FI MU

Faculty of Informatics
Masaryk University

Comparing the Classes BPA and BPA with Deadlocks

by

Jiří Srba

FI MU Report Series FIMU-RS-98-05

June 1998

Comparing the Classes BPA and BPA with Deadlocks*

Jiří Srba†

Abstract

The class of BPA (or context-free) processes has been intensively studied and bisimilarity and regularity appeared to be decidable (see [CHS95], [BCS96]). We extend these processes with a deadlocking state into BPA_δ systems. We show that the BPA_δ class is more expressive w.r.t. bisimulation equivalence but it remains language equivalent to BPA . We prove that bisimilarity and regularity remain decidable in the BPA_δ class. Finally we give a characterisation of those BPA_δ processes which can be equivalently (up to bisimilarity) described within the 'pure' BPA syntax.

1 Introduction

Recently a labelled transition system as the abstract computational model, and the relation of bisimulation as the most suitable behavioural equivalence, have been generally accepted in the process theory. It should be remarked that another equivalences have been explored; probably the most exhaustive spectrum of them can be found in [vG90a] and [vG90b] but the bisimulation still appears as the finest one.

This paper deals with BPA processes (Basic Process Algebra) extended with deadlocking states. BPA represents the class of processes introduced by Bergstra and Klop (see [BK85]). This class corresponds to the transition systems associated with Greibach normal form (GNF) context-free

*Supported by the Grant Agency of the Czech Republic, grant No. 201/97/0456

†Faculty of Informatics MU, Botanická 68a, 60200 Brno, Czech Republic, e-mail: srba@fi.muni.cz

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 and regularity 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. There are two approaches to the deadlocking state. First, δ identifies only with the situation when the process gets into an inner state where it loops forever. However, no actions (for an observer of such a system) can be seen. Second, we identify the deadlock with a regularly finished execution of the process.

We show in Section 3 that extending BPA systems with deadlock does not yield any language extension. On the other hand the class of BPA_δ systems is larger with regard to bisimilarity – the behaviour equivalence. It is known from [CHS95] and [BCS96] that bisimilarity and regularity is decidable in the BPA systems. Bosscher has proved in [Bos97] that decidability of bisimilarity and regularity extends to the BPA_δ systems. The trick used for this extention is based on the idea that δ can be simulated by an unnormed variable. We show in Sections 4 and 5 that this approach can be applied for both strict and nonstrict versions of bisimilarity and regularity. Moreover we show that strict and nonstrict regularity coincide.

The last question explored in this paper (Section 6) is concerned with deciding whether there exists an alternative description of a BPA_δ system in bisimilar BPA syntax. We prove that it is decidable for the strict bisimilarity and we find a nice semantic characterisation of the situation in the nonstrict case. Moreover we show that the corresponding BPA syntax can be effectively constructed.

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 and in the rest of this paper we will understand processes as nodes of a certain type transition system. 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* $(S, \mathcal{A}ct, \longrightarrow, \alpha_0, F)$ *where*

- *S is a set of* states *(or* processes*)*
- *$\mathcal{A}ct$ is a set of* labels *(or* actions*)*
- $\longrightarrow \subseteq S \times \mathcal{A}ct \times S$ *is a* transition relation*, written* $\alpha \stackrel{a}{\longrightarrow} \beta$*, 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* $a \in \mathcal{A}ct$ *and* $\beta \in S$ *such that* $\alpha \stackrel{a}{\longrightarrow} \beta$*.*

The transition relation $\longrightarrow$ can be alternatively understood as a set of binary relations $\{\stackrel{a}{\longrightarrow}\}_{a \in \mathcal{A}ct}$. As usual we extend the transition relation to the elements of $\mathcal{A}ct^*$ ($\alpha \stackrel{\epsilon}{\longrightarrow} \alpha$ and inductively $\alpha \stackrel{aw}{\longrightarrow} \beta$ iff $\exists \gamma : \alpha \stackrel{a}{\longrightarrow} \gamma$ and $\gamma \stackrel{w}{\longrightarrow} \beta$ where $\alpha, \beta, \gamma \in S$, $a \in \mathcal{A}ct$ and $w \in \mathcal{A}ct^*$). We also write $\alpha \longrightarrow^* \beta$ instead of $\alpha \stackrel{w}{\longrightarrow} \beta$ if $w \in \mathcal{A}ct^*$ is irrelevant. A state β is *reachable* from the state α, iff $\alpha \longrightarrow^* \beta$. *Reachable states* in a labelled transition system are the states reachable from the root. We also define the unary relation $\not\longrightarrow$ for $\alpha \in S$ as $\alpha \not\longrightarrow$ iff there is no $\beta \in S$ and no $a \in \mathcal{A}ct$ such that $\alpha \stackrel{a}{\longrightarrow} \beta$.

Definition 2. (language generation) *Let* $(S, \mathcal{A}ct, \longrightarrow, \alpha_0, F)$ *be a labelled transition system and suppose that* $\alpha \in S$*. The* language generated by the process α *is*

$$L(\alpha) \stackrel{\text{def}}{=} \{w \in \mathcal{A}ct^* \mid \exists \alpha' \in F : \alpha \stackrel{w}{\longrightarrow} \alpha'\}.$$

We say that two processes 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.*

In the concurrency theory, language equivalence is generally taken to be too coarse equivalence. Many better equivalences have been introduced e.g. in [vG90b] and [vG90a], with *bisimulation equivalence* being perhaps the finest one. Bisimulation equivalence was defined by Park [Par81] and used with great effect by Milner [Mil89]. Its definition is following.

Definition 3. (bisimilarity) *Let $(S, \mathcal{A}ct, \longrightarrow, \alpha_0, F)$ 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 $a \in \mathcal{A}ct$:*

- *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 bisimulation equivalent *or* bisimilar *, written $\alpha \sim \beta$, iff $(\alpha, \beta) \in R$ for some bisimulation R.*

Now we can state an obvious lemma.

Lemma 1. *Let $(S, \mathcal{A}ct, \longrightarrow, \alpha_0, F)$ be a labelled transition system. Then for all $\alpha, \beta \in S$ if $\alpha \sim \beta$ then $\alpha =_L \beta$.*

2.1 BPA and BPA$_\delta$ systems

Assume that $\mathcal{V}ar$ and $\mathcal{A}ct$ are finite sets of *variables* resp. *actions* such that $\mathcal{V}ar \cap \mathcal{A}ct = \emptyset$. We define the class $\mathcal{E}_{\text{BPA}}$ 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:

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

Here a ranges over $\mathcal{A}ct$ and X ranges over $\mathcal{V}ar$. 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 $\mathcal{V}ar$ and $\mathcal{A}ct$ 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.

Example 1. *The expressions $a.X$, $a.(b + X)$, $(a + b).X.(Y + Z)$ are guarded whereas X, $a + X$, $(a + b + X).c$, ϵ are not guarded.*

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

$$\frac{}{a \xrightarrow{a} \epsilon} \qquad \frac{E \xrightarrow{a} E'}{E.F \xrightarrow{a} E'.F} \text{ if } E' \neq \epsilon \qquad \frac{E \xrightarrow{a} \epsilon}{E.F \xrightarrow{a} F}$$

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

Figure 1: SOS rules

Speaking about variables and actions used in the system $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ we use the notation $\mathcal{V}ar(\Delta)$ and $\mathcal{A}ct(\Delta)$ and for shorter referring to the BPA system we often identify the system $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ with Δ. In what follows we restrict our attention to guarded BPA systems and often omit the word 'guarded'. We also use the notation X^n where $X \in \mathcal{V}ar$, meaning sequential composition $\underbrace{X.X...X}_{n\times}$.

Definition 5. (BPA labelled transition system)
Assume that we have a guarded BPA system $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$. This system determines a labelled transition system $(S, \mathcal{A}ct, \{\xrightarrow{a}\}_{a \in \mathcal{A}ct}, X_1, \{\epsilon\})$ whose states are BPA expressions built over $\mathcal{V}ar$ and $\mathcal{A}ct$, $\mathcal{A}ct$ is the set of labels, the transition relations are the least relations satisfying the SOS rules of Figure 1, X_1 is the root and ϵ is the only final state.

We may assume that the operator '.' for sequential composition is associative and the operator '+' for nondeterministic choice is associative and commutative.
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 6. (guarded BPA_δ system) *A* guarded BPA_δ system *is a quadruple $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ where $\mathcal{V}ar = \{X_1, \ldots, X_n, \delta\}$ (δ is a special variable called* deadlock*), $\mathcal{A}ct$ is a finite set of actions and Δ is a finite set of recursive equations $\Delta = \{X_i \stackrel{\text{def}}{=} E_i \mid i = 1, \ldots, n\}$ where each $E_i \in \mathcal{E}^+_{BPA}$ is a guarded BPA expression with variables drawn from the set $\mathcal{V}ar$ and actions from $\mathcal{A}ct$.*

It is obvious that any BPA system is trivially a BPA_δ system (we simply add δ into variables but we do not use it).
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 $F = \{\epsilon, \delta\} \cup \{\delta.E | E \in \mathcal{E}^+_{\mathrm{BPA}}\}$ we call it *nonstrict*.

Remark. *As there is no defining equation for the variable δ it holds that $\delta.E \not\longrightarrow$ for any $E \in \mathcal{E}^+_{BPA}$.*

Definition 7. *We call the bisimulation* strict *resp.* nonstrict *(and write $\overset{s}{\sim}$ resp. $\overset{n}{\sim}$) according to the type of labelled transition system we take into account ($F = \{\epsilon\}$ resp. $F = \{\epsilon, \delta\} \cup \{\delta.E | E \in \mathcal{E}^+_{BPA}\}$).*

Remark. *These two notions of bisimilarity imply that $\delta \overset{n}{\sim} \epsilon$ but $\delta \overset{s}{\not\sim} \epsilon$.*

We say that a pair of BPA_δ systems Δ and Δ' is (strictly resp. nonstrictly) bisimilar (and we write $\Delta \overset{s}{\sim} \Delta'$ resp. $\Delta \overset{n}{\sim} \Delta'$) iff their corresponding (strict resp. nonstrict) labelled transition systems are bisimilar. Following lemma results from the definitions of $\overset{s}{\sim}$ and $\overset{n}{\sim}$.

Lemma 2. $\overset{s}{\sim} \subseteq \overset{n}{\sim}$

An important subclass of BPA (resp. BPA_δ) systems can be obtained by an extra restriction on the involved processes – *normedness*.

Definition 8. *Let $E \in \mathcal{E}_{BPA}$. We define the* norm *of E as:*

$$||E|| \overset{\mathrm{def}}{=} \begin{cases} \min\{ \, length(w) \mid \exists G : E \overset{w}{\longrightarrow} G \not\longrightarrow\}, \textit{if such } w \textit{ exists} \\ \infty, \ \ \textit{otherwise} \end{cases}$$

We call the expression E normed iff $\| E \| < \infty$. A process Δ is normed iff its leading variable is normed.

We remind the fact that the norm of E can be effectively computed in BPA_δ systems.
An interesting property of processes is *regularity*. A process is regular if it is bisimilar to some finite-state one. Regularity has been intensively studied and there are several positive results in some classes of process algebras. Jančar and Esparza proved in [JE96] that regularity is decidable for

$$\begin{array}{rcll} E+F & = & F+E & \text{(A1)} \\ E+(F+G) & = & (E+F)+G & \text{(A2)} \\ E+E & = & E & \text{(A3)} \\ (E+F).G & = & E.G+F.G & \text{(A4)} \\ E.(F.G) & = & (E.F).G & \text{(A5)} \end{array}$$

$$\begin{array}{rcll} \delta.E & = & \delta & \text{(B1)} \\ \delta+E & = & E & \text{(B2)} \end{array}$$

Figure 2: BPA and BPA_δ laws

labelled Petri nets. Consequently, it is also decidable for BPP processes. Regularity appeared to be decidable in the class of normed PA processes even in polynomial time – result achieved by Kučera in [Kuč96]. A recent result [Jan97] due to Jančar says that regularity is decidable for one-counter processes. Burkart, Caucal and Steffen demonstrated in [BCS96] that regularity is decidable in the class we are interested in – the class of BPA systems (even unnormed).
At this place we give the definition of regular BPA systems. The definition of BPA_δ regularity we delay to the Section 5 where we also show that decidability of regularity extends to BPA_δ systems.

Definition 9. *A BPA system Δ is* regular *iff there is a BPA system Δ' with finitely many reachable states such that $\Delta \sim \Delta'$.*

It is obvious that a process is regular iff it can reach only finitely many states up to bisimilarity.

2.2 Axiomatisation of bisimulation equivalence

In the usual presentation of BPA (see e.g. [BK88]) much effort is usually paid to so–called BPA laws. These laws together with BPA_δ laws can be seen in Figure 2. The BPA and BPA_δ laws are easily shown to be sound w.r.t. bisimilarity, irrespective of any restrictions on the involved processes.

Lemma 3. *[BK88] For any BPA expressions E, F and G we have that $E+F \sim F+E$, $E+(F+G) \sim (E+F)+G$, $E+E \sim E$, $(E+F).G \sim E.G+F.G$ and $E.(F.G) \sim (E.F).G$.*

The BPA laws do not form a complete axiomatisation of BPA systems. Some notion of fixed–point induction must be added to prove equations of recursively defined systems. Details can be found in [Hüt91].

Lemma 4. *For any BPA_δ expression E we have that $\delta.E \overset{s}{\sim} \delta$, $\delta.E \overset{n}{\sim} \delta$, $\delta + E \overset{s}{\sim} E$ and $\delta + E \overset{n}{\sim} E$.*

Proof: The proof is immediate. $\{(\delta.E, \delta)\}$ is both strict and nonstrict bisimulation relation and thus we have $\delta.E \overset{s}{\sim} \delta$ and $\delta.E \overset{n}{\sim} \delta$. Similarly we know that $E \overset{s}{\sim} E$ and $E \overset{n}{\sim} E$ so there is both strict and nonstrict bisimulation relation R such that $(E, E) \in R$. The union $R \cup \{(\delta + E, E)\}$ remains to be bisimulation which implies that $\delta + E \overset{s}{\sim} E$ and $\delta + E \overset{n}{\sim} E$. $\square$

2.3 Greibach normal form

Definition 10. *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 is a natural number ($m > 0$), $a_j \in \mathcal{A}ct(\Delta)$ and $\alpha_j \in \mathcal{V}ar(\Delta)^*$. If $length(\alpha_j) < k$ for each j, $1 \leq j \leq m$, then Δ is said to be in k–GNF.

The normal form is called Greibach normal form by analogy with context-free grammars in Greibach normal form. The proof of the next theorem is based on the proof of 3–GNF for BPA systems that can be found e.g. in [Hüt91, BBK93, BBK87, HM96]. The construction shown in [Hüt91] had to be modified to capture the behaviour of deadlocks.

Theorem 1. *Let Δ be a guarded BPA_δ system. We can effectively find a BPA_δ system Δ' in 3–GNF such that $\Delta' \overset{s}{\sim} \Delta$ resp. $\Delta' \overset{n}{\sim} \Delta$. Moreover, if Δ is normed then so is Δ'.*

Proof: An effective algorithm (working in polynomial time) for rewriting Δ into 3–GNF consists from transforming Δ into GNF and then from rewriting the system into 3–GNF.

First, apply the laws (A4), (B1) and (B2) in Figure 2 (from left to right) as far as possible. Notice that these reductions are strongly normalising.

Now we replace all internal occurences of atomic actions by equations. Let us define inductively two mappings $f, g : \mathcal{E}_{\text{BPA}} \longrightarrow \mathcal{E}_{\text{BPA}}$, according to the following prescription.

$$
\begin{array}{rlll}
 & f(\epsilon) = g(\epsilon) = \epsilon & & \\
\text{for } a \in \mathcal{A}ct(\Delta): & f(a) = a & g(a) = X_a & \text{where } X_a \text{ is a fresh variable} \\
\text{for } X \in \mathcal{V}ar(\Delta): & f(X) = g(X) = X & & \\
\text{for } E, F \in \mathcal{E}_{\text{BPA}}: & f(E+F) = f(E) + f(F) & g(E+F) = g(E) + g(F) & \\
\text{for } E, F \in \mathcal{E}_{\text{BPA}}: & f(E.F) = f(E).g(F) & g(E.F) = g(E).g(F) &
\end{array}
$$

Let us transform every equation $X \stackrel{\text{def}}{=} E \in \Delta$ into $X \stackrel{\text{def}}{=} f(E)$ and add $X_a \stackrel{\text{def}}{=} a$ for each fresh variable X_a introduced by f. Notice that the system remains guarded and both strictly and nonstrictly bisimilar to the previous one. Moreover, due to applying the axiom (A4), every equation in Δ is now of the form:

$$X \stackrel{\text{def}}{=} \sum_i a_i \alpha_i (E_i + F_i) + \sum_j a_j \alpha_j$$

where $a_i, a_j \in \mathcal{A}ct$ are actions, $\alpha_i, \alpha_j \in \mathcal{V}ar^*$ are sequences of variables and $E_i, F_i \in \mathcal{E}_{\text{BPA}}$. Moreover E_i, F_i are built only from variables, i.e. $g(E_i) = E_i$ and $g(F_i) = F_i$.
In what follows let the symbols $a \in Act$, $\alpha \in \mathcal{V}ar^*$ and $E, F, E' \in \mathcal{E}_{\text{BPA}}$ range over their appropriate domains.

Example 2. *Let us have the following system.*

$$\{X \stackrel{\text{def}}{=} (acX + b + \delta).(a + b\delta X)\}$$

After the application of (A4), (B1) and (B2) we get

$$\{X \stackrel{\text{def}}{=} acX.(a + b\delta) + b.(a + b\delta)\}$$

and finally using the mapping f we transform the system into

$$\{X \stackrel{\text{def}}{=} aX_cX.(X_a + X_b\delta) + b.(X_a + X_b\delta),$$

$$X_a \stackrel{\text{def}}{=} a, \quad X_b \stackrel{\text{def}}{=} b, \quad X_c \stackrel{\text{def}}{=} c\}.$$

In each equation and for all the summands of the form $a\alpha(E+F)$ (we call the sum $(E+F)$ as an *unresolved sum*) introduce a fresh variable X_{E+F}. Replace this summand with $a\alpha X_{E+F}$ and introduce a new equation $X_{E+F} \stackrel{\text{def}}{=} E+F$.

Now all the 'old' equations are in GNF (i.e. every summand in such an equation is of the form $a\alpha$). However, in the definition of some 'new' variable X_{E+F} an unguarded summand of the form YE', $Y \in \mathcal{V}ar$ could have been introduced. At this state the defining equation for Y must be in GNF and assuming that the axioms (B1) and (B2) were applied as far as possible, we get $Y \neq \delta$. That means that

$$Y \stackrel{\text{def}}{=} \sum_i b_i\beta_i$$

and using the axiom (A4) we can

replace the summand YE' with $\sum_i b_i\beta_i E'$.

Again the system is (strictly and nonstrictly) bisimilar to the former one, all the equations are guarded and the number of different unresolved sums decreased. Repeat this procedure until there are no unresolved sums.

The resulting system is now in k–GNF for some $k > 0$. We will transform the system into 3–GNF. Each summand with even variable sequence length of the form

$$aX_1X_2X_3X_4\ldots X_{2m}$$

replace with

$$aU_{X_1X_2}U_{X_3X_4}\ldots U_{X_{2m-1}X_{2m}} \text{ for } m > 0.$$

Each summand with odd variable sequence length of the form

$$aX_1X_2X_3X_4\ldots X_{2m+1}$$

replace with

$$aU_{X_1X_2}U_{X_3X_4}\ldots U_{X_{2m-1}X_{2m}}X_{2m+1} \text{ for } m > 0.$$

The variables $U_{X_iX_j}$ are fresh variables with defining equations $U_{X_iX_j} \stackrel{\text{def}}{=} X_iX_j$. These equations are unguarded and we use again the same trick as before. We know that X_i is of the form $X_i \stackrel{\mathrm{def}}{=} \sum_p a_p\alpha_p$. The application of the axiom (B1) ensures $X_i \neq \delta$. So we can

replace X_iX_j in definition of $U_{X_iX_j}$ with $\sum_p a_p\alpha_p X_j$.

If some summand of $U_{X_iX_j}$ is of the form $a_p\alpha_p X_j$ with $\alpha_p = U_{\dots} \dots U_{\dots} X_{2m+1}$ we need to introduce a fresh variable $U_{X_{2m+1}X_j} \stackrel{\mathrm{def}}{=} X_{2m+1}X_j$ and make the equation guarded. Notice that a new variable $U_{X_{i'}X_{j'}}$ comprises just a pair of 'old' variables (i.e. variables different from $U_{\dots}$). There are only finitely many 'old' variables so the procedure must finish.
Observe, the system is in $\lceil k/2 \rceil + 1$ – GNF. Repeat this procedure until the resulting system is in 3–GNF.
We have constructed a BPA$_\delta$ system Δ' such that $\Delta' \stackrel{s}{\sim} \Delta$ and $\Delta' \stackrel{n}{\sim} \Delta$. Moreover, if Δ is normed, so is Δ' – from the construction. $\square$

We may assume that we are working only with BPA$_\delta$ systems in GNF since it has been proved that any BPA$_\delta$ (and also BPA) system can be effectively presented in 3–GNF and this construction preserves bisimilarity. This justifies also the assumption that all reachable states of the given BPA or BPA$_\delta$ system are elements of $\mathcal{V}ar^*$.

3 Expressibility of BPA$_\delta$ 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 than in the case of BPA.

Theorem 2. *There exists a BPA$_\delta$ system such that no BPA system is strictly bisimilar to it.*

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

Theorem 3. *There exists a BPA$_\delta$ system such that no BPA system is nonstrictly bisimilar to it.*

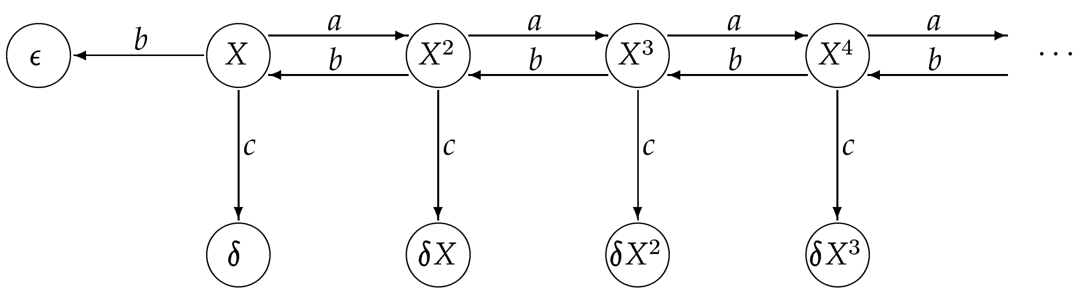

Figure 3: Labelled transition system for $X \stackrel{\text{def}}{=} aXX + b + c\delta$

Proof: We define a BPA$_\delta$ system Δ and show that there is no BPA system Δ' such that $\Delta \stackrel{n}{\sim} \Delta'$. Consider $\Delta = \{X \stackrel{\text{def}}{=} aXX + b + c\delta\}$ (see Figure 3) and suppose that there is a BPA system Δ' in 3–GNF, $\Delta' = \{Y_i \stackrel{\text{def}}{=} E_i \mid i = 1, \ldots, n\}$, such that $\Delta \stackrel{n}{\sim} \Delta'$. 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 a reachable state E from Δ' such that $X^n \stackrel{n}{\sim} E$. 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. $\square$

We show that the classes of BPA systems and BPA$_\delta$ systems are equivalent w.r.t. language generation. We will consider just the nonstrict case ($F = \{\epsilon, \delta\} \cup \{\delta.E | E \in \mathcal{E}^+_{\text{BPA}}\}$) since it is obvious that the strict case does not yield any language extention.

Definition 11. *Let $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ be a BPA$_\delta$ system. We define the* language generated by Δ *as $L(\Delta) \stackrel{\text{def}}{=} L(X_1)$. (For the definition of $L(X_1)$ see page 3.)*

Definition 12. *We define classes of languages generated by BPA and BPA$_\delta$ systems as following:*

$$\mathcal{L}(BPA) = \{L(\Delta) \mid \Delta \text{ is a BPA system }\}$$

$$\mathcal{L}(BPA_\delta) = \{L(\Delta_\delta) \mid \Delta_\delta \text{ is a BPA}_\delta \text{ system }\}$$

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

Proof: We show that for a BPA$_\delta$ 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$_\delta$ system in 3–GNF. The variables of the system Δ will be $\mathcal{V}ar(\Delta) \stackrel{\text{def}}{=} \bigcup_{X \in \mathcal{V}ar(\Delta_\delta)-\{\delta\}} \{X^\epsilon, X^\delta\} \cup \{X_1^{\epsilon\delta}\}$ 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):

$$\textbf{(a)}\ aAB \qquad \textbf{(b)}\ bC \qquad \textbf{(c)}\ c \qquad \textbf{(d)}\ dD\delta \qquad \textbf{(e)}\ e\delta \tag{1}$$

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:

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

if $X_1^\epsilon \stackrel{\text{def}}{=} E$ and $X_1^\delta \stackrel{\text{def}}{=} F$ 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 3. *Let us have a BPA$_\delta$ 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 13. *Let Δ' be a BPA (resp. BPA$_\delta$) system in 3–GNF, $n \geq 1$ and $Y \in \mathcal{V}ar(\Delta')$. We define $L_n^\epsilon(Y)$ and $L_n^\delta(Y)$ as following:*

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

Lemma 5. *For all $n \geq 1$ and $X \in \mathcal{V}ar(\Delta_\delta) - \{\delta\}$ holds that $L_n^\epsilon(X) = L_n^\epsilon(X^\epsilon)$ and $L_n^\delta(X) = L_n^\epsilon(X^\delta)$.*

Proof: By induction on n following the cases from 1.

- **n = 1**: There are only two cases to be considered. If the process X terminates in ϵ we have to consider the case (c) and if it terminates in $\delta\alpha$ we have to consider the case (e). Both these cases are obvious.

- **induction step**: Let us suppose that the assertion is true for all $i \leq n$. Let us prove it for $n + 1$.

  1. $L_{n+1}^\epsilon(X) \subseteq L_{n+1}^\epsilon(X^\epsilon)$
     Let $w \in L_{n+1}^\epsilon(X)$ and $length(w) = n + 1$. There are several cases according to the first action being performed:

     – **case (a)**: Suppose $w = aw_1w_2$ where $a \in \mathcal{A}ct$ and $w_1, w_2 \in \mathcal{A}ct^+$ such that $X \xrightarrow{a} AB \xrightarrow{w_1} B \xrightarrow{w_2} \epsilon$. According to the definition of X^ϵ and using the IH applied to variables A and B for the words w_1 and w_2 (that are sharply shorter than w) we get that $X^\epsilon \xrightarrow{a} A^\epsilon B^\epsilon \xrightarrow{w_1} B^\epsilon \xrightarrow{w_2} \epsilon$ and so $w \in L_{n+1}^\epsilon(X^\epsilon)$. For the rest of this proof all the conditions clear from the context will be omitted.

     – **case (b)**: $w = bw_1$ and $X \xrightarrow{b} C \xrightarrow{w_1} \epsilon$ but then $X^\epsilon \xrightarrow{b} C^\epsilon \xrightarrow{w_1} \epsilon$.

     – **case (c)**: This case is impossible because we suppose that $length(w) \geq 2$.

     – **cases (d), (e)**: These cases are impossible because in this inclusion we consider only $X \xrightarrow{w} \epsilon$ and it is clear that both $dD\delta$ and $e\delta$ are not able to reach the state ϵ.

2. $L^{\epsilon}_{n+1}(X) \supseteq L^{\epsilon}_{n+1}(X^{\epsilon})$
   Let $w \in L^{\epsilon}_{n+1}(X^{\epsilon})$ and $length(w) = n+1$. There are several cases according to the first action being performed:

   - **case (a)**: Suppose $w = aw_1w_2$ and $X^{\epsilon} \xrightarrow{a} A^{\epsilon}B^{\epsilon} \xrightarrow{w_1} B^{\epsilon} \xrightarrow{w_2} \epsilon$ but then $X \xrightarrow{a} AB \xrightarrow{w_1} B \xrightarrow{w_2} \epsilon$. So $w \in L^{\epsilon}_{n+1}(X)$.
   - **case (b)**: $w = bw_1$ and $X^{\epsilon} \xrightarrow{b} C^{\epsilon} \xrightarrow{w_1} \epsilon$ but then $X \xrightarrow{b} C \xrightarrow{w_1} \epsilon$. So $w \in L^{\epsilon}_{n+1}(X)$.
   - **cases (c), (d), (e)**: These cases are impossible.

3. $L^{\delta}_{n+1}(X) \subseteq L^{\epsilon}_{n+1}(X^{\delta})$
   Let $w \in L^{\delta}_{n+1}(X)$ and $length(w) = n+1$. There are several cases according to the first action being performed:

   - **case (a)**: Suppose $w = aw_1$ and $X \xrightarrow{a} AB \xrightarrow{w_1} \delta\alpha$. Then $A \xrightarrow{w_1} \delta\alpha'$ with $\alpha = \alpha' B$ or $w_1 = w_2w_3$ with $AB \xrightarrow{w_2} B \xrightarrow{w_3} \delta\alpha$. For the first subcase we have $X^{\delta} \xrightarrow{a} A^{\delta} \xrightarrow{w_1} \epsilon$ according to the definition of X^{δ} and because of the IH used on A^{δ}. For the second subcase we get $X^{\delta} \xrightarrow{a} A^{\epsilon}B^{\delta} \xrightarrow{w_2} B^{\delta} \xrightarrow{w_3} \epsilon$ because $A \xrightarrow{w_2} \epsilon$ and using the IH we deduce that $A^{\epsilon} \xrightarrow{w_2} \epsilon$ and similarly for B^{δ}. So we have that $w \in L^{\epsilon}_{n+1}(X^{\delta})$.
   - **case (b)**: $w = bw_1$ and $X \xrightarrow{b} C \xrightarrow{w_1} \delta\alpha$ but then $X^{\delta} \xrightarrow{b} C^{\delta} \xrightarrow{w_1} \epsilon$ and so $w \in L^{\epsilon}_{n+1}(X^{\delta})$.
   - **case (d)**: $w = dw_1$ and $X \xrightarrow{d} D\delta \xrightarrow{w_1} \delta\alpha$. Then $D \xrightarrow{w_1} \epsilon$ with $\alpha = \epsilon$ or $D \xrightarrow{w_1} \delta\alpha'$ such that $\alpha = \alpha'\delta$. It is easy to see that for both these subcases we get $X^{\delta} \xrightarrow{d} D^{\epsilon} \xrightarrow{w_1} \epsilon$ or $X^{\delta} \xrightarrow{d} D^{\delta} \xrightarrow{w_1} \epsilon$. This implies that $w \in L^{\epsilon}_{n+1}(X^{\delta})$.
   - **cases (c), (e)**: These cases are impossible.

4. $L^{\delta}_{n+1}(X) \supseteq L^{\epsilon}_{n+1}(X^{\delta})$
   Let $w \in L^{\epsilon}_{n+1}(X^{\delta})$ and $length(w) = n+1$. There are several cases according to the first action being performed:

   - **case (a)** : First suppose $w = aw_1w_2$ and $X^{\delta} \xrightarrow{a} A^{\epsilon}B^{\delta} \xrightarrow{w_1} B^{\delta} \xrightarrow{w_2} \epsilon$. Then $X \xrightarrow{a} AB \xrightarrow{w_1} B \xrightarrow{w_1} \delta\alpha$ and so $w \in L^{\delta}_{n+1}(X)$. Second suppose $w = aw_1$ and $X^{\delta} \xrightarrow{a} A^{\delta} \xrightarrow{w_1} \epsilon$ but then $X \xrightarrow{a} AB \xrightarrow{w_1} \delta\alpha B$ and $w \in L^{\delta}_{n+1}(X)$ as well.

- **case (b)**: $w = bw_1$ and $X^\delta \xrightarrow{b} C^\delta \xrightarrow{w_1} \epsilon$ but then $X \xrightarrow{b} C \xrightarrow{w_1} \delta\alpha$ and so $w \in L^\delta_{n+1}(X)$.
- **case (d)**: First suppose $w = dw_1$ and $X^\delta \xrightarrow{d} D^\epsilon \xrightarrow{w_1} \epsilon$ but then $X \xrightarrow{d} D\delta \xrightarrow{w_1} \delta$ and second suppose $w = dw_1$ and $X^\delta \xrightarrow{d} D^\delta \xrightarrow{w_1} \epsilon$ but then $X \xrightarrow{d} D\delta \xrightarrow{w_1} \delta\alpha\delta$ and so $w \in L^\delta_{n+1}(X)$.
- **cases (c), (e)**: These cases are impossible.

$\square$

To finish the proof of our theorem let us define for $n \geq 1$ the set $L_n(Y) \stackrel{\text{def}}{=} \{w \in L(Y) \mid \mathit{length}(w) \leq n\}$. Notice that because of the Lemma 5 we get $L_n(X_1) = L^\epsilon_n(X_1) \cup L^\delta_n(X_1) = L^\epsilon_n(X^\epsilon_1) \cup L^\epsilon_n(X^\delta_1) = L_n(X^{\epsilon\delta}_1)$ for all $n \geq 1$.
Now it is clear that $L(X_1) = L(X^{\epsilon\delta}_1)$ since if $w \in L(X_1)$ then $\exists n : w \in L_n(X_1)$ and so $w \in L_n(X^{\epsilon\delta}_1)$ which implies that $w \in L(X^{\epsilon\delta}_1)$. The other direction is similar. We have shown that $L(\Delta_\delta) = L(\Delta)$ and our proof is complete.

$\square$

4 Bisimilarity in BPA$_\delta$ systems

The first result indicating that decidability issues for bisimilarity are rather different from the ones for language equivalence is due to Baeten, Bergstra and Klop. They proved in [BBK87, BBK93] that bisimilarity is decidable for normed BPA systems. Much simpler proofs of this were later given in [Cau88],[HS91] and [Gro92].
It is well known result by Christensen, Hüttel and Stirling that the bisimulation equivalence is decidable in the class of all BPA systems – [CHS92]. The proof consists of two semidecidable procedures running in parallel. Burkart, Caucal and Steffen demonstrated in [BCS95] also an elementary decision procedure for BPA bisimilarity.
On the contrary the language equivalence of BPA processes is undecidable. The negative result for BPA [BHPS61] follows from the fact that BPA effectively defines the class of context-free languages. This argument can be shown to hold for the class of normed BPA systems as well. This undecidability result extends also to all equivalences which lie in

Glabbeek's spectrum [vG90b] between bisimilarity and language equivalence [GH94, HT95]. Another result [Jan95] due to Jančar says that bisimilarity is undecidable for Petri Nets.
We generalise the approach of Bosscher [Bos97] and show that the decidability of (strict and nonstrict) bisimilarity in BPA systems extends to BPA$_\delta$ systems. In the proof we exploit the result in [CHS92] and transform the examined BPA$_\delta$ systems into BPA systems, interpreting δ as a new unnormed variable. In this section w.l.o.g. we implicitly assume that all considered systems are in 3–GNF.

4.1 Decidability of nonstrict bisimilarity

Theorem 5. *Let $\mathcal{T} = (\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ and $\overline{\mathcal{T}} = (\overline{\mathcal{V}ar}, \overline{\mathcal{A}ct}, \overline{\Delta}, \overline{X_1})$ be BPA$_\delta$ systems. Then it is decidable whether $\mathcal{T} \overset{n}{\sim} \overline{\mathcal{T}}$.*

Proof: We reduce this problem to the problem of decidability of bisimilarity in BPA systems. We simply substitute the deadlock δ with a fresh unnormed variable.
Let us fix a fresh variable D such that $D \notin \mathcal{V}ar \cup \overline{\mathcal{V}ar}$ and an action d such that $d \notin \mathcal{A}ct \cup \overline{\mathcal{A}ct}$. We define a homomorphism $f : \mathcal{E}_{\text{BPA}} \longrightarrow \mathcal{E}_{\text{BPA}}$ as follows:

$f(a) = a$ for $a \in \mathcal{A}ct \cup \overline{\mathcal{A}ct}$
$f(X) = X$ for $X \in (\mathcal{V}ar \cup \overline{\mathcal{V}ar}) - \{\delta\}$
$f(\delta) = D$
$f(E + F) = f(E) + f(F), f(E.F) = f(E).f(F)$ for $E, F \in \mathcal{E}^+_{\text{BPA}}$

Let us define the systems $\mathcal{T}'$ and $\overline{\mathcal{T}'}$ as

$$\mathcal{T}' = (\mathcal{V}ar \cup \{D, X_1'\}, \mathcal{A}ct \cup \{d\}, \Delta', X_1')$$
$$\overline{\mathcal{T}'} = (\overline{\mathcal{V}ar} \cup \{D, \overline{X_1'}\}, \overline{\mathcal{A}ct} \cup \{d\}, \overline{\Delta'}, \overline{X_1'})$$

where, assuming that $(X_1 \overset{\text{def}}{=} E_1) \in \Delta$ and $(\overline{X_1} \overset{\text{def}}{=} \overline{E_1}) \in \overline{\Delta}$, we state

$$\Delta' = \{X_i \overset{\text{def}}{=} f(E_i) | X_i \overset{\text{def}}{=} E_i \in \Delta\} \cup \{X_1' \overset{\text{def}}{=} f(E_1).D, D \overset{\text{def}}{=} d.D\}$$
$$\overline{\Delta'} = \{\overline{X_i} \overset{\text{def}}{=} f(\overline{E_i}) | \overline{X_i} \overset{\text{def}}{=} \overline{E_i} \in \overline{\Delta}\} \cup \{\overline{X_1'} \overset{\text{def}}{=} f(\overline{E_1}).D, D \overset{\text{def}}{=} d.D\}.$$

The systems $\mathcal{T}'$ and $\overline{\mathcal{T}'}$ are now very similar to the previous ones except for the case when the systems reach the empty process (ϵ) or the deadlock (δ

or $\delta.G$ where $G \in \mathcal{E}^{+}_{\text{BPA}}$). The behaviour in these states is changed to capture the property that the empty process is nonstrict bisimilar to the deadlock. A new unnormed variable D is added to simulate these states.
It is easy to see that $\mathcal{T} \overset{n}{\sim} \overline{\mathcal{T}}$ if and only if $\mathcal{T}' \sim \overline{\mathcal{T}'}$. Moreover the systems $\mathcal{T}'$ and $\overline{\mathcal{T}'}$ are BPA systems and bisimulation is decidable in the class BPA (see [CHS92]). Thus we can also decide whether $\mathcal{T} \overset{n}{\sim} \overline{\mathcal{T}}$.

□

Example 4. *Let $\Delta = \{X \overset{\text{def}}{=} aXX + b + c\delta\}$. The system Δ' from the proof above is following.*

$$\Delta' = \{X' \overset{\text{def}}{=} (aXX + b + cD).D,\ X \overset{\text{def}}{=} aXX + b + cD,\ D \overset{\text{def}}{=} d.D\}$$

4.2 Decidability of strict bisimilarity

Theorem 6. *Let $\mathcal{T} = (\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ and $\overline{\mathcal{T}} = (\overline{\mathcal{V}ar}, \overline{\mathcal{A}ct}, \overline{\Delta}, \overline{X_1})$ be BPA$_\delta$ systems. Then it is decidable whether $\mathcal{T} \overset{s}{\sim} \overline{\mathcal{T}}$.*

Proof: The proof is quite easy because for the strict bisimilarity we have that $\delta \overset{s}{\not\sim} \epsilon$ and we can use the slightly modified trick from the proof above. We construct the same systems $\mathcal{T}'$ and $\overline{\mathcal{T}'}$ as before with one difference. The leading variables of the systems $\mathcal{T}'$ and $\overline{\mathcal{T}'}$ will remain X_1 and $\overline{X_1}$, however we do not add the new equations $X'_1 \overset{\text{def}}{=} f(E_1).D$ and $\overline{X'_1} \overset{\text{def}}{=} f(\overline{E_1}).D$. This ensures that in the newly defined systems (which are BPA systems) we can possibly reach the empty process. This empty process is not bisimilar to the state D (nor $D.G$ for $G \in \mathcal{E}^{+}_{\text{BPA}}$) simulating deadlocking. □

5 Regularity in BPA$_\delta$ systems

Regularity of a transition system means in fact finiteness of the number of states up to bisimilarity. If we prove that a transition system can be expressed (up to bisimilarity) as a finite state system and that the construction is effective, we can decide all the interesting properties within such a regular system. Burkart, Caucal and Steffen demonstrated in [BCS96] that regularity is decidable for BPA processes and we exploit this result, thus extending the decidability to the BPA$_\delta$ systems. This section again generalises the results by Bosscher [Bos97].

Defining regularity of a BPA system is not difficult. We state a BPA system Δ to be *regular* iff it is bisimilar to a BPA system with finitely many reachable states. But in the case of BPA_δ we introduced two notions of bisimilarity (strict and nonstrict) and moreover we may consider regularity with regard to finite state BPA and BPA_δ system. There is no sense to consider strict bisimilarity w.r.t. finite state BPA. The nonstrict case is solved by the following lemma.

Lemma 6. *Let Δ be a BPA_δ system with finitely many reachable states. Then there exists a BPA system Δ' with finitely many reachable states such that $\Delta \overset{n}{\sim} \Delta'$.*

Proof: We can assume that the process Δ is in normal form, i.e. every equation is of the form:

$$X_i \stackrel{\mathrm{def}}{=} \sum_j a_j X_j + \sum_k a_k,$$

where X_j can possibly be δ. This can be done because if there are only finitely reachable states, we give a special new name to every such a state. The set of variables will be formed with the names of these states and we add corresponding transitions. This trivially preserves nonstrict bisimilarity (the resulting transition systems are even isomorphic). We construct the system Δ' from Δ by deleting all occurences of δ in each defining equation. The systems Δ and Δ' are easily seen to be nonstrictly bisimilar. $\square$

When dealing with regularity we give two definitions for the case of strict and nonstrict bisimilarity. The second one is motivated by the Lemma 6 above.

Definition 14. *A BPA_δ system Δ is* strictly regular *iff there exists a BPA_δ system Δ' with finitely many reachable states such that $\Delta \overset{s}{\sim} \Delta'$.*

Definition 15. *A BPA_δ system Δ is* nonstrictly regular *iff there exists a BPA system Δ' with finitely many reachable states such that $\Delta \overset{n}{\sim} \Delta'$.*

We show that both strict and nonstrict regularity is decidable in the class BPA_δ and thus extend the results from [BCS96] and [Bos97].

5.1 Decidability of strict regularity

Theorem 7. *Let Δ be a BPA$_\delta$ system. It is decidable whether Δ is strictly regular. If it is the case, the corresponding finite state BPA$_\delta$ system can be effectively constructed.*

Proof: We use again the trick from the proof of the Theorem 6. We reduce the problem to the problem of decidability of regularity in the BPA class. As in the proof above we transform the system Δ into Δ' such that all occurences of δ are replaced with a fresh variable D and a new defining equation for D, $D \stackrel{\text{def}}{=} d.D$, is added where $d \in \mathcal{A}ct$ is a fresh action. Now it is obvious that Δ' is regular (in the sense of BPA systems) if and only if Δ is strictly regular. Since regularity in the class of BPA systems is decidable (see [BCS96]), the strict regularity in the BPA$_\delta$ systems is also decidable. Moreover the corresponding finite state BPA$_\delta$ system can be easily constructed as we can find a finite state BPA system Δ'' in normal form, such that $\Delta'' \sim \Delta'$. It is enough to replace all occurences of each variable bisimilar to D with δ and remove definitions of such variables. $\square$

Example 5. *Let us have a BPA$_\delta$ system*

$$\Delta = \{A \stackrel{\text{def}}{=} aBA\delta + a\delta,\ B \stackrel{\text{def}}{=} bAB + b\delta\}.$$

After the transformation we get

$$\Delta' = \{A \stackrel{\text{def}}{=} aBAD + aD,\ B \stackrel{\text{def}}{=} bAB + bD,\ D \stackrel{\text{def}}{=} aD\}.$$

This BPA system is regular and the bisimilar finite state system in normal form is e.g.

$$\Delta'' = \{A' \stackrel{\text{def}}{=} aB' + aD',\ B' \stackrel{\text{def}}{=} bA' + bD',\ D' \stackrel{\text{def}}{=} aD'\}.$$

By replacing D' with δ ($D' \sim D$) we get

$$\Delta''' = \{A' \stackrel{\text{def}}{=} aB' + a\delta,\ B' \stackrel{\text{def}}{=} bA' + b\delta\}$$

such that $\Delta \stackrel{s}{\sim} \Delta'''$.

5.2 Decidability of nonstrict regularity

For the proof of the nonstrict case we use the following lemma where we show that strict and nonstrict regularity coincide.

Lemma 7. *A BPA_δ system Δ is strictly regular iff Δ is nonstrictly regular.*

Proof: We prove the implication from left to right. Suppose that Δ is strictly regular, i.e. there exists a BPA_δ system Δ' with finitely many reachable states such that $\Delta \overset{s}{\sim} \Delta'$. Because of the Lemma 2 we know that $\Delta \overset{n}{\sim} \Delta'$ and using the Lemma 6 we can see that there exists a BPA system Δ'' with finitely many reachable states such that $\Delta' \overset{n}{\sim} \Delta''$. Thus we have shown that $\Delta \overset{n}{\sim} \Delta''$ which implies that Δ is nonstrictly regular.
The implication from right to left is a bit more complicated. Suppose that Δ is nonstrictly regular, i.e. there exists a BPA system Δ' with finitely many reachable states such that $\Delta \overset{n}{\sim} \Delta'$. W.l.o.g. we may assume that Δ' is in normal form introduced in the proof of the Lemma 6. Let X_1 and X_1' be leading variables of the systems Δ resp. Δ'. Then we know that there exists some relation of nonstrict bisimulation R such that $(X_1, X_1') \in R$. Let us modify the system Δ' into Δ'' following the rules below. For each $X \in \mathcal{V}ar(\Delta')$ and $a \in \mathcal{A}ct(\Delta')$:

- Remove all the summands of the form a from the definition of X.
- If $(E, X) \in R$ such that $E \xrightarrow{a} \delta$ or $E \xrightarrow{a} \delta.G$ for some $G \in \mathcal{E}^+_{\mathrm{BPA}}$ then add the summand $a\delta$ into the definition of X.
- If $(E, X) \in R$ such that $E \xrightarrow{a} \epsilon$ then add the summand a into the definition of X.

Let us define the relation S as following.

$$S \overset{\mathrm{def}}{=} (R - \{(\delta, \epsilon)\} - \{(\delta.G, \epsilon) \mid G \in \mathcal{E}^+_{\mathrm{BPA}}\}) \cup \{(\delta, \delta)\} \cup \{(\delta.G, \delta) \mid G \in \mathcal{E}^+_{\mathrm{BPA}}\}$$

Then obviously $(X_1, X_1') \in S$ and moreover we show that S is the relation of strict bisimulation. This implies that $\Delta \overset{s}{\sim} \Delta''$.
In fact we have removed all the inconvenient pairs from R and added all the deadlocking pairs. It is an easy observation that if $(\alpha, \beta) \in S$ then $\alpha \in F$ iff $\beta \in F$. This means that there is no collision between ϵ and δ any more.

- Let $(E, X) \in S$ and $a \in \mathcal{A}ct(\Delta'')$.
  - If $E \xrightarrow{a} E'$ such that $E' \neq \epsilon$ and $E' \neq \delta$ and $E' \neq \delta.G$ for all $G \in \mathcal{E}^{+}_{\text{BPA}}$ then $X \xrightarrow{a} X'$ such that $(E', X') \in R$ which implies that $(E', X') \in S$.
  - If $E \xrightarrow{a} \delta$ or $E \xrightarrow{a} \delta.G$ for some $G \in \mathcal{E}^{+}_{\text{BPA}}$ then $X \xrightarrow{a} \delta$ and $(\delta, \delta) \in S$ resp. $(\delta.G, \delta) \in S$.
  - If $E \xrightarrow{a} \epsilon$ then $X \xrightarrow{a} \epsilon$ and obviously $(\epsilon, \epsilon) \in S$.
- Let $(E, X) \in S$ and $a \in \mathcal{A}ct(\Delta'')$.
  - If $X \xrightarrow{a} X'$ such that $X' \neq \epsilon$ and $X' \neq \delta$ then $E \xrightarrow{a} E'$ such that $(E', X') \in R$ which implies that $(E', X') \in S$.
  - If $X \xrightarrow{a} \delta$ then $E \xrightarrow{a} \delta$ or $E \xrightarrow{a} \delta.G$ for some $G \in \mathcal{E}^{+}_{\text{BPA}}$ and we can see that $(\delta, \delta) \in S$ resp. $(\delta.G, \delta) \in S$.
  - If $X \xrightarrow{a} \epsilon$ then $E \xrightarrow{a} \epsilon$ and $(\epsilon, \epsilon) \in S$.

□

Theorem 8. *Let Δ be a BPA$_\delta$ system. It is decidable whether Δ is nonstrictly regular. If it is the case, the corresponding finite state BPA system can be effectively constructed.*

Proof: Using the Lemma 7 and the Theorem 7 we can decide whether Δ is nonstrictly regular since Δ is nonstrictly regular iff Δ is strictly regular. Moreover the first part in the proof of the Lemma 7 gives directions how to construct the corresponding finite state BPA system. □

6 Describing BPA$_\delta$ in BPA syntax

In the Section 3 we have shown that the class of BPA$_\delta$ systems is strictly larger (w.r.t. bisimilarity) than that of BPA. This challenges the question whether a given BPA$_\delta$ system can be equivalently described in BPA syntax. The answer for both strict and nonstrict bisimilarity taken as the equivalence relation is the topic of this section. The characterisation for the strict bisimulation is given by Theorem 9 and Theorem 12 demonstrates the corresponding result for the nonstrict bisimulation.

6.1 Strict case

Theorem 9. *Let $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ be a BPA_δ system. It is decidable whether there exists a BPA system Δ' such that $\Delta \overset{s}{\sim} \Delta'$. Moreover if the answer is positive, the system Δ' can be effectively constructed.*

Proof: The proof is rather technical and is based on the fact that $\delta \overset{s}{\not\sim} \epsilon$. Consider the system Δ. If a state of the form δ or $\delta.E$ for $E \in \mathcal{E}^{+}_{\text{BPA}}$ is reachable from the leading variable then there cannot be any BPA system bisimilar to Δ. If the deadlocking state is not reachable the system Δ can be easily transformed into a BPA system.
Suppose w.lo.g. that the system Δ is in 3–GNF. We construct the sets $M_0, M_1, \ldots$ of variables from which the deadlock is reachable as following. The notation $\alpha \in E$ means again that α is a summand in the expression E.

$$M_0 \overset{\text{def}}{=} \{\delta\}$$

And for $i \geq 0$ the sets M_{i+1} are defined as:

$$M_{i+1} \overset{\text{def}}{=} M_i \cup \{X \in \mathcal{V}ar \mid \exists a \in \mathcal{A}ct, \exists Y \in \mathcal{V}ar, \exists D \in M_i :$$

$$(X \overset{\text{def}}{=} E) \in \Delta,\ \ a.D \in E\ \vee\ a.D.Y \in E\ \vee\ (a.Y.D \in E\ \text{ and }\ \|Y\| < \infty)\}$$

We remind that the norm of a variable can be effectively computed. Since there are only finitely many variables used in the system Δ then for some $k \geq 0$ the set M_k is a fixed point of this construction, i.e. $M_k = M_{k+l}$ for each $l > 0$. Let us denote the set M_k simply as M.
Now we get an easy consequence clear from the construction of the sets M_i. For each $X \in \mathcal{V}ar$:

$$X \longrightarrow^* \delta.\alpha \ \text{ for some } \alpha \in \mathcal{V}ar^* \quad \Longleftrightarrow \quad X \in M$$

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 transform Δ into a BPA system. For this case realize that if $Y \in M$ then $X_1 \not\longrightarrow^* Y.\alpha$ for any $\alpha \in \mathcal{V}ar^*$. Let us define $(\mathcal{V}ar - M, \mathcal{A}ct, \Delta', X_1)$ where for each $(X \overset{\text{def}}{=} E) \in \Delta$ we have that $(X \overset{\text{def}}{=} E') \in \Delta'$ whenever $X \notin M$ and E' is the same as E except for the summand of the type $a.YD$ where $Y \in \mathcal{V}ar$ and $D \in M$, which is replaced with $a.Y$. This can be done because Y must be an unnormed variable, otherwise $X \in M$.

It is clear that Δ' is strictly bisimilar to Δ (only irredundant variables were disposed) and moreover Δ' is a BPA system – from the construction. $\square$

6.2 Nonstrict case

In this section we focus on those BPA$_\delta$ systems which can be described in corresponding BPA syntax w.r.t. nonstrict bisimilarity. The situation, when allowing deadlocks can bring more descriptive power, is nicely characterised by the Theorem 12.
We can simply observe that in a BPA$_\delta$ labelled transition system there are only finitely many successors of each state. In such case we call the system as *image-finite*.

Definition 16. *A labelled transition system* $(S, \mathcal{A}ct, \longrightarrow, \alpha_0, F)$ *is* image-finite *if the set* $\{\beta \mid \alpha \xrightarrow{a} \beta\}$ *is finite for each* $\alpha \in S$ *and* $a \in \mathcal{A}ct$.

Bisimilarity in such image-finite systems is characterisable using the following sequence of approximations.

Definition 17. *Let* $(S, \mathcal{A}ct, \longrightarrow, \alpha_0, F)$ *be a labelled transition system. The* stratified bisimulation relations *[Mil89]* $\sim_k$ *are defined as follows.*

- $\alpha \sim_0 \beta$ *for all* $\alpha, \beta \in S$
- $\alpha \sim_{k+1} \beta$ *iff for each* $a \in \mathcal{A}ct$*:*
  - *if* $\alpha \xrightarrow{a} \alpha'$ *then* $\beta \xrightarrow{a} \beta'$ *for some* β' *such that* $\alpha' \sim_k \beta'$
  - *if* $\beta \xrightarrow{a} \beta'$ *then* $\alpha \xrightarrow{a} \alpha'$ *for some* α' *such that* $\alpha' \sim_k \beta'$
  - $\alpha \in F$ *iff* $\beta \in F$

The following lemma is standard.

Lemma 8. *Let* $(S, \mathcal{A}ct, \longrightarrow, \alpha_0, F)$ *be an image-finite labelled transition system and* $\alpha, \beta \in S$*. Then* $\alpha \sim \beta$ *iff* $\alpha \sim_k \beta$ *for all* $k \geq 0$.

Remark. *In the case of BPA$_\delta$ systems and considering the nonstrict bisimilarity, the third condition* $\alpha \in F$ *iff* $\beta \in F$ *in the Definition 17 is always true since all the terminal states are included in* F.

In what follows, the set of variables from which the deadlock is reachable will be of great importance. Hence we define the set $\mathcal{V}ar_\delta$ of such variables.

Definition 18. *Let $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ be a BPA$_\delta$ system. Let us define the sets*

$$\mathcal{V}ar_\delta \stackrel{\text{def}}{=} \{X \in \mathcal{V}ar \mid X \longrightarrow^* \delta \ \text{ or } \ \exists E \in \mathcal{E}^+_{\text{BPA}} : X \longrightarrow^* \delta.E\} - \{\delta\}$$

$$\mathcal{V}ar_\epsilon \stackrel{\text{def}}{=} \mathcal{V}ar - \{\delta\} - \mathcal{V}ar_\delta.$$

This separates the variables from $\mathcal{V}ar$ into two sets $\mathcal{V}ar_\delta$ and $\mathcal{V}ar_\epsilon$ (i.e. $\mathcal{V}ar = \mathcal{V}ar_\delta \cup \mathcal{V}ar_\epsilon \cup \{\delta\}$). For the purpose of this section let the variables U, V, X, Y, Z range over $\mathcal{V}ar_\delta$ and A, B over $\mathcal{V}ar_\epsilon$.

Remark. *We remind that 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 9.*

Theorem 10. *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.
Let us 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 (possibly adding some new variables of the form X^β). 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 $\gamma \in \mathcal{V}ar^*$.

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

Let us now construct the nonstrictly bisimilar BPA system Δ' where

$$\mathcal{V}ar' \stackrel{\text{def}}{=} \mathcal{V}ar_\epsilon \cup \mathsf{Added},$$

$$\mathcal{A}ct' \stackrel{\text{def}}{=} \mathcal{A}ct,$$

$$\Delta' \stackrel{\text{def}}{=} \Delta_\epsilon \cup \Gamma,$$

$$X_1' \stackrel{\text{def}}{=} X_1^\epsilon.$$

The sets Added and Γ are outputs of the following algorithm and $\Delta_\epsilon \subseteq \Delta$ contains exactly the defining equations for variables from $\mathcal{V}ar_\epsilon$.

The transformation of the defining equations of the variables from $\mathcal{V}ar_\delta$ is the goal of the Algorithm 1. The set Solve contains the variables that need to be defined; Added is the set of variables that have been already defined or are in the set Solve; Γ is the set of the current definitions; Add is the set of variables born in each repetition of the main loop.

Algorithm 1.

1 Solve:=$\{X_1^\epsilon\}$
2 Added:=$\{X_1^\epsilon\}$
3 $\Gamma := \emptyset$
4 **while** Solve $\neq \emptyset$ **do**
5 *Let us fix* $X^\alpha \in$ Solve *with* $(X \stackrel{\text{def}}{=} E) \in \Delta$
6 $\Gamma := \Gamma \cup \{X^\alpha \stackrel{\text{def}}{=} f_\alpha(E)\}$
7 Add:=$\{Y^\beta \in r_\alpha(E) \mid \forall Z^\omega \in \text{Added} : Y\beta \stackrel{n}{\not\sim} Z\omega\}$
8 **while** $\exists Y^\beta, Z^\omega \in \text{Add} : Y^\beta \neq Z^\omega \;\wedge\; Y\beta \stackrel{n}{\sim} Z\omega$ **do**
9 Add:= Add $- \{Y^\beta\}$
10 **endwhile**
11 Solve:= (Solve $- \{X^\alpha\}$) $\cup$ Add
12 Added:= Added $\cup$ Add
13 **for** $\forall Y^\beta \in r_\alpha(E) -$ Add **do**
14 *replace all occurences of* Y^β *in* Γ *with* Z^ω
15 *where* $Z^\omega \in$ Added : $Y\beta \stackrel{n}{\sim} Z\omega$
16 **endfor**
17 **endwhile**

In the following lemmas we demonstrate that the algorithm is correct and produces a BPA system Δ' such that $\Delta \stackrel{n}{\sim} \Delta'$.

Lemma 9. *For the loop 4–17 of the Algorithm 1 holds the following invariant $\mathcal{I}$.*

$$\forall Y^\beta, Z^\omega \in \text{Added} : Y^\beta \neq Z^\omega \Rightarrow Y\beta \stackrel{n}{\not\sim} Z\omega$$

Proof: The invariant $\mathcal{I}$ holds at line 3, because the set Added contains just one variable. Some new variables can potentially be added to the set Added at line 12. Because of the loop 8–10 the variables in Add are pairwise nonstrictly nonbisimilar. Line 7 ensures that $\mathcal{I}$ will hold for Added:=Added$\cup$Add also. $\square$

Lemma 10. *Whenever during the execution of the Algorithm 1 we have $Y^\alpha \in$* Added *then $Y \in \mathcal{V}ar_\delta$.*

Proof: All variables in Added had to be produced by the function r_α (see line 7 and 12). It is an easy observation that $\{Y | Y^\beta \in r_\alpha(E)\} \subseteq \mathcal{V}ar_\delta$ for any $\alpha \in \mathcal{V}ar^*$ and $E \in \mathcal{E}^+_{\text{BPA}}$ such that E is in 3–GNF. $\square$

Lemma 11. *Whenever during the execution of the Algorithm 1 we have $Y^\beta \in$ 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 $X^\alpha \in$ 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 $\gamma \in \mathcal{V}ar_\epsilon^*$ and $\| \gamma \| < \infty$. From the construction of the function f_α we can also see that $X\alpha \longrightarrow^* Y\beta$. Thus we get $X_1 \longrightarrow^* X\alpha \longrightarrow^* Y\beta$.

$\square$

Lemma 12. *Under the assumptions of the Theorem 10 the Algorithm 1 cannot loop forever.*

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). From the Lemma 11 and Lemma 10 we know that $\forall Y^\beta \in \mathsf{Added} : Y \in \mathcal{V}ar_\delta \;\; \wedge \;\; X_1 \longrightarrow^* Y\beta$. Moreover from the Lemma 9 follows that these states are pairwise nonstrictly nonbisimilar. The contradiction is immediate as we have shown that if the algorithm loops then there is no upper bound on the cardinality of the set Added. $\square$

From the previous lemma we know that the Algorithm 1 will stop after finitely many repetitions of the main loop and thus the set Added will also be finite. The following lemma is crucial for the proof of our theorem.

Lemma 13. *After the execution of the Algorithm 1 we have $V^\alpha \stackrel{n}{\sim} V\alpha$ for all $V^\alpha \in \mathsf{Added}$.*

Proof: By induction on k we show that $V^\alpha \sim_k V\alpha$ for all $k \geq 0$. This implies that $V^\alpha \stackrel{n}{\sim} V\alpha$.
Basic step: We receive $V^\alpha \sim_0 V\alpha$ from the definition.
Induction step: We show that $V^\alpha \sim_{k+1} V\alpha$.

Suppose that $V^{\alpha} \xrightarrow{a} V'$. Then one of the following cases applies (according to the definition of f_{α}):

- Let us consider the summand aXY. Then one of the following cases will hold:
  - $V^{\alpha} \xrightarrow{a} X^{Y\alpha}$ but then $V\alpha \xrightarrow{a} XY\alpha$. Using the induction hypothesis we get $X^{Y\alpha} \sim_k XY\alpha$, because $X^{Y\alpha} \in \mathsf{Added}$.
  - $V^{\alpha} \xrightarrow{a} Z^{\omega}$ where $Z^{\omega} \in \mathsf{Added}$ and $X^{Y\alpha}$ was at lines 14,15 replaced with Z^{ω}. Then $Z^{\omega} \sim_k Z\omega$ (induction hypothesis) and $Z\omega \overset{n}{\sim} XY\alpha$. This implies that $V\alpha \xrightarrow{a} XY\alpha$ and $XY\alpha \sim_k Z\omega \sim_k Z^{\omega}$.
- Let us consider the summand $aX\delta$:
  - $V^{\alpha} \xrightarrow{a} X^{\epsilon}$ but then $V\alpha \xrightarrow{a} X\delta\alpha$. We know that $X\delta\alpha \overset{n}{\sim} X$ and using the induction hypothesis we get $X^{\epsilon} \sim_k X$ because $X^{\epsilon} \in \mathsf{Added}$. Thus we get $X\delta\alpha \sim_k X^{\epsilon}$.
  - $V^{\alpha} \xrightarrow{a} Z^{\omega}$ where $Z^{\omega} \in \mathsf{Added}$ and X^{ϵ} was at lines 14,15 replaced with Z^{ω}. Then $Z^{\omega} \sim_k Z\omega$ (induction hypothesis) and $Z\omega \overset{n}{\sim} X\delta\alpha$. This implies that $V\alpha \xrightarrow{a} X\delta\alpha$ and $X\delta\alpha \sim_k Z\omega \sim_k Z^{\omega}$.
- Let us consider the summand $a\delta$:
  - $V^{\alpha} \xrightarrow{a} \epsilon$ but then $V\alpha \xrightarrow{a} \delta\alpha$ and $\epsilon \sim_k \delta\alpha$ because trivially $\epsilon \overset{n}{\sim} \delta\alpha$.
- Let us consider the summand aX:
  - this is very similar to aXY
- Let us consider the summand aAB:
  - $V^{\alpha} \xrightarrow{a} AB\beta U^{\gamma}$ but then $V\alpha \xrightarrow{a} AB\beta U\gamma$. Using the induction hypothesis we know $U^{\gamma} \sim_k U\gamma$ because $U^{\gamma} \in \mathsf{Added}$ and we get $AB\beta U^{\gamma} \sim_k AB\beta U\gamma$.
  - $V^{\alpha} \xrightarrow{a} AB\beta Z^{\omega}$ where $Z^{\omega} \in \mathsf{Added}$ and U^{γ} was at lines 14,15 replaced with Z^{ω}. Then $Z^{\omega} \sim_k Z\omega$ (induction hypothesis) and $Z\omega \overset{n}{\sim} U\gamma$. This implies that $V\alpha \xrightarrow{a} AB\beta U\gamma$ and $AB\beta U\gamma \sim_k AB\beta Z^{\omega}$.

- $V^{\alpha} \xrightarrow{a} AB\beta C$ such that $\|C\|= \infty$ but then $V\alpha \xrightarrow{a} AB\beta C\gamma$ and easily $AB\beta C \sim_k AB\beta C\gamma$.
- $V^{\alpha} \xrightarrow{a} AB\alpha$ but then $V\alpha \xrightarrow{a} AB\alpha$ and obviously $AB\alpha \sim_k AB\alpha$.

* Let us consider the summands a and aA:
  - these are very similar to aAB
* Let us consider the summand $aA\delta$:
  - this is very similar to $a\delta$
* Let us consider the summand aXA:
  - this is very similar to aXY
* Let us consider the summand aAX:
  - $V^{\alpha} \xrightarrow{a} AX^{\alpha}$ but then $V\alpha \xrightarrow{a} AX\alpha$. Using the induction hypothesis we get $X^{\alpha} \sim_k X\alpha$ because $X^{\alpha} \in \mathtt{Added}$ and so $AX^{\alpha} \sim_k AX\alpha$.
  - $V^{\alpha} \xrightarrow{a} AZ^{\omega}$ where $Z^{\omega} \in \mathtt{Added}$ and X^{α} was at lines 14,15 replaced with Z^{ω}. Then $Z^{\omega} \sim_k Z\omega$ (induction hypothesis) and $Z\omega \overset{n}{\sim} X\alpha$. This implies that $V\alpha \xrightarrow{a} AX\alpha$ and $AX\alpha \sim_k AZ\omega \sim_k AZ^{\omega}$.

Suppose that $V\alpha \xrightarrow{a} V'$.Then one of the following cases applies (according to the definition of f_{α}):

* Let us consider the summand aXY. If $V\alpha \xrightarrow{a} XY\alpha$ then one of the following cases will hold:
  - $V^{\alpha} \xrightarrow{a} X^{Y\alpha}$, where $X^{Y\alpha} \in \mathtt{Added}$ and using the induction hypothesis we get $X^{Y\alpha} \sim_k XY\alpha$.
  - $V^{\alpha} \xrightarrow{a} Z^{\omega}$, where $Z^{\omega} \in \mathtt{Added}$ and $X^{Y\alpha}$ was at lines 14,15 replaced with Z^{ω}. Then $Z^{\omega} \sim_k Z\omega$ (induction hypothesis) and $Z\omega \overset{n}{\sim} XY\alpha$. This means that $Z^{\omega} \sim_k XY\alpha$.
* Let us consider the summand $aX\delta$. If $V\alpha \xrightarrow{a} X\delta\alpha$ then one of the following cases will hold:

- $V^{\alpha} \xrightarrow{a} X^{\epsilon}$, where $X^{\epsilon} \in \mathsf{Added}$ and using the induction hypothesis we get $X^{\epsilon} \sim_k X$ and so $X^{\epsilon} \sim_k X\delta\alpha$.
- $V^{\alpha} \xrightarrow{a} Z^{\omega}$, where $Z^{\omega} \in \mathsf{Added}$ and X^{ϵ} was at lines 14,15 replaced with Z^{ω}. Then $Z^{\omega} \sim_k Z\omega$ (induction hypothesis) and $Z\omega \overset{n}{\sim} X$. This means that $Z^{\omega} \sim_k X\delta\alpha$.

- Let us consider the summand $a\delta$. If $V\alpha \xrightarrow{a} \delta\alpha$ then
  - $V^{\alpha} \xrightarrow{a} \epsilon$ and $\epsilon \sim_k \delta\alpha$.
- Let us consider the summand aX:
  - this case is very similar to aXY
- Let us consider the summand aAB. If $V\alpha \xrightarrow{a} AB\alpha$ then one of the following cases will hold:
  - $V^{\alpha} \xrightarrow{a} AB\beta U^{\gamma}$, where $\alpha = \beta U\gamma$ and $U^{\gamma} \in \mathsf{Added}$. Using the induction hypothesis we get $U^{\gamma} \sim_k U\gamma$ and so $AB\beta U^{\gamma} \sim_k AB\alpha$.
  - $V^{\alpha} \xrightarrow{a} AB\beta Z^{\omega}$, where $\alpha = \beta U\gamma$, $Z^{\omega} \in \mathsf{Added}$ and U^{γ} was at lines 14,15 replaced with Z^{ω}. Then $Z^{\omega} \sim_k Z\omega$ (induction hypothesis) and $Z\omega \overset{n}{\sim} U\gamma$. This means that $AB\beta Z^{\omega} \sim_k AB\alpha$.
  - $V^{\alpha} \xrightarrow{a} AB\beta C$ such that $||C|| = \infty$ and $\alpha = \beta C\gamma$ but then $AB\beta C \sim_k AB\alpha$.
  - $V^{\alpha} \xrightarrow{a} AB\alpha$ but then $AB\alpha \sim_k AB\alpha$.
- Let us consider the summands a and aA:
  - these cases are very similar to aAB
- Let us consider the summands $aA\delta$:
  - this case is very similar to $a\delta$
- Let us consider the summands aXA:
  - this case is very similar to aXY
- Let us consider the summand aAX. If $V\alpha \xrightarrow{a} AX\alpha$ then one of the following cases will hold:

– $V^\alpha \xrightarrow{a} AX^\alpha$, where $X^\alpha \in$ Added and using the induction hypothesis we get $X^\alpha \sim_k X\alpha$ and so $AX^\alpha \sim_k AX\alpha$.

– $V^\alpha \xrightarrow{a} AZ^\omega$, where $Z^\omega \in$ Added and X^α was at lines 14,15 replaced with Z^ω. Then $Z^\omega \sim_k Z\omega$ (induction hypothesis) and $Z\omega \overset{n}{\sim} X\alpha$. This means that $AZ^\omega \sim_k AX\alpha$.

$\square$

Lemma 14. *The system Δ' is a BPA system and moreover $X_1 \overset{n}{\sim} X_1^\epsilon$.*

Proof: There are no undefined variables in Δ', which follows from the fact that each variable added into the set Added (line 12) had to be put into Solve (line 11) and so had to be expanded (line 6). Moreover observe that all δ's were removed by the function f_α. The fact $X_1 \overset{n}{\sim} X_1^\epsilon$ follows from the Lemma 13. $\square$

Under the condition of our theorem (and for the given BPA_δ system Δ) we have constructed a BPA system Δ' such that $\Delta \overset{n}{\sim} \Delta'$.

$\square$

Theorem 11. *Let $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ be a BPA_δ system. Suppose that there are infinitely many pairwise nonstrictly nonbisimilar $Y\alpha \in \mathcal{V}ar_\delta.\mathcal{V}ar^*$ such that $X_1 \longrightarrow^* Y\alpha$. Then there is no BPA system Δ' such that $\Delta \overset{n}{\sim} \Delta'$.*

Proof: The proof of this theorem is based on an immediate lemma.

Lemma 15. *Suppose that α and β are states of some BPA_δ system. Then*

$$\alpha \overset{n}{\sim} \beta \;\Rightarrow\; \|\alpha\| = \|\beta\| \, .$$

Let us assume that there exists Δ' (w.l.o.g. we may suppose that Δ' is in 3–GNF) such that $\Delta \overset{n}{\sim} \Delta'$. We show that this is not possible. Since there are infinitely many reachable states $Y_1\alpha_1, Y_2\alpha_2, \ldots$ of Δ which are pairwise nonstrictly nonbisimilar there must be corresponding states $\beta_1, \beta_2, \ldots$ of the system Δ' such that $Y_i\alpha_i \overset{n}{\sim} \beta_i$ for $i = 1, 2, \ldots$. Let us now define a constant N_{max} as $N_{max} \overset{\text{def}}{=} \max\{\| Y_i \|_\delta \mid i = 1, 2, \ldots\}$ where $\| Y \|_\delta \overset{\text{def}}{=} \min\{length(w) \mid Y \xrightarrow{w} \delta \text{ or } \exists E \in \mathcal{E}^+_{\text{BPA}} : Y \xrightarrow{w} \delta.E\}$. 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 (Lemma 15). 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 $\beta_k \overset{n}{\sim} \beta_l$. Then also $Y_k\alpha_k \overset{n}{\sim} Y_l\alpha_l$, which is contradiction. □

Theorems above give us more intuitive image of what is the power of deadlocks. Suppose now that we have a BPA$_\delta$ 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 theorem.

Theorem 12. *Let $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ be a BPA$_\delta$ system. There are only finitely many pairwise nonstrictly nonbisimilar $Y\alpha \in \mathcal{V}ar_\delta.\mathcal{V}ar^*$ such that $X_1 \longrightarrow^* Y\alpha$ if and only if there exists a BPA system $(\mathcal{V}ar', \mathcal{A}ct', \Delta', X_1')$ such that $\Delta \overset{n}{\sim} \Delta'$.*

Proof: The implication from left to right follows from the Theorem 10 and from the fact that a BPA$_\delta$ system can be bisimilarly described in 3–GNF, which has been proved in the Theorem 1. The other implication is an immediate consequence of the Theorem 11. □

7 Conclusion

In this paper we have focused on the class of BPA processes extended with deadlocks. It has been shown that for input-output semantics the extention is no acquisition. On the other hand the BPA$_\delta$ 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 the state ϵ and δ, we speak about nonstrict bisimilarity and if we do, we call the appropriate bisimulation equivalence as strict. We have shown that some decidable properties of BPA systems remain decidable in the BPA$_\delta$ class, e.g. decidability of bisimulation equivalence and regularity extends to BPA$_\delta$ systems.

Finally we have solved the question whether, given a BPA$_\delta$ system Δ, there is an equivalent description (with regard to bisimilarity) of Δ in terms of BPA syntax. The solution for strict bisimilarity is rather technical.

However, the answer to the problem dealing with nonstrict bisimilarity exploited a nice semantic characterisation 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 characterisation is syntactically checkable. Future research could answer to this problem and there still remain many issues to examine such as extending the classes BPP or PA with deadlocks.

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

[BBK87] 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-Verlag, 1987.

[BBK93] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *Journal of the Association for Computing Machinery*, 40:653–682, 1993.

[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, 1995.

[BCS96] O. Burkart, D. Caucal, and B. Steffen. Bisimulation collapse and the process taxonomy. In *Proceedings of CONCUR'96* [Con96], pages 247–262.

[BHPS61] Y. Bar-Hillel, M. Perles, and E. Shamir. On formal properties of simple phrase structure grammars. *Zeitschrift* *für Phonetik, Sprachwissenschaft und Kommunikationsforschung*, 14:143–177, 1961.

[BK85] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37:77–121, 1985.

[BK88] J.A. Bergstra and J.W. Klop. Process theory based on bisimulation semantics. In *Advanced Topics in Artificial Intelligence*, volume 345 of *LNCS*, pages 50–122. Springer-Verlag, 1988.

[Bos97] D. Bosscher. *Grammars Modulo Bisimulation*. PhD thesis, CWI, University of Amsterdam, 1997.

[Cau88] D. Caucal. Graphes canoniques de graphes algebriques. Rapport de Recherche 872, INRIA, 1988.

[CHS92] 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-Verlag, 1992.

[CHS95] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. *Information and Computation*, 121:143–148, 1995.

[Con96] *Proceedings of CONCUR'96*, volume 1119 of *LNCS*. Springer-Verlag, 1996.

[GH94] J.F. Groote and H. Hüttel. Undecidable equivalences for basic process algebra. *Information and Computation*, 115(2):353–371, 1994.

[Gro92] J.F. Groote. A short proof of the decidability of bisimulation for normed BPA processes. *IPL*, 42:167–171, 1992.

[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. Faron Moller and Graham Birtwistle, 1996.

[HS91] 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 Computer Society Press, 1991.

[HT95] D.T. Huynh and L. Tian. On deciding readiness and failure equivalences for processes in Σ_2^P. *Information and Computation*, 117(2):193–205, 1995.

[Hüt91] H. Hüttel. *Decidability, Behavioural Equivalences and Infinite Transition Graphs*. PhD thesis, The University of Edinburgh, 1991.

[Jan95] P. Jančar. Undecidability of bisimilarity for Petri nets and some related problems. *Theoretical Computer Science*, 148(2):281–301, 1995.

[Jan97] P. Jančar. Bisimulation equivalence is decidable for one-counter processes. In *Proceedings of ICALP'97*, volume 1256 of *LNCS*, pages 549–559. Springer-Verlag, 1997.

[JE96] 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-Verlag, 1996.

[Kuč96] A. Kučera. Regularity is decidable for normed PA processes in polynomial time. In *Proceedings of FST&TCS'96*, volume 1180 of *LNCS*, pages 111–122. Springer-Verlag, 1996.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[Mol96] F. Moller. Infinite results. In *Proceedings of CONCUR'96* [Con96], pages 195–216.

[Par81] D.M.R. Park. Concurrency and automata on infinite sequences. In *Proceedings* 5^{th} *GI Conference*, volume 104 of *LNCS*, pages 167–183. Springer-Verlag, 1981.

[vG90a] R.J. van Glabbeek. *Comparative Concurrency Semantics and Refinement of Actions*. PhD thesis, CWI/Vrije Universiteit, 1990.

[vG90b] R.J. van Glabbeek. The linear time—branching time spectrum. In *Proceedings of CONCUR'90*, volume 458 of *LNCS*, pages 278–297. Springer-Verlag, 1990.

Publications in the FI MU Report Series are in general accessible
via WWW and anonymous FTP:

`http://www.fi.muni.cz/informatics/reports/`
`ftp  ftp.fi.muni.cz (cd pub/reports)`

Copies may be also obtained by contacting:

Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic