

Deadlocking States in Context-free Process Algebra

Jiří Srba*

Faculty of Informatics MU,
Botanická 68a, 60200 Brno, Czech Republic
srba@fi.muni.cz

Abstract. Recently the class of BPA (or context-free) processes has been intensively studied and bisimilarity and regularity appeared to be decidable (see [CHS95, BCS95, BCS96]). We extend these processes with a deadlocking state into BPA_δ systems. Bosscher has proved that bisimilarity and regularity remain decidable [Bos97]. We generalise his approach introducing strict and nonstrict version of bisimilarity. We show that the BPA_δ class is more expressive w.r.t. (both strict and nonstrict) bisimilarity but it remains language equivalent to BPA. Finally we give a characterization of those BPA_δ processes which can be equivalently (up to bisimilarity) described within the ‘pure’ BPA syntax.

1 Introduction

This paper deals with BPA processes (Basic Process Algebra) extended with deadlocks. BPA represents the class of processes introduced by Bergstra and Klop (see [BK85]), which corresponds to the transition systems associated with Greibach normal form (GNF) context-free grammars in which only left-most derivations are permitted. For detailed description of the relation between language and process theory we refer to [HM96]. We define the class BPA_δ of BPA processes extended with deadlocks and introduce two alternative definitions – strict and nonstrict – of bisimilarity within this class.

The definition of BPA_δ systems is based on a special variable δ (we call it a deadlock). In the usual presentation every variable used in a BPA system is supposed to be defined but for the deadlock variable we allow no definition. This causes that if the system reaches a state where the first variable is δ , the system sticks at this state and no more actions can be performed.

Bosscher has proved in [Bos97] that decidability of bisimilarity and regularity in BPA systems extends to the BPA_δ systems. The trick used for this extension is based on the idea that δ can be simulated by an unnormed variable.

The main topic this article deals with is the issue of the language equivalence and of describing BPA_δ in bisimilar BPA syntax. We show in Section 3 that extending the BPA systems with deadlocks does not yield any language extension.

* The author is supported by the Grant Agency of the Czech Republic, grant No. 201/97/0456

On the other hand the class of BPA_δ systems is larger with regard to bisimilarity. An interesting question explored in this paper (Section 4) is concerned with deciding whether there exists an alternative description of a BPA_δ system in bisimilar BPA syntax. We show that it is decidable for the strict bisimilarity and we find a nice semantic characterization of the situation in the nonstrict case. Moreover we show that the corresponding BPA syntax can be effectively constructed.

Several proofs in this paper are just sketched and their full version can be obtained in [Srb98].

2 Basic definitions

When dealing with processes we need some structure to describe their operational semantics. As the most suitable structure transition systems are widely used. We introduce the labelled transition system in the extended version with the set of final states as can be found e.g. in [Mol96].

Definition 1. (labelled transition system) *A labelled transition system is a tuple $(S, \text{Act}, \longrightarrow, \alpha_0, F)$ where S is a set of states; Act is a set of actions (or labels); $\longrightarrow \subseteq S \times \text{Act} \times S$ is a transition relation, written $\alpha \xrightarrow{a} \beta$, for $(\alpha, a, \beta) \in \longrightarrow$; $\alpha_0 \in S$ is the root (or start state) of the transition system; $F \subseteq S$ is the set of final states which are terminal: for each $\alpha \in F$ there is no $a \in \text{Act}$ and $\beta \in S$ such that $\alpha \xrightarrow{a} \beta$.*

As usual we extend the transition relation to the elements of Act^* . We also write $\alpha \longrightarrow^* \beta$ instead of $\alpha \xrightarrow{w} \beta$ if $w \in \text{Act}^*$ is irrelevant.

Definition 2. (language generation) *Let $(S, \text{Act}, \longrightarrow, \alpha_0, F)$ be a labelled transition system and suppose that $\alpha \in S$. The language generated by the state α is $L(\alpha) \stackrel{\text{def}}{=} \{w \in \text{Act}^* \mid \exists \alpha' \in F : \alpha \xrightarrow{w} \alpha'\}$. We say that two states α and β are language equivalent, written $\alpha =_L \beta$, iff $L(\alpha) = L(\beta)$. Two labelled transition systems are language equivalent iff their roots are language equivalent.*

Definition 3. (bisimilarity) *Let $(S, \text{Act}, \longrightarrow, \alpha_0, F)$ be a labelled transition system. A binary relation $R \subseteq S \times S$ is a bisimulation iff whenever $(\alpha, \beta) \in R$ then for each $a \in \text{Act}$:*

- if $\alpha \xrightarrow{a} \alpha'$ then $\exists \beta' \in S : \beta \xrightarrow{a} \beta' \wedge (\alpha', \beta') \in R$
- if $\beta \xrightarrow{a} \beta'$ then $\exists \alpha' \in S : \alpha \xrightarrow{a} \alpha' \wedge (\alpha', \beta') \in R$
- $\alpha \in F \Leftrightarrow \beta \in F$

States $\alpha, \beta \in S$ are bisimilar ($\alpha \sim \beta$), iff $(\alpha, \beta) \in R$ for some bisimulation R .

2.1 BPA and BPA_δ systems

Assume that Var and Act are finite sets of *variables* and *actions* such that $\text{Var} \cap \text{Act} = \emptyset$. We define the class \mathcal{E}_{BPA} of *BPA expressions* as the union of ϵ (*empty process*) and a set $\mathcal{E}_{\text{BPA}}^+$, which is defined by the following abstract syntax:

$$E ::= a \mid X \mid E_1.E_2 \mid E_1 + E_2$$

Here a ranges over \mathcal{Act} and X ranges over \mathcal{Var} . We state $\mathcal{E}_{\text{BPA}} \stackrel{\text{def}}{=} \{\epsilon\} \cup \mathcal{E}_{\text{BPA}}^+$.

We call the BPA expressions as processes and later on we assume fixed sets \mathcal{Var} and \mathcal{Act} if no confusion is caused. As usual, we restrict our attention to *guarded* expressions: a BPA expression is guarded iff every variable occurrence is within the scope of an atomic action.

Definition 4. (BPA system) A BPA system is a quadruple $(\mathcal{Var}, \mathcal{Act}, \Delta, X_1)$ where \mathcal{Var} and \mathcal{Act} are finite sets of distinct variables ($\mathcal{Var} = \{X_1, \dots, X_n\}$) resp. actions; $X_1 \in \mathcal{Var}$ is the leading variable; Δ is a finite set of recursive equations $\Delta = \{X_i \stackrel{\text{def}}{=} E_i \mid i = 1, \dots, n\}$ where each $E_i \in \mathcal{E}_{\text{BPA}}^+$ is a guarded BPA expression with variables drawn from the set \mathcal{Var} and actions from \mathcal{Act} .

Speaking about variables and actions used in the system $(\mathcal{Var}, \mathcal{Act}, \Delta, X_1)$ we use the notation $\mathcal{Var}(\Delta)$ and $\mathcal{Act}(\Delta)$ and for shorter referring to the BPA system we often identify the system $(\mathcal{Var}, \mathcal{Act}, \Delta, X_1)$ with Δ .

Assume that we have a BPA system $(\mathcal{Var}, \mathcal{Act}, \Delta, X_1)$. This system determines a labelled transition system $(S, \mathcal{Act}, \longrightarrow, X_1, \{\epsilon\})$ whose states are BPA expressions built over \mathcal{Var} and \mathcal{Act} , \mathcal{Act} is the set of labels, the transition relation is the least relation satisfying the following SOS rules, X_1 is the root and ϵ is the only final state.

$\frac{}{a \xrightarrow{a} \epsilon}$	$\frac{E \xrightarrow{a} E'}{E.F \xrightarrow{a} E'.F}$ if $E' \neq \epsilon$	$\frac{E \xrightarrow{a} \epsilon}{E.F \xrightarrow{a} F}$
$\frac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'}$	$\frac{F \xrightarrow{a} F'}{E + F \xrightarrow{a} F'}$	$\frac{E \xrightarrow{a} E'}{X \xrightarrow{a} E'}$ if $X \stackrel{\text{def}}{=} E \in \Delta$

We now define the class BPA_δ of BPA systems with deadlock. The definition is very similar to the definition of BPA systems except for a new distinct variable δ . There is no operational rule for δ in the BPA_δ systems.

Definition 5. (BPA_δ system) A BPA_δ system is a quadruple $(\mathcal{Var}, \mathcal{Act}, \Delta, X_1)$ where $\mathcal{Var} = \{X_1, \dots, X_n, \delta\}$ (δ is a special variable called deadlock), \mathcal{Act} is a finite set of actions and Δ is a finite set of recursive equations $\Delta = \{X_i \stackrel{\text{def}}{=} E_i \mid i = 1, \dots, n\}$ where each $E_i \in \mathcal{E}_{\text{BPA}}^+$ is a guarded BPA expression with variables drawn from the set \mathcal{Var} and actions from \mathcal{Act} .

It is obvious that any BPA system is trivially a BPA_δ system. BPA_δ labelled (strict or nonstrict) transition system is defined as in the case of BPA systems. If $F = \{\epsilon\}$ is the only final state we call the labelled transition system *strict* and if the final states are $F = \{\epsilon, \delta\} \cup \{\delta.E \mid E \in \mathcal{E}_{\text{BPA}}^+\}$ we call it *nonstrict*.

This means that the relation of bisimulation differs for both these approaches. Similarly, we call the bisimulation *strict* resp. *nonstrict* (and write $\overset{s}{\sim}$ resp. $\overset{n}{\sim}$) according to the type of the labelled transition system we take into account. These two notions of bisimilarity imply that $\delta \overset{n}{\sim} \epsilon$ but $\delta \not\overset{s}{\sim} \epsilon$. An easy consequence of decidability of bisimilarity in BPA_δ [Bos97] is that both $\overset{s}{\sim}$ and $\overset{n}{\sim}$ are decidable. Following lemma results from the definition of $\overset{s}{\sim}$ and $\overset{n}{\sim}$.

Lemma 1. $\overset{s}{\sim} \subseteq \overset{n}{\sim}$

Let $X \in \mathcal{Var}$. We define the *norm* of X as $\|X\| \stackrel{\text{def}}{=} \min\{\text{length}(w) \mid \exists E : X \xrightarrow{w} E \not\rightarrow\}$, if such w exists; or $\|X\| \stackrel{\text{def}}{=} \infty$ otherwise. We call the variable X *normed* iff $\|X\| < \infty$. A process Δ is *normed* iff its leading variable is normed.

Definition 6. A BPA (resp. BPA_δ) system Δ is said to be in Greibach Normal Form (GNF) iff all its defining equations are of the form $X \stackrel{\text{def}}{=} \sum_{j=1}^m a_j \alpha_j$ where $m > 0$, $a_j \in \text{Act}(\Delta)$ and $\alpha_j \in \mathcal{Var}(\Delta)^*$. If $\text{length}(\alpha_j) < k$ for each j then Δ is said to be in k -GNF.

Following theorem justifies the usage of 3-GNF.

Theorem 1. Let Δ be a BPA_δ system. We can effectively find a BPA_δ system Δ' in 3-GNF such that $\Delta' \overset{s}{\sim} \Delta$ resp. $\Delta' \overset{n}{\sim} \Delta$.

Proof. The proof is based on the proof of 3-GNF for BPA systems (see e.g. [Hüt91]), which had to be modified to capture the behaviour of deadlocks. In fact we had to use some additional transformations exploiting (from left to right) the rules $\delta + E \sim E$ and $\delta.E \sim \delta$. \square

3 Expressibility of BPA_δ systems

In this section we justify the importance of introducing a deadlocking state into the BPA systems. We show that deadlocks enlarge the descriptive power of BPA systems w.r.t. both strict and nonstrict bisimilarity. On the other hand introducing deadlocks does not allow to generate more languages.

Theorem 2. There exists a BPA_δ system such that no BPA system is strictly bisimilar to it.

Proof. No BPA system can be strictly bisimilar to the system $\{X \stackrel{\text{def}}{=} a\delta\}$ since δ is reachable in this system and there is no match for δ in any BPA system. \square

Theorem 3. There exists a BPA_δ system such that no BPA system is non-strictly bisimilar to it.

Proof. We define a BPA_δ system Δ and show that there is no BPA system Δ' such that $\Delta \overset{n}{\sim} \Delta'$. Consider $\Delta = \{X \stackrel{\text{def}}{=} aXX + b + c\delta\}$ and suppose that there is a BPA system Δ' in 3-GNF, $\Delta' = \{Y_i \stackrel{\text{def}}{=} E_i \mid i = 1, \dots, n\}$, such that $\Delta \overset{n}{\sim} \Delta'$. Then there are infinitely many states reachable from the leading variable X of the system Δ . They are of the form X^n for $n \geq 1$ and for each such state there must be reachable a state E from Δ' such that $X^n \overset{n}{\sim} E$. The state X^n still has norm 1 whereas norm 1 for BPA processes implies that it must be a single variable. Thus Δ is nonstrictly bisimilar to a system with finitely many reachable states, which is contradiction – Δ is a system where infinitely many nonstrictly nonbisimilar states are reachable. \square

In what follows we show that the classes of BPA and BPA_δ systems are equivalent w.r.t. language generation. We will consider just the nonstrict case ($F = \{\epsilon, \delta\} \cup \{\delta.E \mid E \in \mathcal{E}_{\text{BPA}}^+\}$) since it is obvious that the strict case cannot bring any language extension.

Definition 7. We define classes of languages generated by BPA resp. BPA_δ systems as following: $\mathcal{L}(\text{BPA}) \stackrel{\text{def}}{=} \{L(\Delta) \mid \Delta \text{ is a BPA system}\}$ and $\mathcal{L}(BPA_\delta) \stackrel{\text{def}}{=} \{L(\Delta_\delta) \mid \Delta_\delta \text{ is a } BPA_\delta \text{ system}\}$.

Theorem 4. It holds that $\mathcal{L}(\text{BPA}) = \mathcal{L}(BPA_\delta)$.

Proof. We show that for a BPA_δ system Δ_δ there exists a BPA system Δ such that $L(\Delta_\delta) = L(\Delta)$. The other direction is obvious.

Our proof will be constructive. For each variable $X \in \Delta_\delta$ we define a couple of new variables X^ϵ, X^δ . The first one will simulate the language behaviour of X when reaching the state ϵ , the second one will simulate ending in the suffix of the form $\delta\alpha$. We use the notation $a\alpha \in Y$ meaning that $a\alpha$ is a summand in the defining equation of the variable Y . W.l.o.g. let Δ_δ be a BPA_δ system in 3-GNF. The variables of the system Δ will be $\mathcal{V}ar(\Delta) \stackrel{\text{def}}{=} \cup_{X \in \mathcal{V}ar(\Delta_\delta) - \{\delta\}} \{X^\epsilon, X^\delta\} \cup \{X_1^{\epsilon\delta}\}$ where X^ϵ, X^δ are distinct fresh variables and $X_1^{\epsilon\delta}$ is the leading variable, supposing that X_1 was the leading variable of Δ_δ . Next we realize that the summands of the defining equation for $X \in \mathcal{V}ar(\Delta_\delta) - \{\delta\}$ are exactly of one of the following form (because of 3-GNF):

$$(a) \ aAB \quad (b) \ bC \quad (c) \ c \quad (d) \ dD\delta \quad (e) \ e\delta \quad (1)$$

where $a, b, c, d, e \in \mathcal{A}ct(\Delta_\delta)$ and $A, B, C, D \in \mathcal{V}ar(\Delta_\delta)$ such that $A, B, C, D \neq \delta$. Notice that we can suppose that there is no summand of the form $a\delta A$ because it can be replaced with $a\delta$.

We now define the variables from Δ . For each $X \in \mathcal{V}ar(\Delta_\delta) - \{\delta\}$ and for the summands of the variables X^ϵ and X^δ will hold:

$$\begin{array}{llll} \text{if } aAB \in X & \text{then } aA^\epsilon B^\epsilon \in X^\epsilon & \text{and } aA^\epsilon B^\delta + aA^\delta \in X^\delta \\ \text{if } bC \in X & \text{then } bC^\epsilon \in X^\epsilon & \text{and } bC^\delta \in X^\delta \\ \text{if } c \in X & \text{then } c \in X^\epsilon & & \\ \text{if } dD\delta \in X & \text{then } dD^\epsilon + dD^\delta \in X^\delta & & \\ \text{if } e\delta \in X & \text{then } e \in X^\delta & & \end{array}$$

$$\text{if } X_1^\epsilon \stackrel{\text{def}}{=} E \text{ and } X_1^\delta \stackrel{\text{def}}{=} F \text{ then } X_1^{\epsilon\delta} \stackrel{\text{def}}{=} E + F$$

If it is the case that there is a variable $Y \in \mathcal{V}ar(\Delta)$ such that Y does not have any summand we define $Y \stackrel{\text{def}}{=} aY$. (This variable cannot generate any nonempty language because it is unnormed). Finally we state $X_1^{\epsilon\delta}$ to be the leading variable of the system Δ .

Example 1. Let us have a BPA_δ system $\Delta_\delta = \{X \stackrel{\text{def}}{=} aXX + b + c\delta + bY, Y \stackrel{\text{def}}{=} b\}$. The corresponding language equivalent BPA system Δ looks as following: $\Delta = \{X^\epsilon \stackrel{\text{def}}{=} aX^\epsilon X^\epsilon + b + bY^\epsilon, X^\delta \stackrel{\text{def}}{=} aX^\epsilon X^\delta + aX^\delta + c + bY^\delta, Y^\epsilon \stackrel{\text{def}}{=} b, Y^\delta \stackrel{\text{def}}{=} a.Y^\delta, X^{\epsilon\delta} \stackrel{\text{def}}{=} aX^\epsilon X^\epsilon + b + bY^\epsilon + aX^\epsilon X^\delta + aX^\delta + c + bY^\delta\}$.

It is not difficult to see that the newly defined system Δ is in 3-GNF and we show that $L(\Delta_\delta) = L(\Delta)$. For this we need one lemma using following notation.

Definition 8. Let Δ' be a BPA (resp. BPA_δ) system in 3-GNF, $n \geq 1$ and $Y \in \mathcal{Var}(\Delta')$. We define $L_n^\epsilon(Y)$ and $L_n^\delta(Y)$ as following:

$$\begin{aligned} L_n^\epsilon(Y) &\stackrel{\text{def}}{=} \{w \in \mathcal{Act}(\Delta')^* \mid Y \xrightarrow{w} \epsilon \wedge \text{length}(w) \leq n\} \\ L_n^\delta(Y) &\stackrel{\text{def}}{=} \{w \in \mathcal{Act}(\Delta')^* \mid \exists \alpha \in \mathcal{Var}(\Delta')^* : Y \xrightarrow{w} \delta\alpha \wedge \text{length}(w) \leq n\}. \end{aligned}$$

Lemma 2. For all $n \geq 1$ and $X \in \mathcal{Var}(\Delta_\delta) - \{\delta\}$ holds that $L_n^\epsilon(X) = L_n^\epsilon(X^\epsilon)$ and $L_n^\delta(X) = L_n^\epsilon(X^\delta)$.

Proof. The proof is led by induction on n , following the subcases from (1). \square

To finish the proof of our theorem let us define for $n \geq 1$ the set $L_n(Y) \stackrel{\text{def}}{=} \{w \in L(Y) \mid \text{length}(w) \leq n\}$. Notice that because of the Lemma 2 we get $L_n(X_1) = L_n^\epsilon(X_1) \cup L_n^\delta(X_1) = L_n^\epsilon(X_1^\epsilon) \cup L_n^\epsilon(X_1^\delta) = L_n(X_1^{\epsilon\delta})$ for all $n \geq 1$.

Now it is clear that $L(X_1) = L(X_1^{\epsilon\delta})$ since if $w \in L(X_1)$ then $\exists n : w \in L_n(X_1)$ and so $w \in L_n(X_1^{\epsilon\delta})$ which implies that $w \in L(X_1^{\epsilon\delta})$. The other direction is similar. We have shown that $L(\Delta_\delta) = L(\Delta)$ and our proof is complete. \square

4 Describing BPA_δ in BPA syntax

We have shown that w.r.t. bisimilarity the class of BPA_δ systems is strictly larger than that of BPA. This challenges the question whether a given BPA_δ system can be equivalently described in BPA syntax.

Theorem 5. Let $(\mathcal{Var}, \mathcal{Act}, \Delta, X_1)$ be a BPA_δ system. It is decidable whether there exists a BPA system Δ' such that $\Delta \stackrel{\approx}{\sim} \Delta'$. Moreover if the answer is positive, the system Δ' can be effectively constructed.

Proof. The proof is standard and is based on the fact that $\delta \not\stackrel{\approx}{\sim} \epsilon$. Suppose w.l.o.g. that the system Δ is in 3-GNF. The notation $\alpha \in E$ means again that α is a summand in the expression E .

We will construct the sets M_0, M_1, \dots of variables from which the deadlock is reachable as following: $M_0 \stackrel{\text{def}}{=} \{\delta\}$ and for $i \geq 0$ the sets M_{i+1} are defined as $M_{i+1} \stackrel{\text{def}}{=} M_i \cup \{X \in \mathcal{Var} \mid \exists a \in \mathcal{Act}, \exists Y \in \mathcal{Var}, \exists Z \in M_i : (X \stackrel{\text{def}}{=} E) \in \Delta, a.Z \in E \vee a.Z.Y \in E \vee (a.Y.Z \in E \text{ and } \|Y\| < \infty)\}$.

We remind that the norm of a variable can be effectively computed. Let us denote the fixed point of this construction as M . We can see that for each $X \in \mathcal{Var}$: $X \xrightarrow{*} \delta.\alpha$ for some $\alpha \in \mathcal{Var}^*$ iff $X \in M$. If $X_1 \in M$ then Δ cannot be expressed by a BPA syntax since the deadlocking state is reachable from X_1 . If $X_1 \notin M$ we can naturally transform Δ into a BPA system. \square

The situation for the nonstrict case will be nicely characterised by the Corollary 1. In what follows, the set of variables from which a deadlocking state is

reachable will be of great importance. Hence we define the set $\mathcal{V}ar_\delta$ of such variables: $\mathcal{V}ar_\delta \stackrel{\text{def}}{=} \{X \in \mathcal{V}ar \mid X \longrightarrow^* \delta \text{ or } \exists E \in \mathcal{E}_{\text{BPA}}^+ : X \longrightarrow^* \delta.E\} - \{\delta\}$ and we state $\mathcal{V}ar_\epsilon \stackrel{\text{def}}{=} \mathcal{V}ar - \{\delta\} - \mathcal{V}ar_\delta$. The sets $\mathcal{V}ar_\delta$ and $\mathcal{V}ar_\epsilon$ can be effectively constructed as we have demonstrated in the proof of the Theorem 5. In what follows let the variables U, V, X, Y, Z range over $\mathcal{V}ar_\delta$ and A, B, C over $\mathcal{V}ar_\epsilon$.

Theorem 6. *Let $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ be a BPA $_\delta$ system in 3-GNF. Suppose that there are only finitely many pairwise nonstrictly nonbisimilar $Y\alpha \in \mathcal{V}ar_\delta.\mathcal{V}ar^*$ such that $X_1 \longrightarrow^* Y\alpha$. Then there exists a BPA system $(\mathcal{V}ar', \mathcal{A}ct', \Delta', X'_1)$ such that $\Delta \stackrel{n}{\sim} \Delta'$.*

Proof. Let us suppose that $X_1 \in \mathcal{V}ar_\epsilon$. Then the system Δ can be trivially transformed into bisimilar BPA system Δ' . Thus assume that $X_1 \in \mathcal{V}ar_\delta$.

We may suppose w.l.o.g. that each summand of every defining equation in Δ does not contain an unnormed variable (resp. δ) followed by another variable. We define functions f_α for each $\alpha \in \mathcal{V}ar^*$. These functions take an expression from $\mathcal{E}_{\text{BPA}}^+$ in 3-GNF and transform it into another expression. Our goal is following. We want to achieve $f_\alpha(E) \stackrel{n}{\sim} E\alpha$ and there should be no deadlock in $f_\alpha(E)$. For each $\alpha \in \mathcal{V}ar^*$ let us also define a function r_α which returns the set of the new variables added by the function f_α . Let us assume that $X, Y, U \in \mathcal{V}ar_\delta$, $A, B, C \in \mathcal{V}ar_\epsilon$ with $\|C\| = \infty$, $\beta \in \mathcal{V}ar_\epsilon^*$ such that $\|\beta\| < \infty$ and $\gamma \in \mathcal{V}ar^*$.

$$\begin{array}{ll}
f_\alpha(\sum_{i=1}^n a_i\alpha_i) = \sum_{i=1}^n f_\alpha(a_i\alpha_i) & r_\alpha(\sum_{i=1}^n a_i\alpha_i) = \bigcup_{i=1}^n r_\alpha(a_i\alpha_i) \\
f_\alpha(aXY) = aX^{Y\alpha} & r_\alpha(aXY) = \{X^{Y\alpha}\} \\
f_\alpha(aX\delta) = aX^\epsilon & r_\alpha(aX\delta) = \{X^\epsilon\} \\
f_\alpha(a\delta) = a & r_\alpha(a\delta) = \emptyset \\
f_\alpha(aX) = aX^\alpha & r_\alpha(aX) = \{X^\alpha\} \\
f_\alpha(aAB) = aAB\beta U^\gamma & r_\alpha(aAB) = \{U^\gamma\} \quad \text{if } \alpha = \beta U\gamma \\
& = aAB\beta C & = \emptyset & \text{if } \alpha = \beta C\gamma \\
& = aAB\alpha & = \emptyset & \text{otherwise} \\
f_\alpha(a) = a\beta U^\gamma & r_\alpha(a) = \{U^\gamma\} \quad \text{if } \alpha = \beta U\gamma \\
& = a\beta C & = \emptyset & \text{if } \alpha = \beta C\gamma \\
& = a\alpha & = \emptyset & \text{otherwise} \\
f_\alpha(aA\delta) = aA & r_\alpha(aA\delta) = \emptyset \\
f_\alpha(aA) = aA\beta U^\gamma & r_\alpha(aA) = \{U^\gamma\} \quad \text{if } \alpha = \beta U\gamma \\
& = aA\beta C & = \emptyset & \text{if } \alpha = \beta C\gamma \\
& = aA\alpha & = \emptyset & \text{otherwise} \\
f_\alpha(aXA) = aX^{A\alpha} & r_\alpha(aXA) = \{X^{A\alpha}\} \\
f_\alpha(aAX) = aAX^\alpha & r_\alpha(aAX) = \{X^\alpha\}
\end{array}$$

Let us now construct the nonstrictly bisimilar BPA system Δ' where $\mathcal{V}ar' \stackrel{\text{def}}{=} \mathcal{V}ar_\epsilon \cup \text{Added}$; $\mathcal{A}ct' \stackrel{\text{def}}{=} \mathcal{A}ct$; $\Delta' \stackrel{\text{def}}{=} \Delta_\epsilon \cup \Gamma$; $X'_1 \stackrel{\text{def}}{=} X_1^\epsilon$. The sets Added and Γ are outputs of the following algorithm and $\Delta_\epsilon \subseteq \Delta$ contains exactly the defining equations for variables from $\mathcal{V}ar_\epsilon$.

Algorithm 1

```

1   Solve := {X1ε}
2   Added := {X1ε}
3   Γ := ∅
4   while Solve ≠ ∅ do
5       Let us fix Xα ∈ Solve with (X  $\stackrel{\text{def}}$  E) ∈ Δ
6       Γ := Γ ∪ {Xα  $\stackrel{\text{def}}$  fα(E)}
7       Add := {Yβ ∈ rα(E) | ∀Zω ∈ Added : Yβ  $\not\approx$  Zω}
8       while ∃Yβ, Zω ∈ Add : Yβ ≠ Zω ∧ Yβ  $\approx$  Zω do
9           Add := Add - {Yβ}
10      endwhile
11      Solve := (Solve - {Xα}) ∪ Add
12      Added := Added ∪ Add
13      for ∀Yβ ∈ rα(E) - Add do
14          replace all occurrences of Yβ in Γ with Zω
15          where Zω ∈ Added : Yβ  $\approx$  Zω
16      endfor
17  endwhile

```

In the following lemmas we demonstrate that the algorithm is correct and yields a BPA system Δ' such that $\Delta \approx \Delta'$.

Lemma 3. For the loop 4–17 of Alg.1 holds the following invariant: $\forall Y^\beta, Z^\omega \in \text{Added} : Y^\beta \neq Z^\omega \Rightarrow Y^\beta \not\approx Z^\omega$.

Proof. An easy observation. □

Lemma 4. Whenever during the execution of Alg.1 we have $Y^\alpha \in \text{Added}$ then $Y \in \text{Var}_\delta$.

Proof. All variables in **Added** had to be produced by the function r_α (see line 7 and 12). It is easily seen that $\{Y | Y^\beta \in r_\alpha(E)\} \subseteq \text{Var}_\delta$ for any $\alpha \in \text{Var}^*$ and $E \in \mathcal{E}_{\text{BPA}}^+$ such that E is in 3-GNF. □

Lemma 5. Whenever during the execution of Alg.1 we have $Y^\beta \in \text{Added}$ then $X_1 \longrightarrow^* Y^\beta$.

Proof. By induction on the number of repetitions of the loop 4–17.

Basic step: The only variable in the set **Added** before the execution of the loop 4–17 started is X_1^ϵ . However $X_1^\epsilon = X_1$ and so $X_1 \longrightarrow^* X_1^\epsilon$.

Induction step: Suppose that at line 12 we have added a new variable Y^β into **Added**. So at line 7 we had to have $Y^\beta \in r_\alpha(E)$ for some $X^\alpha \in \text{Solve}$ and $(X \stackrel{\text{def}}{=} E) \in \Delta$. The induction hypothesis says that $X_1 \longrightarrow^* X^\alpha$ (X^α had to be added in some previous repetition of the main loop). It must hold that $a\gamma Y^\beta \in f_\alpha(E)$ where $\gamma \in \text{Var}_\epsilon^*$ and $\|\gamma\| < \infty$. From the construction of f_α we can also see that $X^\alpha \longrightarrow^* Y^\beta$. Thus we get $X_1 \longrightarrow^* X^\alpha \longrightarrow^* Y^\beta$. □

Lemma 6. *Alg.1 cannot loop forever (under the assumption of the Theorem 6).*

Proof. Suppose that the algorithm loops forever which means that the set **Solve** is never empty. But in every loop we remove exactly one element from the set **Solve** (line 11). This implies that the set **Added** will grow arbitrarily because the set **Add** is infinitely often unempty (otherwise the algorithm would stop). The contradiction is immediate from the Lemmas 3, 4 and 5. \square

The following lemma is crucial for the proof of our theorem.

Lemma 7. *After the execution of Alg.1 we have $V^\alpha \stackrel{n}{\sim} V\alpha$ for all $V^\alpha \in \text{Added}$.*

Proof. We use the *stratified bisimulation relations* [Mil89] \sim_k . By induction on k we show that $V^\alpha \sim_k V\alpha$ for all $k \geq 0$. This implies that $V^\alpha \stackrel{n}{\sim} V\alpha$. This straightforward but also long and technical proof can be found in [Srb98]. \square

Lemma 8. *The system Δ' is a BPA system and moreover $X_1 \stackrel{n}{\sim} X_1^\epsilon$.*

Proof. Immediately from the Lemma 7. \square

We have constructed a BPA system Δ' such that $\Delta \stackrel{n}{\sim} \Delta'$. \square

Theorem 7. *Let $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ be a BPA_δ system. Suppose that there are infinitely many pairwise nonstrictly nonbisimilar $Y\alpha \in \mathcal{V}ar_\delta \cdot \mathcal{V}ar^*$ such that $X_1 \xrightarrow{*} Y\alpha$. Then there is no BPA system Δ' such that $\Delta \stackrel{n}{\sim} \Delta'$.*

Proof. The proof is based on the fact that $\alpha \stackrel{n}{\sim} \beta$ implies $\|\alpha\| = \|\beta\|$. Let us assume w.l.o.g. that there exists Δ' in 3-GNF such that $\Delta \stackrel{n}{\sim} \Delta'$. We show that this is not possible. Since there are infinitely many reachable states $Y_1\alpha_1, Y_2\alpha_2, \dots$ of Δ which are pairwise nonstrictly nonbisimilar there must be corresponding states β_1, β_2, \dots of the system Δ' such that $Y_i\alpha_i \stackrel{n}{\sim} \beta_i$ for $i = 1, 2, \dots$. Let us now define a constant N_{max} as $N_{max} \stackrel{\text{def}}{=} \max\{\|Y_i\|_\delta \mid i = 1, 2, \dots\}$ where $\|Y\|_\delta \stackrel{\text{def}}{=} \min\{\text{length}(w) \mid Y \xrightarrow{w} \delta \text{ or } \exists E \in \mathcal{E}_{\text{BPA}}^+ : Y \xrightarrow{w} \delta.E\}$. Notice that the definition of N_{max} is correct since for all i $\|Y_i\|_\delta < \infty$ (because $Y_i \in \mathcal{V}ar_\delta$) and there are only finitely many different Y_i 's.

Clearly $\|Y_i\alpha_i\| \leq N_{max}$ for all i . This implies that the norm of β_i is also less or equal N_{max} for all i . However, Δ' is a BPA system and all variables in Δ' are guarded. This means that there are only finitely many different states of Δ' such that their norm is less or equal N_{max} . Hence there must be two states β_k and β_l with $k \neq l$ such that $\beta_k = \beta_l$. This implies that $\beta_k \stackrel{n}{\sim} \beta_l$. Then also $Y_k\alpha_k \stackrel{n}{\sim} Y_l\alpha_l$, which is contradiction. \square

Suppose that we have a BPA_δ system and that there are infinitely many nonbisimilar states from which, after some 'short' sequence of actions, a deadlocking state is reachable. Then the corresponding (nonstrictly bisimilar) BPA system does not exist. This condition appears to be both necessary and sufficient as is illustrated by the following corollary.

Corollary 1. *Let $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ be a BPA_δ system. There are only finitely many pairwise nonstrictly nonbisimilar $Y\alpha \in \mathcal{V}ar_\delta \cdot \mathcal{V}ar^*$ such that $X_1 \xrightarrow{*} Y\alpha$ if and only if there exists a BPA system $(\mathcal{V}ar', \mathcal{A}ct', \Delta', X_1')$ such that $\Delta \stackrel{n}{\sim} \Delta'$.*

Proof. An immediate consequence of the Theorems 6 and 7. \square

5 Conclusion remarks

In this paper we have focused on the class of BPA processes extended with deadlocks. We have shown that for language equivalence the extension is no acquisition. On the other hand the BPA_δ class is larger with regard to the relation of bisimulation. We introduce two notions of bisimilarity to capture the different understanding of deadlock behaviour. If we do not distinguish between ϵ and δ , we speak about nonstrict bisimilarity and if we do, we call the appropriate bisimulation as strict. We have solved the question whether, given a BPA_δ system Δ , there is an equivalent description (with regard to bisimilarity) of Δ in terms of BPA syntax. The solution for the strict bisimilarity is straightforward. However, the answer to the problem dealing with the nonstrict bisimilarity exploited a nice semantic characterization of the subclass of BPA_δ processes bisimilarly describable in BPA syntax: a BPA_δ system can be transformed into a BPA system (preserving nonstrict bisimilarity) if and only if finitely many nonbisimilar states starting with some in δ -ending variable are reachable. There is still an open problem whether this semantic characterization is syntactically checkable.

Acknowledgements: First of all, I would like to thank Ivana Černá for her help and encouragement throughout the work. I am very grateful for her advise and valuable discussions. My warm thanks go also to Mojmír Křetínský and Antonín Kučera for their constant support and comments.

References

- [BCS95] O. Burkart, D. Caucal, and B. Steffen. An elementary decision procedure for arbitrary context-free processes. In *Proceedings of MFCS'95*, volume 969 of *LNCS*, pages 423–433. Springer-Verlag, 1995.
- [BCS96] O. Burkart, D. Caucal, and B. Steffen. Bisimulation collapse and the process taxonomy. In *Proceedings of CONCUR'96*, volume 1119 of *LNCS*, pages 247–262. Springer-Verlag, 1996.
- [BK85] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37:77–121, 1985.
- [Bos97] D. Bosscher. *Grammars Modulo Bisimulation*. PhD thesis, CWI, University of Amsterdam, 1997.
- [CHS95] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. *Information and Computation*, 121:143–148, 1995.
- [HM96] Y. Hirshfeld and F. Moller. Decidability results in automata and process theory. In *Logics for Concurrency: Automata vs Structure*, volume 1043 of *LNCS*, pages 102–148. F. Moller and G. Birtwistle, 1996.
- [Hüt91] H. Hüttel. *Decidability, Behavioural Equivalences and Infinite Transition Graphs*. PhD thesis, The University of Edinburgh, 1991.
- [Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
- [Mol96] F. Moller. Infinite results. In *Proceedings of CONCUR'96*, volume 1119 of *LNCS*, pages 195–216. Springer-Verlag, 1996.
- [Srb98] Jiří Srba. Comparing the classes BPA and BPA with deadlocks. Technical Report FIMU-RS-98-05, Faculty of Informatics, Masaryk University, 1998.