def}{=} c(B.B) + dA + d \\ Z \stackrel{def}{=} aX + b(Z.Z) + a & C \stackrel{def}{=} cA + d(C.C) + c \end{array}$$

Then there is no normed BPA or BPP process bisimilar to the sPA process  $X \parallel A$  (it can be proved using pumping lemmas for context-free languages and for BPP languages—see [4]).

## 2.2 Regular processes

The main question considered in this paper is, whether the behaviour of a given process is regular, i.e. whether it is bisimilar to a process with finitely many states.

**Definition 3.** A process  $\Delta$  is regular if there is a process  $\Delta'$  with finitely many states such that  $\Delta \sim \Delta'$ .

It is easy to show that a process is regular iff it can reach only finitely many states up to bisimilarity. In [15] it is shown, that regular processes can be represented in the following normal form:

**Definition 4.** A regular process  $\Delta$  is said to be in normal form if all its equations are of the form

$$X_i \stackrel{\text{def}}{=} \sum_{j=1}^{n_i} a_{ij} X_{ij}$$

where  $1 \leq i \leq n$ ,  $n_i \in N$ ,  $a_{ij} \in \text{Act}$  and  $X_{ij} \in \text{Var}(\Delta)$ .

Thus a process  $\Delta$  is regular iff there is a regular process  $\Delta'$  in normal form such that  $\Delta \sim \Delta'$ .

## 3 Constructive regularity tests for normed BPA, normed BPP and sPA processes

In this section we show that regularity is decidable for normed BPA, normed BPP and sPA processes. Proofs are mostly omitted due to the lack of space and can be found in [13].

**Definition 5.** Let  $\Delta$  be a normed BPA (resp. BPP) process. A variable  $X \in \text{Var}(\Delta)$  is growing if  $X \rightarrow^* X.\alpha$  (resp.  $X \rightarrow^* X\|\alpha$ ) where  $\alpha \neq \epsilon$ .

**Lemma 1.** Let  $\Delta$  be a normed BPA (resp. BPP) process in 3-GNF. It is decidable, whether  $\text{Var}(\Delta)$  contains a growing variable.

*Proof.* We define the binary relation *Grow* on  $\text{Var}(\Delta)$  in the following way:

$$(X, Y) \in \text{Grow} \iff X \rightarrow^* Y.\alpha \text{ (resp. } X \rightarrow^* Y\|\alpha) \text{ where } \alpha \neq \epsilon.$$

Clearly  $\text{Var}(\Delta)$  contains a growing variable iff there is  $X \in \text{Var}(\Delta)$  such that  $(X, X) \in \text{Grow}$ . We show that the relation *Grow* can be effectively constructed. We need two auxiliary binary relations on  $\text{Var}(\Delta)$ :

$$\rightsquigarrow: \quad X \rightsquigarrow Y \iff \text{there is a summand } a\alpha \text{ in the defining equation for } X \text{ in } \Delta \text{ such that } \alpha = Y.Z \text{ (resp. } \alpha = Y\|Z \text{ or } \alpha = Z\|Y) \text{ where } Z \in \text{Var}(\Delta)$$

$$\hookrightarrow: \quad X \hookrightarrow Y \iff \text{there is a summand } a\alpha \text{ in the defining equation for } X \text{ in } \Delta \text{ such that } Y \in \text{Var}(\alpha).$$

It is easy to prove that  $Grow = \hookrightarrow^* . \rightsquigarrow . \hookrightarrow^*$ , where  $\hookrightarrow^*$  denotes the reflexive and transitive closure of  $\hookrightarrow$ . Moreover, the composition  $\hookrightarrow^* . \rightsquigarrow . \hookrightarrow^*$  can be effectively constructed.

Let  $\Delta$  be a normed BPA or BPP process in GNF. Let  $n$  denote the number of summands which are contained in defining equations of  $\Delta$ . Relations  $\rightsquigarrow$  and  $\hookrightarrow$  can be computed in  $n$  steps. As relations can be represented as boolean matrices and closure as well as product of boolean matrices can be computed in polynomial time, we can conclude that the relation  $Grow$  can be computed in polynomial time.

**Proposition 1.** *Let  $\Delta$  be a normed BPA or BPP process in 3-GNF.  $\Delta$  is regular iff  $Var(\Delta)$  does not contain any growing variable.*

*Proof.* It can be found in [13].

**Proposition 2.** *Let  $\Delta$  be a normed BPA or BPP process. If  $\Delta$  is regular, then a regular process  $\Delta'$  in normal form such that  $\Delta \sim \Delta'$  can be effectively constructed.*

*Proof.* It can be found in [13].

Propositions 1 and 2 together say that there are constructive regularity tests for normed BPA and BPP processes.

It is interesting to compare this result with decidability issues of context-free languages. For any normed BPA or BPP process  $\Delta$  we can define its associated language over the alphabet  $Act$  in the following way:

$$Language(\Delta) = \{w \in Act^* \mid X_1 \xrightarrow{w}^* \epsilon\}$$

The variable  $X_1$  is the leading variable of  $\Delta$ . Normed BPA processes in 3-GNF can be seen as context-free grammars in 3-GNF and hence the class of languages generated by normed BPA processes is exactly the class of context-free languages. It is well-known that a CF language  $L$  is not regular iff each CF grammar generating  $L$  contains a self-embedding nonterminal. Self-embedding nonterminals are in some sense related to growing variables (see Definition 5). Growing variables and self-embedding nonterminals can be effectively recognised. However, an existence of a self-embedding nonterminal in a given CF grammar  $G$  does not allow to conclude that  $L(G)$  is not regular. There can still be an equivalent grammar which does not contain any self-embedding nonterminal. Bisimulation equivalence is a finer relation—if a given normed BPA process  $\Delta$  contains a growing variable, then each bisimilar normed BPA process  $\Delta'$  has this property. We could also speak about “regularity w.r.t. language equivalence” (a process  $\Delta$  is regular w.r.t. language equivalence if there is a process  $\Delta'$  with finitely many states such that  $Language(\Delta) = Language(\Delta')$ ). Hence regularity (w.r.t. bisimulation equivalence) is decidable for normed BPA processes, while regularity w.r.t. language equivalence is not.

Normed BPP processes generate a class of languages which is incomparable with context-free languages but it forms a proper subclass of context-sensitive languages.

Now we show that we can combine these two results and obtain a positive decidability result also for sPA processes:

**Proposition 3.** *Regularity is decidable for sPA processes in polynomial time.*

*Proof.* (sketch) It suffices to prove that an sPA process  $\alpha$  is regular iff all its variables from  $\text{Var}(\alpha)$  are regular (see remark in Section 1). The proof is easy to complete by induction on the structure of  $\alpha$ . The only subcase which is not immediate is  $\alpha = \beta_1.\beta_2$ . It can be proved using a cancelation lemma for normed processes (see [3]).

Moreover, if the tested sPA process is regular then the regular process  $\Delta'$  in normal form such that  $\alpha \sim \Delta'$  can be effectively constructed. It implies the following proposition:

**Proposition 4.** *Bisimilarity is decidable for pairs of processes such that one process of this pair is sPA and the other is a regular process in normal form.*

## 4 Conclusions, future work

If we compare the decidability results, obtained for classes of normed BPP and normed BPA processes, we can observe that they are of a similar form. This is not surprising if fact—the only difference between BPP and BPA algebras is the form of binary composition they provide—the parallel composition in the case of BPP and the sequential composition in the case of BPA. But these two operators have similar algebraic properties and it reflects in many things—processes of BPP and BPA can be represented in similar normal forms (GNF), there are similar cancelation properties, the notion of self-bisimulation, introduced in [3], can be defined in a uniform way (see [8]) and so on.

Presented regularity tests are of polynomial time complexity (all what has to be done is the computation of the *Grow* relation). But if we want to *construct* a bisimilar process with finitely many states in normal form, we are faced to the problem of exponential state explosion—hence the space complexity is exponential because we need a special variable (and a special defining equation) for each such state.

An open problem still remains the question of deciding regularity in the class of PA processes and this is the area we would like to examine in the future.

## Acknowledgement

All presented results were obtained during my stay at BRICS (Basic Research in Computer Science), Department of Computer Science, University of Aarhus.

I would like to thank Mogens Nielsen for his support and encouragement. He was always willing to listen to my ideas, commenting them in a very inspiring way. Thanks are also due to Mojmir Křetínský for reading the first draft of this paper and providing me with various helpful information.

## References

1. J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. In *Proceedings of PARLE'87*, volume 259 of *LNCS*, pages 93–114. Springer, 1987.
2. J.C.M. Baeten and W.P. Weijland. *Process Algebra*. Number 18 in Cambridge Tracts in TCS. Cambridge University Press, 1990.
3. D. Caucal. Graphes canoniques des graphes algébriques. Rapport de Recherche 872, INRIA, 1988.
4. S. Christensen. *Decidability and Decomposition in Process Algebras*. PhD thesis, The University of Edinburgh, 1993.
5. S. Christensen, Y. Hirshfeld, and F. Moller. Bisimulation is decidable for all basic parallel processes. In *Proceedings of CONCUR'93*, volume 715 of *LNCS*, pages 143–157. Springer, 1993.
6. S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. In *Proceedings of CONCUR'92*, volume 630 of *LNCS*, pages 138–147. Springer, 1992.
7. J.F. Groote. A short proof of the decidability of bisimulation for normed BPA processes. *IPL*, 42:167–171, 1992.
8. Y. Hirshfeld. Deciding equivalences in simple process algebras. Technical report ECS-LFCS-94-294, Department of Computer Science, University of Edinburgh, 1994.
9. H. Hüttel and C. Stirling. Actions speak louder than words: Proving bisimilarity for context-free processes. In *Proceedings of LICS'91*, pages 376–386. IEEE, 1991.
10. P. Jančar. Decidability questions for bisimilarity of Petri nets and some related problems. In *Proceedings of STACS'94*, volume 775 of *LNCS*, pages 581–592. Springer, 1994.
11. P. Jančar and J. Esparza. Deciding finiteness of Petri nets up to bisimilarity. In *Proceedings of ICALP'96*, volume 1099 of *LNCS*, pages 478–489. Springer, 1996.
12. P. Jančar and F. Moller. Checking regular properties of Petri nets. In *Proceedings of CONCUR'95*, volume 962 of *LNCS*, pages 348–362. Springer, 1995.
13. A. Kučera. Deciding regularity in process algebras. BRICS Report Series RS-95-52, Department of Computer Science, University of Aarhus, October 1995.
14. S. Mauw and H. Mulder. Regularity of BPA-systems is decidable. In *Proceedings of CONCUR'94*, volume 836 of *LNCS*, pages 34–47. Springer, 1994.
15. R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
16. D.M.R. Park. Concurrency and automata on infinite sequences. In *Proceedings 5<sup>th</sup> GI Conference*, volume 104 of *LNCS*, pages 167–183. Springer, 1981.
17. D. Taubner. *Finite Representations of CCS and TCSP Programs by Automata and Petri Nets*, volume 369 of *LNCS*. Springer, 1989.