Basic Process Algebra with Deadlocking States

Jiří Srba*

Faculty of Informatics
Masaryk University
Botanická 68a
CZ-60200 Brno, Czech Republic
srba@fi.muni.cz

BRICS**
Department of Computer Science
University of Aarhus
Ny Munkegade bld. 540
DK-8000 Aarhus C, Denmark
srba@brics.dk

Abstract. Bisimilarity and regularity are decidable properties for the class of BPA (or context–free) processes [CHS95, BCS96]. We extend BPA with a deadlocking state obtaining BPA_δ systems. We show that the BPA_δ class is more expressive w.r.t. bisimilarity, but it remains language equivalent to BPA. We prove that bisimilarity and regularity remain decidable for BPA_δ. Finally we give a characterisation of those BPA_δ processes that can be equivalently (up to bisimilarity) described within the "pure" BPA syntax.

Keywords: process algebra, BPA, bisimulation, regularity, deadlocks

1 Introduction

This paper deals with BPA (Basic Process Algebra) processes 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 context-free grammars in Greibach normal form (GNF), in which only left-most derivations are allowed. For a detailed description of the relation between language and process theory we refer to [HM96]. We define the class BPA_δ as 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 means that if a system reaches a state where the first variable is δ, the

* The author is supported by the Grant Agency of Czech Republic, grant No. 201/97/0456 and by grant FRVŠ 409/1999.

** Basic Research in Computer Science, Centre of the Danish National Research Foundation.

system is stuck at this state and no more actions can be performed. There are two approaches to giving a semantics to the δ construct. First, δ is identified 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 the empty process ϵ.

We show in Section 3 that extending BPA systems with deadlock does not allow us to define a larger class of languages. On the other hand the class of BPA$_\delta$ systems is larger with regard to bisimilarity – the behaviour equivalence. It is known from [CHS95] and [BCS96] that bisimilarity and regularity is decidable in BPA systems. In Sections 4 and 5 we extend these results to both strict and nonstrict versions of bisimilarity and regularity. The trick used for this extension is based on the idea that δ can be simulated by a special unnormed variable. Moreover we show that strict and nonstrict regularity coincide, which is not the case for strict vs. nonstrict bisimilarity. Bosscher has independently proved in his PhD thesis [Bos97] that decidability of bisimilarity and regularity extends to a kind of BPA system with deadlocks, using a related calculus with explicit termination relation. However, he considers only the strict versions and it is not obvious whether his results imply ours.

The last question explored in this paper (Section 6) is concerned with deciding whether there exists a bisimilar description of a BPA$_\delta$ system in a BPA syntax. We prove that this is decidable for both strict and nonstrict bisimilarity. Moreover we show that the corresponding BPA syntax can be effectively constructed.

We also provide a semantic characterisation of the situation in the nonstrict case yielding an effective algorithm for the transformation.

2 Basic definitions

When dealing with processes we need some structure to describe their operational semantics. Transition systems are widely used as a suitable structure for this purpose and in the rest of this paper we will understand processes as nodes of certain transition systems. We introduce labelled transition systems 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 5-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* $\alpha \xrightarrow{a} \beta$*, 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* $a \in \mathcal{A}ct$ *and* $\beta \in S$ *such that* $\alpha \xrightarrow{a} \beta$*.*

The transition relation $\longrightarrow$ can alternatively be understood as a set of binary relations $\{\xrightarrow{a}\}_{a \in \mathcal{A}ct}$. As usual we extend the transition relation to the elements of $\mathcal{A}ct^*$ ($\alpha \xrightarrow{\epsilon} \alpha$ and inductively $\alpha \xrightarrow{aw} \beta$ iff $\exists \gamma : \alpha \xrightarrow{a} \gamma$ and $\gamma \xrightarrow{w} \beta$ 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 $\alpha \xrightarrow{w} \beta$ if $w \in \mathcal{A}ct^*$ is irrelevant. A state β is *reachable* from a 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 $\alpha \xrightarrow{a} \beta$.

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*

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

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 concurrency theory, language equivalence is generally taken to be a coarse equivalence. Many stronger equivalences have been introduced e.g. in [vG90b, vG90a], with *bisimulation equivalence* being the strongest one. Bisimulation equivalence was defined by Park [Par81] and studied e.g. by Milner [Mil89]. Its definition is the following (see [Mol96]).

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. If $\alpha \sim \beta$ then $\alpha =_L \beta$ for all $\alpha, \beta \in S$.*

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.

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

Here a ranges over $\mathcal{A}ct$ and X ranges over $\mathcal{V}ar$. We define $\mathcal{E}_{\text{BPA}} \stackrel{\text{def}}{=} \{\epsilon\} \cup \mathcal{E}^+_{\text{BPA}}$. We call BPA expressions processes and we do not list the elements of the sets $\mathcal{V}ar$ and $\mathcal{A}ct$, if they are clear from the context. 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 $X,\ a+X,\ (a+b+X).c,\ \epsilon$ 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* $E_i \in \mathcal{E}^+_{BPA}$ *is a guarded BPA expression with variables drawn from the set* $\mathcal{V}ar$ *and actions from* $\mathcal{A}ct$.

Speaking about variables resp. actions used in a system $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$, we use the notation $\mathcal{V}ar(\Delta)$ resp. $\mathcal{A}ct(\Delta)$ and we often identify $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ 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* $(S, \mathcal{A}ct, \{\stackrel{a}{\longrightarrow}\}_{a \in \mathcal{A}ct}, X_1, \{\epsilon\})$ *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.*

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

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

Fig. 1. SOS rules

We may assume that the operator '.' for sequential composition is associative and the operator '+' for nondeterministic choice is associative, commutative and idempotent.
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 semantics.

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 $\Delta = \{X_i \stackrel{\mathrm{def}}{=} E_i \mid i = 1, \ldots, n\}$ 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$_\delta$ system (we simply include δ into $\mathcal{V}ar$ but we do not use it).
A BPA$_\delta$ 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 1. As there is no defining equation for δ, it holds that $\delta.E \not\longrightarrow$ for any $E \in \mathcal{E}^+_{\mathrm{BPA}}$, i.e. there is no transition from a process expression starting with δ.

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

Remark 2. These two notions of bisimilarity imply that $\delta \stackrel{n}{\sim} \epsilon$ but $\delta \stackrel{s}{\not\sim} \epsilon$.

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

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

Definition 8. *A BPA (resp. BPA$_\delta$) system Δ is said to be in* Greibach Normal Form (GNF) *iff all its defining equations are of the form*

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

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 the existence of an equivalent Δ in 3–GNF for BPA systems, which can be found e.g. in [Hüt91, BBK93, BBK87, HM96].

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

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 $\delta.E \sim \delta$. □

We may assume that we are working only with BPA_δ systems in GNF since it has been proved that any BPA_δ (and also BPA) system can be effectively presented in 3–GNF and this construction preserves bisimilarity. This also justifies the assumption that all reachable states of a given BPA or BPA_δ system are elements of $\mathcal{V}ar^*$.
An important subclass of BPA resp. BPA_δ systems can be obtained by an extra restriction on the involved processes – *normedness.*

Definition 9. *Let $E \in \mathcal{E}_{BPA}$. We define the* norm *of E as:*

$$\|E\| \stackrel{\text{def}}{=} \begin{cases} \min\{\ \mathit{length}(w) \mid \exists G : E \stackrel{w}{\longrightarrow} G \not\longrightarrow\}, \text{ if such } w \text{ exists} \\ \infty, \quad \text{otherwise} \end{cases}$$

We call the expression E normed iff $\|E\| < \infty$. A process Δ is normed iff all its variables are normed.

We remind the reader of 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 labelled Petri nets. Consequently, it is also decidable for BPP processes since BPP is a subclass of Petri nets (see e.g. [Mol96]). Regularity is also decidable in the class of normed PA processes and even in polynomial time – a 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 is postponed to Section 5 where we also show that decidability of regularity extends to BPA_δ systems.

Definition 10. *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.

3 Expressibility of BPA_δ systems

In this section we justify the importance of introducing a deadlocking state into the class of 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 us to generate a richer family of 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 $\Delta = \{X \stackrel{\text{def}}{=} a\delta\}$ 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 non-strictly bisimilar to it.*

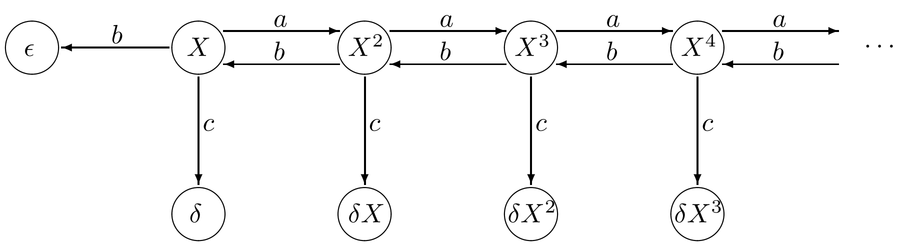

Fig. 2. Labelled transition system for $X \stackrel{\text{def}}{=} aXX + b + c\delta$

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 2) 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 $\Delta \stackrel{n}{\sim} \Delta'$. 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 $X^n \stackrel{n}{\sim} E$. A state X^n has norm 1 for any $n \geq 1$, 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 a contradiction – Δ is a system where infinitely many nonstrictly nonbisimilar states are reachable. $\square$

We show that the classes BPA and BPA$_\delta$ 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 allow us to be more expressive.

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 Definition 2.)*

Definition 12. *We define the family of languages generated by BPA resp. BPA$_\delta$ systems as follows:*

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

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

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 pair 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 reaching a suffix of the form $\delta\alpha$. We use the notation $a\alpha \in Y$ as 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 $\mathcal{V}ar(\Delta) \stackrel{\text{def}}{=} \cup_{X \in \mathcal{V}ar(\Delta_\delta)-\{\delta\}}\{X^\epsilon, X^\delta\} \cup \{X_1^{\epsilon\delta}\}$ 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 forms (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 of Δ. For each $X \in \mathcal{V}ar(\Delta_\delta) - \{\delta\}$, the summands of the variables X^ϵ and X^δ satisfy:

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

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 2. Consider 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 follows: $\Delta = \{X^\epsilon \stackrel{\text{def}}{=} aX^\epsilon X^\epsilon + b + bY^\epsilon,\ X^\delta \stackrel{\text{def}}{=} aX^\epsilon X^\delta + aX^\delta + c + bY^\delta,\ Y^\epsilon \stackrel{\text{def}}{=} b,\ Y^\delta \stackrel{\text{def}}{=} a.Y^\delta,\ X^{\epsilon\delta} \stackrel{\text{def}}{=} aX^\epsilon X^\epsilon + b + bY^\epsilon + aX^\epsilon X^\delta + aX^\delta + c + bY^\delta\}$.

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 the 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 follows:*

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

Lemma 3. *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. The proof proceeds 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 $L_n(Y) \stackrel{\text{def}}{=} \{w \in L(Y) \mid \mathit{length}(w) \leq n\}$. Notice that because of Lemma 3 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 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 a well known result by Christensen, Hüttel and Stirling that 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 later gave in [BCS95] an elementary decision procedure for BPA bisimilarity.
On the other hand, 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 van 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 show that 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 we implicitly assume w.l.o.g. 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 $\mathcal{T} \stackrel{n}{\sim} \overline{\mathcal{T}}$.*

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 $f : \mathcal{E}_{\mathrm{BPA}_\delta} \longrightarrow \mathcal{E}_{\mathrm{BPA}}$ as follows:

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

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 $(X_1 \stackrel{\mathrm{def}}{=} E_1) \in \Delta$ and $(\overline{X_1} \stackrel{\mathrm{def}}{=} \overline{E_1}) \in \overline{\Delta}$, we state

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

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 $G \in \mathcal{E}^+_{\mathrm{BPA}}$). 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 $\mathcal{T} \stackrel{n}{\sim} \overline{\mathcal{T}}$ 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 $\mathcal{T} \stackrel{n}{\sim} \overline{\mathcal{T}}$. □

Example 3. Let $\Delta = \{X \stackrel{\mathrm{def}}{=} aXX + b + c\delta\}$. The system Δ' from the proof above is the following.

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

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 $\mathcal{T} \stackrel{s}{\sim} \overline{\mathcal{T}}$.*

Proof. The proof is quite easy because for strict bisimilarity we have that $\delta \stackrel{s}{\not\sim} \epsilon$ and we can use a 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 $\overline{X_1}$, and we do not add the new equations $X_1' \stackrel{\mathrm{def}}{=} f(E_1).D$ and $\overline{X_1'} \stackrel{\mathrm{def}}{=} f(\overline{E_1}).D$. 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 $G \in \mathcal{E}^+_{\mathrm{BPA}}$) 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 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 class of BPA$_\delta$ systems.
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$_\delta$ we introduced two notions of bisimilarity (strict and nonstrict) and moreover we may consider regularity with regard to finite-state BPA or BPA$_\delta$ system. It does not make any sense to consider strict bisimilarity w.r.t. finite-state BPA . The nonstrict case is solved by the following lemma.

Lemma 4. *Let Δ be a BPA$_\delta$ 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:

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

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

When dealing with regularity we give two definitions, as in the case of strict and nonstrict bisimilarity. The second one is motivated by Lemma 4 above.

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

Definition 15. *A BPA$_\delta$ 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 are decidable in the class BPA$_\delta$ and thus extend the result from [BCS96].

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, a corresponding finite state BPA$_\delta$ system can be effectively constructed.*

Proof. We use again the trick from the proof of 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 occurrences 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 for the class of BPA systems is decidable (see [BCS96]), strict regularity for BPA$_\delta$ is also decidable. Moreover a 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 occurrences of each variable bisimilar to D with δ and remove definitions of such variables. $\square$

Example 4. 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

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

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

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

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 5. *A BPA$_\delta$ system Δ is strictly regular iff Δ is nonstrictly regular.*

Proof. First, we prove the implication from left to right. Suppose that Δ is strictly regular, i.e. there exists a BPA$_\delta$ system Δ' with finitely many reachable states such that $\Delta \stackrel{s}{\sim} \Delta'$. Because of Lemma 2 we know that $\Delta \stackrel{n}{\sim} \Delta'$ and using Lemma 4 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 Lemma 4. Let X_1 and X_1' be leading variables of the systems Δ resp. Δ'. Then we know that there exists some 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 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 a relation S as follows:

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

Then obviously $(X_1, X_1') \in S$ and moreover we show that S is a 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 $G \in \mathcal{E}^+_{\mathrm{BPA}}$ 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 $G \in \mathcal{E}^+_{\mathrm{BPA}}$ 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 $G \in \mathcal{E}^+_{\mathrm{BPA}}$ 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, a corresponding finite state BPA system can be effectively constructed.*

Proof. Using Lemma 5 and 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 Lemma 5 gives directions as to how to construct a corresponding finite state BPA system. □

6 Describing BPA_δ in BPA syntax

In Section 3 we have shown that the class of BPA_δ systems is strictly larger (w.r.t. bisimilarity) than that of BPA. This gives rise to the question of whether a given BPA_δ 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 strict bisimulation is given by Theorem 9, and Theorem 12 demonstrates the corresponding result for nonstrict bisimulation.

6.1 Strict case – decidability

The proof of the following theorem uses a construction, which is essentially the marking algorithm used for checking if a context-free grammar defines the empty language.

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, a system Δ' can be effectively constructed.*

Proof. Our proof is based on the fact that $\delta \overset{s}{\not\sim} \epsilon$. Consider a 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 sets $M_0, M_1, \ldots$ of variables from which deadlock is reachable as the following. The notation $\alpha \in E$ means again that α is a summand in the expression E.

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

For $i \geq 0$ the set M_{i+1} is 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 the reader of the fact 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 any $l > 0$. 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 $(X \stackrel{\mathrm{def}}{=} E) \in \Delta$ we have that $(X \stackrel{\mathrm{def}}{=} E') \in \Delta'$ 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 – semantic characterization

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 Theorem 12.

We can simply observe that in a BPA$_\delta$ labelled transition system there are only finitely many successors of each state. In that case we call the system *image-finite.*

Definition 16. *A labelled transition system* $(S, \mathcal{A}ct, \longrightarrow, \alpha_0, F)$ *is* image-finite *if the set* $\{\beta \mid \alpha \stackrel{a}{\longrightarrow} \beta\}$ *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* $\alpha, \beta \in S$ *whenever* ($\alpha \in F$ *iff* $\beta \in F$)
- $\alpha \sim_{k+1} \beta$ *iff for each* $a \in \mathcal{A}ct$*:*
  - *if* $\alpha \stackrel{a}{\longrightarrow} \alpha'$ *then* $\beta \stackrel{a}{\longrightarrow} \beta'$ *for some* β' *such that* $\alpha' \sim_k \beta'$
  - *if* $\beta \stackrel{a}{\longrightarrow} \beta'$ *then* $\alpha \stackrel{a}{\longrightarrow} \alpha'$ *for some* α' *such that* $\alpha' \sim_k \beta'$
  - $\alpha \in F$ *iff* $\beta \in F$

The following lemma is standard.

Lemma 6. *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 3. In the case of BPA$_\delta$ systems and considering nonstrict bisimilarity, the third condition $\alpha \in F$ iff $\beta \in F$ in Definition 17 is always true since all the terminal states are included in F.

In what follows, the set of variables from which 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*

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

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

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, C over $\mathcal{V}ar_\epsilon$.

Remark 4. We remind the reader of the fact that the sets $\mathcal{V}ar_\delta$ and $\mathcal{V}ar_\epsilon$ can be effectively constructed as we have demonstrated in the proof of 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 $\Delta \overset{n}{\sim} \Delta'$.*

Proof. Let us suppose that $X_1 \in \mathcal{V}ar_\epsilon$. Then the system Δ can be trivially transformed into a 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 the following. We want to achieve

$$f_\alpha(E) \overset{n}{\sim} E\alpha$$

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 $\gamma \in \mathcal{V}ar^*$. For the definition see Figure 3.

Let us now construct a BPA system Δ' where

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

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

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

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

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

A transformation of defining equations of the variables from $\mathcal{V}ar_\delta$ is the goal of 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.

$$f_\alpha(\sum_{i=1}^{n} a_i\alpha_i) = \sum_{i=1}^{n} f_\alpha(a_i\alpha_i) \qquad r_\alpha(\sum_{i=1}^{n} a_i\alpha_i) = \bigcup_{i=1}^{n} r_\alpha(a_i\alpha_i)$$

$$f_\alpha(aXY) = aX^{Y\alpha} \qquad r_\alpha(aXY) = \{X^{Y\alpha}\}$$

$$f_\alpha(aX\delta) = aX^\epsilon \qquad r_\alpha(aX\delta) = \{X^\epsilon\}$$

$$f_\alpha(a\delta) = a \qquad r_\alpha(a\delta) = \emptyset$$

$$f_\alpha(aX) = aX^\alpha \qquad r_\alpha(aX) = \{X^\alpha\}$$

$$f_\alpha(aAB) = \begin{cases} aAB\beta U^\gamma \\ aAB\beta C \\ aAB\alpha \end{cases} \qquad r_\alpha(aAB) = \begin{cases} \{U^\gamma\} & \text{if } \alpha = \beta U\gamma \\ \emptyset & \text{if } \alpha = \beta C\gamma \\ \emptyset & \text{otherwise} \end{cases}$$

$$f_\alpha(a) = \begin{cases} a\beta U^\gamma \\ a\beta C \\ a\alpha \end{cases} \qquad r_\alpha(a) = \begin{cases} \{U^\gamma\} & \text{if } \alpha = \beta U\gamma \\ \emptyset & \text{if } \alpha = \beta C\gamma \\ \emptyset & \text{otherwise} \end{cases}$$

$$f_\alpha(aA\delta) = aA \qquad r_\alpha(aA\delta) = \emptyset$$

$$f_\alpha(aA) = \begin{cases} aA\beta U^\gamma \\ aA\beta C \\ aA\alpha \end{cases} \qquad r_\alpha(aA) = \begin{cases} \{U^\gamma\} & \text{if } \alpha = \beta U\gamma \\ \emptyset & \text{if } \alpha = \beta C\gamma \\ \emptyset & \text{otherwise} \end{cases}$$

$$f_\alpha(aXA) = aX^{A\alpha} \qquad r_\alpha(aXA) = \{X^{A\alpha}\}$$

$$f_\alpha(aAX) = aAX^\alpha \qquad r_\alpha(aAX) = \{X^\alpha\}$$

Fig. 3. Definition of f_α and r_α.

Algorithm 1

1 $\quad \mathsf{Solve} := \{X_1^\epsilon\}$
2 $\quad \mathsf{Added} := \{X_1^\epsilon\}$
3 $\quad \Gamma := \emptyset$
4 $\quad$ **while** $\mathsf{Solve} \neq \emptyset$ **do**
5 $\qquad$ *choose an* $X^\alpha \in \mathsf{Solve}$ *with* $(X \stackrel{\mathrm{def}}{=} E) \in \Delta$
6 $\qquad \Gamma := \Gamma \cup \{X^\alpha \stackrel{\mathrm{def}}{=} f_\alpha(E)\}$
7 $\qquad \mathsf{Add} := \{Y^\beta \in r_\alpha(E) \mid \forall Z^\omega \in \mathsf{Added} : Y\beta \stackrel{n}{\not\sim} Z\omega\}$
8 $\qquad$ **while** $\exists Y^\beta, Z^\omega \in \mathsf{Add} : Y^\beta \neq Z^\omega \ \wedge \ Y\beta \stackrel{n}{\sim} Z\omega$ **do**
9 $\qquad\quad \mathsf{Add} := \mathsf{Add} - \{Y^\beta\}$
10 $\qquad$ **endwhile**
11 $\qquad \mathsf{Solve} := (\mathsf{Solve} - \{X^\alpha\}) \cup \mathsf{Add}$
12 $\qquad \mathsf{Added} := \mathsf{Added} \cup \mathsf{Add}$
13 $\qquad$ **for** $\forall Y^\beta \in r_\alpha(E) - \mathsf{Add}$ **do**
14 $\qquad\quad$ *replace all occurrences of* Y^β *in* Γ *with* Z^ω
15 $\qquad\quad$ *where* $Z^\omega \in \mathsf{Added} : Y\beta \stackrel{n}{\sim} Z\omega$
16 $\qquad$ **endfor**
17 $\quad$ **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 7. *For the loop 4–17 of Algorithm 1 the following invariant $\mathcal{I}$ holds.*

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

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. Finally, line 7 ensures that $\mathcal{I}$ will hold for $\mathsf{Added}{:=}\mathsf{Added} \cup \mathsf{Add}$ also. □

Lemma 8. *Whenever during the execution of Algorithm 1 we have $Y^\alpha \in \mathsf{Added}$ 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 $E \in \mathcal{E}^+_{\mathrm{BPA}}$ such that E is in 3–GNF. □

Lemma 9. *Whenever during the execution of Algorithm 1 we have $Y^\beta \in \mathsf{Added}$ 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 $X^\alpha \in \mathsf{Solve}$ 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 $X_1 \longrightarrow^* X\alpha \longrightarrow^* Y\beta$.

□

Lemma 10. *Under the assumptions of Theorem 10, 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 Lemma 9 and 8 we know that $\forall Y^\beta \in \mathsf{Added} : Y \in \mathcal{V}ar_\delta \quad \wedge \quad X_1 \longrightarrow^* Y\beta$. Moreover from Lemma 7 it 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. □

From the previous lemma we know that 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 11. *After the execution of Algorithm 1 we have $V^\alpha \overset{n}{\sim} V\alpha$ 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 $V^\alpha \overset{n}{\sim} V\alpha$.
Basic step: We get $V^\alpha \sim_0 V\alpha$ 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 $X^\alpha \in \mathsf{Added}$ and so $AX^\alpha \sim_k AX\alpha$.
  - $V^\alpha \xrightarrow{a} AZ^\omega$ where $Z^\omega \in \mathsf{Added}$ 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 $X^{Y\alpha} \in \mathsf{Added}$ and using the induction hypothesis we get $X^{Y\alpha} \sim_k XY\alpha$.
  - $V^\alpha \xrightarrow{a} Z^\omega$, where $Z^\omega \in \mathsf{Added}$ 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 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 $V\alpha \stackrel{a}{\longrightarrow} AB\alpha$ then one of the following cases will hold:
  - $V^{\alpha} \stackrel{a}{\longrightarrow} AB\beta U^{\gamma}$, 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 $AB\beta U^{\gamma} \sim_k AB\alpha$.
  - $V^{\alpha} \stackrel{a}{\longrightarrow} AB\beta Z^{\omega}$, 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 $Z\omega \stackrel{n}{\sim} U\gamma$. This means that $AB\beta Z^{\omega} \sim_k AB\alpha$.
  - $V^{\alpha} \stackrel{a}{\longrightarrow} AB\beta C$ such that $\|C\|= \infty$ and $\alpha = \beta C\gamma$ but then $AB\beta C \sim_k AB\alpha$.
  - $V^{\alpha} \stackrel{a}{\longrightarrow} AB\alpha$ 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 $V\alpha \stackrel{a}{\longrightarrow} AX\alpha$ then one of the following cases will hold:
  - $V^{\alpha} \stackrel{a}{\longrightarrow} AX^{\alpha}$, where $X^{\alpha} \in \mathsf{Added}$ and using the induction hypothesis we get $X^{\alpha} \sim_k X\alpha$ and so $AX^{\alpha} \sim_k AX\alpha$.
  - $V^{\alpha} \stackrel{a}{\longrightarrow} AZ^{\omega}$, where $Z^{\omega} \in \mathsf{Added}$ and X^{α} was at lines 14,15 replaced with Z^{ω}. Then $Z^{\omega} \sim_k Z\omega$ (induction hypothesis) and $Z\omega \stackrel{n}{\sim} X\alpha$. This means that $AZ^{\omega} \sim_k AX\alpha$.

$\square$

Lemma 12. *The system Δ' is a BPA system and moreover $X_1 \stackrel{n}{\sim} X_1^{\epsilon}$.*

Proof. There are no undefined variables in Δ', which follows from the fact that each variable added to 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 $X_1 \stackrel{n}{\sim} X_1^{\epsilon}$ follows from Lemma 11. $\square$

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

Theorem 11. *Let $(\mathcal{V}ar, \mathcal{A}ct, \Delta, X_1)$ be a BPA$_\delta$ 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 $\Delta \stackrel{n}{\sim} \Delta'$.*

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

Lemma 13. *Suppose that α and β are states of some BPA$_\delta$ system. Then*

$$\alpha \stackrel{n}{\sim} \beta \quad \Rightarrow \quad \|\alpha\|=\|\beta\| \ .$$

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 non-strictly 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 $\|Y\|_\delta \overset{\text{def}}{=} \min\{\mathit{length}(w) \mid Y \xrightarrow{w} \delta$ or $\exists E \in \mathcal{E}^+_{\text{BPA}} : Y \xrightarrow{w} \delta.E\}$. Notice that the definition of N_{max} is correct since $\|Y_i\|_\delta < \infty$ for all i (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 to N_{max} for all i (Lemma 13). 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 to 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 a contradiction. □

The theorems above give us a more intuitive image of 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 a corresponding (nonstrictly bisimilar) BPA system does not exist. This condition is 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 Theorem 10 and from the fact that a BPA$_\delta$ system can be bisimilarly described in 3–GNF, which has been proved in Theorem 1. The other implication is an immediate consequence of Theorem 11. □

6.3 Nonstrict case – decidability

In Theorem 12 we have given a sufficient and necessary condition for a BPA$_\delta$ system to be expressible in a BPA syntax. We show that this condition is decidable. We exploit the result by Burkart, Caucal and Steffen (see [BCS96]) where they give an algorithm for describing the factorization of a BPA system w.r.t. bisimulation equivalence in terms of graph grammars. Our condition is decidable by searching for cyclic dependencies in the collapsed graph grammar generating some node with infinite in-degree, from which a deadlock is reachable in a constant distance.
Let $F = \cup_{n\geq 1} F_n$ is a graded set of *labels* such that $\Sigma \subseteq F_2$. A *hyperarc* of arity n is a word $As_1 \ldots s_n$ labelled by $A \in F_n$ joining the vertices $s_1, \ldots, s_n$ in that order. A *hypergraph* is then a set of hyperarcs. The hyperarcs labelled by a label $A \in \Sigma$ we call *terminal arcs.*

Definition 19. *A* (hyper)graph grammar *is a quadruple* (N, Σ, R, G_0) *where*

- $N \subseteq F - \Sigma$ *is a set of graded* nonterminals
- Σ *is the set of* terminals
- R *is a finite set of* rules *of the form* $Ax_1 \dots x_n \triangleright H$ *where* $A \in F_n$*,* H *is a finite hypergraph over* $\Sigma \cup N$ *and* $x_1, \dots, x_n$ *are distinct vertices of* H
- G_0 *is an initial finite hypergraph over* $\Sigma \cup N$

Graph grammars generate (infinite) transition systems over Σ by means of graph rewriting. A *graph rewriting* $G \to G'$ consists of replacing a nonterminal hyperarc $As_1 \dots s_n$ of G by a copy of H where $Ax_1 \dots x_n \triangleright H$ is a rule of the graph grammar such that the vertices s_i and x_i are identified.
The (infinite) graph $\mathcal{G}^\omega(G_0)$ is defined by

$$\mathcal{G}^\omega(G_0) \stackrel{\text{def}}{=} \cup\{[G] \mid G_0 \to^* G\} \quad \text{where} \quad [G] = \{As_1s_2 \in G \mid A \in \Sigma\}.$$

If the graph grammar G is *deterministic* (i.e. there is only a single rule for each nonterminal) than $\mathcal{G}^\omega(G_0)$ is unique up to graph isomorphism. Finally, we call a graph G *regular* if there is a deterministic graph grammar (N, Σ, R, G_0) such that $G = \mathcal{G}^\omega(G_0)$.
Let us consider a labelled transition system where we collapse all the states that are bisimilar. Since bisimulation is a congruence with respect to the operators of BPA, the construction is correct. We will denote the equivalence class represented by a state α as $\sim(\alpha) \stackrel{\text{def}}{=} \{\beta \mid \beta \sim \alpha\}$.

Definition 20. *Let* $\mathcal{T} = (S, \mathcal{A}ct, \longrightarrow, \alpha_0, F)$ *be a labelled transition system. The* factorization of $\mathcal{T}$ w.r.t. bisimulation equivalence *is a labelled transition system* $\mathcal{T}/\!\sim \stackrel{\text{def}}{=} (S/\!\sim, \mathcal{A}ct, \longmapsto, \sim(\alpha_0), F/\!\sim)$ *where*

- $S/\!\sim \stackrel{\text{def}}{=} \{\sim(\alpha) \mid \alpha \in S\}$
- $\longmapsto \stackrel{\text{def}}{=} \{\sim(\alpha) \stackrel{a}{\longmapsto} \sim(\beta) \mid \alpha \stackrel{a}{\longrightarrow} \beta\}$
- $F/\!\sim \stackrel{\text{def}}{=} \{\sim(\alpha) \mid \alpha \in F\}$.

The following theorem shows how the factorization of a BPA system can be described in terms of a graph grammar.

Theorem 13. *[BCS96] The factorization of a BPA transition system w.r.t. bisimulation equivalence is effectively a regular graph.*

We can now state a theorem which gives a characterization of the situation described in Theorem 9 for the nonstrict case.

Theorem 14. *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* $\Delta \stackrel{n}{\sim} \Delta'$*. Moreover if the answer is positive, a system* Δ' *can be effectively constructed.*

Proof. As in the proof of Theorem 6 we transform the system Δ into Δ' such that all occurrences 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 new fresh action. This construction yields a BPA system. We show that the property "there are infinitely many pairwise nonstrictly nonbisimilar $Y\alpha \in \mathcal{V}ar_\delta.\mathcal{V}ar^*$ such that $X_1 \longrightarrow^* Y\alpha$ holds" is decidable (see Theorem 12) and thus we prove our theorem. Let $Y_1\alpha_1, Y_2\alpha_2, \ldots$ denote the nonstrictly nonbisimilar states. Let us now recall the notation from the proof of Theorem 11: $N_{max} \stackrel{\text{def}}{=} \max\{\|Y\|_\delta \mid Y \in \mathcal{V}ar_\delta\}$ where $\|Y\|_\delta \stackrel{\text{def}}{=} \min\{\mathit{length}(w) \mid Y \stackrel{w}{\longrightarrow} \delta \text{ or } \exists E \in \mathcal{E}^+_{\text{BPA}} : Y \stackrel{w}{\longrightarrow} \delta.E\}$. Let us consider the factorization of Δ' w.r.t. bisimulation equivalence, which is effectively a regular graph. Then it is the case that there are infinitely many pairwise nonstrictly nonbisimilar $Y\alpha \in \mathcal{V}ar_\delta.\mathcal{V}ar^*$ such that $X_1 \longrightarrow^* Y\alpha$ if and only if there is a vertex in the regular graph of infinite in-degree such that a vertex with a loop labelled by d is reachable along a path of at most N_{max} edges. This property is decidable from the corresponding graph grammar simply by searching for cyclic dependencies in the graph grammar generating some vertex with infinite in-degree and checking the reachability condition. First, we have to be able to detect if a given vertex v from the graph G_0 or from a right-hand graph of some rule is of infinite in-degree. Assume that all the rules from the grammar are used when building $\mathcal{G}^\omega(G_0)$. Let n denote the number of rules in our grammar. Let v be a vertex corresponding to some hypergraph H with a hyperarc h containing v. To see if this vertex is of infinite in-degree, it is enough to check all derivations starting in H (using the hyperarc h in the first derivation) of length at most n. The vertex v is of infinite in-degree iff there is a derivation of length at most n, which introduces a cycle containing v again in hyperarc h. Moreover, we require that all v's were identified and there is at least one edge pointing to v labelled by some symbol from $\mathcal{A}ct$. Thus we can decide if a given vertex is of infinite in-degree. Checking the reachability condition from v is also easily decidable. We can use a kind of marking algorithm to find all the vertices from which the vertex with a d-loop (this vertex is unique) is reachable in at most N_{max} steps. □

7 Conclusion

In this paper we have focused on the class of BPA processes extended with deadlocks. It has been shown that introducing deadlocks does not allow us to generate a richer family of languages. On the other hand the BPA$_\delta$ class is larger with respect to bisimulation equivalence. We have introduced two notions of bisimilarity to capture a 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 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 of 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$_\delta$ processes bisimilarly describable in BPA syntax: a BPA$_\delta$ system can be transformed into a BPA system (preserving nonstrict bisimilarity) if and only if there are only finitely many nonbisimilar reachable states starting with some variable from which δ is reachable. Moreover, we show that this semantic characterisation is syntactically checkable by using graph grammars and bisimulation collapse [BCS96].

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 grammmars. *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.

[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.