Normed BPA vs. Normed BPP Revisited[⋆]

Petr Jančar, Martin Kot, and Zdeněk Sawa

Center for Applied Cybernetics,
Dept. of Computer Science, Technical University of Ostrava,
17. listopadu 15, 70833 Ostrava-Poruba, Czech Republic.
petr.jancar@vsb.cz, martin.kot@vsb.cz, zdenek.sawa@vsb.cz

Abstract. We present a polynomial-time algorithm deciding bisimilarity between a normed BPA process and a normed BPP process. This improves the previously known exponential upper bound by Černá, Křetínský, Kučera (1999). The algorithm relies on a polynomial bound for the "finite-state core" of the transition system generated by the BPP process. The bound is derived from the "prime form" of the underlying BPP system (where bisimilarity coincides with equality); we suggest an original algorithm for the respective transformation.

Key words: verification, equivalence checking, bisimulation equivalence, Basic Process Algebra, Basic Parallel Processes

1 Introduction

Decidability and complexity of bisimilarity on various classes of processes is a classical topic in process algebra and concurrency theory; see, e.g., [1, 2] for surveys.

One long-standing open problem is the decidability question for the class PA (process algebra), which comprises "context-free" rewrite systems using both sequential and parallel composition. For the subcase of *normed* PA, a procedure working in doubly-exponential nondeterministic time was shown by Hirshfeld and Jerrum [3].

More is known about the "sequential" subclass called BPA (Basic Process Algebra) and the "parallel" subclass called BPP (Basic Parallel Processes). In the case of BPA, the best known algorithm for deciding bisimilarity seems to have doubly-exponential upper bound [4, 1]; the problem is known to be PSPACE-hard [5]. In the case of BPP, the problem is PSPACE-complete [6, 7]. A polynomial-time algorithm for *normed* BPA was shown in [8] (with an upper bound $O(n^{13})$); more recently, an algorithm with running time in $O(n^8 \mathit{polylog}\ n)$ was shown in [9]. For normed BPP, a polynomial time algorithm was presented

⋆ The authors acknowledge the support by the Czech Ministry of Education, Grant No. 1M0567.

in [10] (without a precise complexity analysis), based on so called *prime decompositions*; the upper bound $O(n^3)$ was shown in [11] by another algorithm, based on so called *dd-functions*.

The most difficult matter in the above mentioned algorithm for normed PA [3] is the case when (a process expressed as) sequential composition is bisimilar with (a process expressed as) parallel composition. A basic (sub)problem of this problem is to analyze when a BPA process is bisimilar with a BPP process. Černá, Křetínský, Kučera [12] have shown that this (sub)problem is decidable in the *normed* case; their suggested algorithm is exponential. Decidability in the general (unnormed) case was shown in [13] (without giving any complexity bound).

In this paper, we revisit the normed case, and we present a *polynomial time* algorithm deciding whether a given normed BPA process α is bisimilar with a given normed BPP process M. The main idea is to derive a polynomial bound for the "finite-state core" of the transition system generated by the BPP process M. To this aim we provide a new algorithm, based on *dd*-functions, with time complexity $O(n^3)$, which transforms a given normed BPP process into a "prime form", where bisimilarity coincides with equality. Such a transformation could be based on the prime decompositions in [10] but with worse complexity (which was, in fact, not analyzed in [10]). If the constructed finite-state core exceeds the derived bound, we answer negatively; otherwise we construct a BPA process α' which is bisimilar with M, and the final step is to decide if BPA processes α and α' are bisimilar. This final step can be handled by referring to [8] or [9]. We also sketch a simple self-contained algorithm which uses the fact that α' is close to a finite-state system. This should lead to a better complexity estimation, though we provide no analysis here.

As a side result, our approach also shows a clear polynomial time algorithm testing if there exists a bisimilar BPA process to a given BPP process. This is an alternative to the respective polynomiality result in [12].

Remark. We hope that the new insight will also help to clarify the general (unnormed) case BPA vs. BPP. E.g., the problem mentioned in the previous paragraph seems to be PSPACE-complete in this case.

This paper has the following structure. After basic definitions in Section 2, we describe a transformation of a normed BPP system into the prime form in Section 3. Section 4 contains the crucial result showing the polynomial bound on the "finite-state core". Section 5 finishes the main polynomiality proof.

2 Definitions

We use $\mathbb{N} = \{0, 1, 2, \ldots\}$ to denote the set of nonnegative integers, and we put $\mathbb{N}_{-1} = \mathbb{N} \cup \{-1\}$.

For a set X, $|X|$ denotes the size of X, X^+ denotes the set of nonempty sequences of elements of X, and $X^* = X^+ \cup \{\varepsilon\}$ where ε is the empty sequence. The length of a sequence $x \in X^*$ is denoted by $|x|$ ($|\varepsilon| = 0$). We use x^k (where $x \in X^*$, $k \in \mathbb{N}$) to denote the sequence $xx \cdots x$ where x is repeated k times (in particular $x^0 = \varepsilon$).

A *labelled transition system* (LTS) is a triple $(S, \mathcal{A}, \longrightarrow)$, where S is a set of *states*, $\mathcal{A}$ is a finite set of *actions*, and $\longrightarrow \subseteq S \times \mathcal{A} \times S$ is a *transition relation*. We write $s \xrightarrow{a} s'$ instead of $(s, a, s') \in \longrightarrow$ and we extend this notation to elements of $\mathcal{A}^*$ in the natural way. We write $s \longrightarrow s'$ if there is $a \in \mathcal{A}$ such that $s \xrightarrow{a} s'$ and $s \longrightarrow^* s'$ if $s \xrightarrow{w} s'$ for some $w \in \mathcal{A}^*$. We write $s \xrightarrow{w}$ if there is some s' such that $s \xrightarrow{w} s'$.

Let $(S, \mathcal{A}, \longrightarrow)$ be an LTS. A binary relation $\mathcal{R} \subseteq S \times S$ is a *bisimulation* iff for each $(s, t) \in \mathcal{R}$ and $a \in \mathcal{A}$ we have:

– $\forall s' \in S : s \xrightarrow{a} s' \Rightarrow (\exists t' : t \xrightarrow{a} t' \wedge (s', t') \in \mathcal{R})$, and
– $\forall t' \in S : t \xrightarrow{a} t' \Rightarrow (\exists s' : s \xrightarrow{a} s' \wedge (s', t') \in \mathcal{R})$.

Less formally, each transition $s \xrightarrow{a} s'$ can be *matched* by a transition $t \xrightarrow{a} t'$ where $(s', t') \in \mathcal{R}$ and vice versa.

States s and t are *bisimulation equivalent (bisimilar)*, written $s \sim t$, iff they are related by some bisimulation. We can also relate states of two different LTSs by taking their disjoint union.

A *BPA (system)* is given by a context-free grammar in Greibach normal form. Formally it is a triple $\Sigma = (V_\Sigma, \mathcal{A}_\Sigma, \Gamma_\Sigma)$, where V_Σ is a finite set of *variables* (nonterminals), $\mathcal{A}_\Sigma$ is a finite set of *actions* (terminals) and $\Gamma_\Sigma \subseteq V_\Sigma \times \mathcal{A}_\Sigma \times V_\Sigma^*$ is a finite set of *rewrite rules*. We will use $V, \mathcal{A}, \Gamma$ for the sets of variables, actions and rules if the underlying BPA is clear from context. Again, we write $X \xrightarrow{a} \alpha$ instead of $(X, a, \alpha) \in \Gamma$. A *BPA process* is a pair (α, Σ) where Σ is a BPA system and $\alpha \in V^*$; we often write just α when Σ is clear from context. A BPA Σ gives rise to the LTS $\mathcal{S}_\Sigma = (V^*, \mathcal{A}, \longrightarrow)$ where $\longrightarrow$ is induced from the rewrite rules by the following (deduction) rule: if $X \xrightarrow{a} \alpha$ then $X\beta \xrightarrow{a} \alpha\beta$ for every $\beta \in V^*$.

A *BPP (system)* is defined in a similar way, as a triple $\Delta = (V_\Delta, \mathcal{A}_\Delta, \Gamma_\Delta)$. The only difference is the deduction rule for the associated LTS $\mathcal{S}_\Delta$: if $X \xrightarrow{a} \alpha$ then $\gamma X \delta \xrightarrow{a} \gamma\alpha\delta$ for any $\gamma, \delta \in V^*$ (thus *any* occurrence of a variable can be rewritten, not just the first one). It is easy to observe that BPP processes α, β with the same Parikh image (i.e., containing the same number of occurrences of each variable) are bisimilar. Hence BPP processes can be read modulo commutativity of concatenation and interpreted as multisets of variables; in the rest of the paper we interpret BPP processes in this way whenever convenient. This also suggests to identify a BPP system Δ with a *BPP net*, a labelled Petri net in which each place corresponds to a variable and each transition corresponds to a rewrite rule (and thus has a unique input place); we will freely do this in our later considerations.

Formally, a *BPP net* is a tuple $\Delta = (P_\Delta, Tr_\Delta, \textsc{pre}_\Delta, F_\Delta, \mathcal{A}_\Delta, l_\Delta)$ where P_Δ is a finite set of *places* (variables), Tr_Δ is a finite set of *transitions*, $\textsc{pre}_\Delta : Tr_\Delta \to P_\Delta$ is a function assigning an input place to each transition, $F_\Delta : (P_\Delta \times Tr_\Delta) \to \mathbb{N}$ is a *flow function*, $\mathcal{A}_\Delta$ is a finite set of *actions*, and $l_\Delta : Tr_\Delta \to \mathcal{A}_\Delta$ is a *labelling function*. We will use P, $Tr, \textsc{pre}, F, \mathcal{A}, l$ if the underlying BPP net is clear from context. A rewrite rule $p \xrightarrow{a} \alpha$ of Δ is represented by a transition $t \in Tr$ such that $\textsc{pre}(t) = p$ and $F(t, p')$ is the number of occurrences of p' in α, for each $p' \in P$.

A BPP process is thus, in fact, a *marking*, i.e. a function $M : P \to \mathbb{N}$ which associates a finite number of *tokens* to each place. Note that p^k represents marking M where all k tokens are in one place p ($M(p) = k$ and $M(p') = 0$ for each $p' \neq p$), p represents marking p^1, and ε represents the *zero marking* ($M(p) = 0$ for all $p \in P$).

A transition t is *enabled* at marking M if $M(\textsc{pre}(t)) \geq 1$. An enabled transition t may fire from M, producing a marking M' defined by

$$M'(p) = \begin{cases} M(p) - 1 + F(t,p) & \text{if } p = \textsc{pre}(t) \\ M(p) + F(t,p) & \text{otherwise} \end{cases} .$$

This is denoted by $M \xrightarrow{t} M'$; the notation is extended to $M \xrightarrow{\sigma} M'$ for sequences $\sigma \in T^*$. We write $M \xrightarrow{\sigma}$ if $M \xrightarrow{\sigma} M'$ for some M'.

In the above sence, a BPP Δ gives rise to the LTS $\mathcal{S}_\Delta = (\mathcal{M}_\Delta, \mathcal{A}, \longrightarrow)$ where $\mathcal{M}_\Delta$ is the set of all markings (of the respective BPP net), and $M \xrightarrow{a} M'$ iff there is some $t \in Tr$ such that $l(t) = a$ and $M \xrightarrow{t} M'$.

In the rest of the paper we use symbols $\alpha, \beta, \ldots$ for both BPA processes and BPP processes, and $M_1, M_2, \ldots$ only for the latter.

We say that a BPA system Σ (a BPP net Δ) is *normed* iff $\alpha \longrightarrow^* \varepsilon$ for each state α of $\mathcal{S}_\Sigma$ ($\mathcal{S}_\Delta$). We will use nBPA (nBPP) for normed BPA (normed BPP).

Our central problem, denoted nBPA-nBPP-BISIM, is defined as follows:

Instance: A normed BPA-process (α_0, Σ), a normed BPP-process (M_0, Δ).
Question: Is $\alpha_0 \sim M_0$ (in the disjoint union of $\mathcal{S}_\Sigma$ and $\mathcal{S}_\Delta$) ?

As the size n of an instance of nBPA-nBPP-BISIM we understand the number of bits needed for its (natural) presentation; in particular we consider the numbers $F(t,p)$ in Δ and the numbers in M_0 to be written in binary.

In the rest of this section we assume a fixed nBPA Σ and a fixed nBPP Δ. By a *state* we generally mean a state in the disjoint union of $\mathcal{S}_\Sigma$ and $\mathcal{S}_\Delta$.

Let α be a state (of $\mathcal{S}_\Sigma$ or $\mathcal{S}_\Delta$). *Norm* of α, denoted $\|\alpha\|$, is the length of the shortest $w \in \mathcal{A}^*$ such that $\alpha \xrightarrow{w} \varepsilon$. Note that this also defines norm $\|X\|$ for each variable (place) X. We now note some obvious properties of norms.

– If $\alpha \neq \varepsilon$ then $\|\alpha\| > 0$ for any state α.

- In each nBPA (or nBPP), there is at least one variable (place) with norm 1.
- If $X \xrightarrow{a} \alpha$ is used for a transition $\beta \xrightarrow{a} \beta'$ then $\|\beta'\| - \|\beta\| = \|\alpha\| - \|X\|$.
- $\|\alpha\beta\| = \|\alpha\| + \|\beta\|$ (for BPP-net representation it induces $\|M_1 + M_2\| = \|M_1\| + \|M_2\|$ where marking $M = M_1 + M_2$ is defined componentwise).
- If $\alpha \sim \beta$ then $\|\alpha\| = \|\beta\|$.
- Let $\alpha_1 \sim \alpha_2$, $w \in \mathcal{A}^*$ and $\alpha_1 \xrightarrow{w} \alpha_1'$. There must be a *matching sequence* $\alpha_2 \xrightarrow{w} \alpha_2'$ such that $\alpha_1' \sim \alpha_2'$ (and thus also $\|\alpha_1'\| = \|\alpha_2'\|$).

Finally we note that all norms $\|X\|$, $\|p\|$ for $X \in V_\Sigma$, $p \in P_\Delta$ can be easily computed in polynomial time ($O(n^3)$) and written in polynomial space ($O(n^2)$).

For two states α_1, α_2 we write $\alpha_1 \longrightarrow_R \alpha_2$ if $\alpha_1 \longrightarrow \alpha_2$ and $\|\alpha_2\| = \|\alpha_1\| - 1$. Such a step is called a *norm-reducing step* and the respective rule (transition) is also called *norm reducing*. We write $\alpha_1 \longrightarrow_R^* \alpha_2$ if there is a sequence (called *norm reducing sequence*) of norm reducing steps leading from α_1 to α_2. For each variable (place) X there is at least one norm-reducing rule (transition) $X \longrightarrow_R \alpha$.

We finish by a few notions concerning the BPP net Δ.

For a marking M and a set $Q \subseteq P$ we define $\|M\|_Q$ as the length of the shortest $w \in \mathcal{A}^*$ such that $M \xrightarrow{w} M'$ where $M'(p) = 0$ for all $p \in Q$.

A place $p \in P$ is *unbounded* in (M_0, Δ) iff for each $c \in \mathbb{N}$ there is a marking M' such that $M_0 \longrightarrow^* M'$ and $M'(p) > c$.

We define $\mathit{Tok}(M) = \sum_{p \in P} M(p)$ and $\mathit{Car}(M) = \{p \in P \mid M(p) \geq 1\}$.

A place p is called a *single final place*, an SF-place, if all transitions that take a token from p are of the form $p \xrightarrow{a} p^k$ (i.e., they can only put tokens back to p). It is easy to see that $\|p\| = 1$ for every SF-place p (since Δ is normed). We say that p is a non-SF-place if it is not an SF-place.

3 Normed BPP Systems in the Prime Form

We say that a *BPP system Δ* is *in the prime form* iff bisimilarity coincides with identity on the generated LTS, i.e., $M \sim M'$ iff $M = M'$.

One way to transform a normed BPP system Δ into an equivalent Δ' in the prime form can be based on the algorithm in [10] which computes certain prime decompositions of BPP-variables (i.e., BPP-net places); it is a polynomial algorithm whose precise complexity has not been analyzed. We use another transformation, which is based on the *dd*-functions and is achieved by an algorithm with time complexity in $O(n^3)$.

In [11], the algorithm from [6] was applied to normed processes. Given a normed BPP system $\Delta = (P, \mathit{Tr}, \textsc{pre}, F, \mathcal{A}, l)$, the algorithm finishes in time $O(n^3)$ and constructs a partition $\{T_1, T_2, \ldots, T_m\}$ of the set of transitions such that

$$M \sim M' \text{ iff } d_i(M) = d_i(M') \text{ for all } i = 1, 2, \ldots, m$$

where $d_i(M)$ is the distance to disabling T_i (i.e., the length of the shortest w such that $M \xrightarrow{w} M'$ and in M' all $t \in T_i$ are disabled). Moreover, each class T_i is characterized by the pair (a_i, δ_i) where a_i is the label of all $t \in T_i$ and $\delta_i = (\delta_{i1}, \delta_{i2}, \ldots, \delta_{im})$ is a vector in $(\mathbb{N}_{-1})^m$ such that the following holds for any M, M':

$$\text{if } M \xrightarrow{t} M' \text{ for } t \in T_i \text{ then } d(M') = d(M) + \delta_i$$

where $d(M)$ denotes the vector $(d_1(M), d_2(M), \ldots, d_m(M))$. The type (a_i, δ_i) determines T_i since $(a_i, \delta_i) \neq (a_j, \delta_j)$ for $i \neq j$. For convenience, we will say *transition (of the type)* t_i when meaning any transition $t \in T_i$.

Remark. Space $O(n)$ is sufficient for writing each element of a vector δ_i. There are $O(n)$ such elements in δ_i and $O(n)$ vectors. It follows that space $O(n^3)$ is sufficient for writing all pairs (a_i, δ_i) in binary.

Due to normedness, for every class T_i there is at least one transition t_j which decreases d_i (whenever enabled in M, which also entails $d_i(M) > 0$); this is concisely captured by the next proposition.

Proposition 1. $\forall i \exists j : \delta_{ji} = -1$.

We say that t_i is a *key transition* if it decreases some component of d, i.e. some d_j. Formally we define

$$\text{KEY} = \{i \mid \delta_{ij} = -1 \text{ for some } j\}\,.$$

Proposition 2. $\forall i \in \text{KEY} : \delta_{ii} = -1$.

Proof. If t_i (an element of T_i) decreases some d_j then for each M there is the greatest ℓ such that $M \xrightarrow{(t_i)^\ell}$. The last firing of t_i necessarily decreases d_i. Hence $\delta_{ii} = -1$. □

Thus for each $i \in \text{KEY}$, $d_i(M)$ is the greatest ℓ such that $M \xrightarrow{(t_i)^\ell}$. (A shortest way to disable t_i is to fire it as long as possible.)

We say that t_i *reduces* t_j iff $\delta_{ij} = -1$. Formally we define the following relation RED on KEY:

$$\text{for } i, j \in \text{KEY we put } i \text{ RED } j \text{ iff } \delta_{ij} = -1\,.$$

Proposition 3. RED *is an equivalence relation.*

Proof. Reflexivity follows from Proposition 2.

To show symmetricity, assume $i, j \in \text{KEY}$ (so $\delta_{ii} = \delta_{jj} = -1$) such that $\delta_{ij} = -1$ but $\delta_{ji} \geq 0$. Then firing t_j from M with $d_i(M) > 0$ as long as possible results in M' with $d_j(M') = 0$ and $d_i(M') > 0$. Thus $M' \xrightarrow{t_i}$, which is a contradiction since d_j can not be decreased.

Transitivity follows similarly: Suppose i RED j and j RED k but $\neg(i$ RED $k)$. So all $\delta_{ii}, \delta_{jj}, \delta_{kk}, \delta_{ij}, \delta_{ji}, \delta_{jk}, \delta_{kj}$ are -1 but $\delta_{ik} \geq 0$. Starting from M with $d_k(M) > 0$, we fire t_i as long as possible and thus get M' with $d_i(M') = d_j(M') = 0$ and $d_k(M') > 0$. Thus $M' \xrightarrow{t_k}$, which is a contradiction since d_j can not be decreased. □

Theorem 4. *There is an algorithm, with time complexity in $O(n^3)$, which transforms a given normed BPP system Δ into Δ' in the prime form, and any given state (marking) M of Δ into M' of Δ' such that $M \sim M'$.*

Proof. In the first phase we compute the partition $\{T_1, T_2, \ldots, T_m\}$ as discussed above. We put $Q_i = \text{PRE}(T_i)$ (where $\text{PRE}(T_i) = \{\text{PRE}(t) \mid t \in T_i\}$) and note that $d_i(M) = \|M\|_{Q_i}$. We now easily verify that $Q_i = Q_j$ for $i, j \in \text{KEY}$ iff i RED j (and so j RED i).

The crucial idea is that Δ' will have a place p_C for each class C of the equivalence RED. For any M of Δ, the number $M'(p_C)$ will be equal to $\|M\|_{Q_i}$ for each $i \in C$.

For every $i \in \text{KEY}$ we add a transition t'_i in Δ' such that $\text{PRE}(t'_i) = p_C$ where $i \in C$; t'_i is labelled with a_i and it realizes the (nonnegative) change on the other places $p_{C'}$ according to δ_i (restricted to KEY).

A non-key transition t_i (with $\delta_i \geq (0, 0, \ldots, 0)$) is enabled precisely when a (key) transition decreasing d_i is enabled (recall Proposition 1). Thus for each p_C where C contains j with $\delta_{ji} = -1$ we add a transition t with label a_i and $\text{PRE}(t) = p_C$ which (gives a token back to p_C and) realizes the change δ_i (restricted to KEY). □

In the following text we only consider BPP systems in the prime form.

4 A Bound on the Number of the "Not-all-in-one-SF" Markings

In this subsection we prove the following theorem.

Theorem 5. *Assume a normed BPA system Σ, with the set V of variables, and a normed BPP system Δ in the prime form, with the set P of places. The number of markings M of Δ such that $\alpha \sim M$ for some $\alpha \in V^+$ and M does not have all tokens in one SF-place is at most $4n^2$, where $n = \max\{|V|, |P|\}$.*

We start with a simple observation and then we bound the total number of tokens in the markings mentioned in the theorem.

Proposition 6. *If $A\alpha \sim M$ where $\alpha \in V^*$ and $|Car(M)| \geq 2$ then $\|A\| \geq 2$.*

Proof. From M with $|Car(M)| \geq 2$ we can obviously perform two different norm-reducing steps resulting in two different, and thus nonbisimilar, markings. On the other hand, any $A\alpha$ with $\|A\| = 1$ has a single outcome (namely α) of any norm-reducing step. □

Proposition 7. *If $|Car(M)| \geq 2$ and $\alpha \sim M$ for $\alpha \in V^+$ then $Tok(M) \leq |V|$.*

Proof. In fact, we prove a stronger proposition. To this aim, we order the variables from V into a sequence $A_1, A_2, \ldots, A_{|V|}$ so that $\|A_i\| \leq \|A_j\|$ for $i \leq j$. We now show the following claim: if $A_i\alpha \sim M$, where $|Car(M)| \geq 2$ (and $\alpha \in V^*$), then $Tok(M) \leq i$.

For the sake of contradiction, suppose a counterexample $A_i\alpha \sim M$, $Tok(M) \geq i{+}1$, for minimal i. Proposition 6 shows that $\|A_i\| \geq 2$, hence also $i \geq 2$ (since necessarily $\|A_1\| = 1$); therefore $Tok(M) \geq i{+}1 \geq 3$. There is thus a norm-reducing step $M \longrightarrow_R M'$ such that $|Car(M')| \geq 2$, $Tok(M') \geq i$. This step is matched by $A_i\alpha \longrightarrow_R A_j\beta\alpha$, $A_j\beta\alpha \sim M'$, where necessarily $\|A_j\| < \|A_i\|$ and thus $j < i$. This is a contradiction with the minimality of our counterexample. □

Since a token from any non-SF-place can be moved to another place (with the total number of tokens non-decreasing), we get the following corollary.

Corollary 8. *If $\alpha \sim M$ then $M(p) \leq |V|$ for every non-SF-place p.*

We now partition the markings in the theorem into four classes:

Class 1. Markings M with all tokens in one (non-SF) place ($|Car(M)| = 1$).
Class 2. Markings M with $|Car(M)| \geq 2$ where at least two different places with norm 1 are reachable; this necessarily means $M \longrightarrow^* M'$ for some M' satisfying $M'(p_1) \geq 1$, $M'(p_2) \geq 1$ for some $p_1 \neq p_2$ and $\|p_1\| = \|p_2\| = 1$.
Class 3. Markings M with $|Car(M)| \geq 2$ and with exactly one reachable ("sink") place p with norm 1, where p is a non-SF-place.
Class 4. Markings M with $|Car(M)| \geq 2$ and with exactly one reachable ("sink") place p with norm 1, where p is an SF-place.

We will show that each class contains at most n^2 markings by which we prove the theorem. (In fact, our bound is a bit generous, allowing to avoid some technicalities.)

Proposition 9. *The number of markings in Class 1 is bounded by $|V| \cdot |P| \leq n^2$.*

Proof. According to Corollary 8 there can be at most $|V|$ tokens in any non-SF-place and there are at most $|P|$ non-SF-places. It follows that Class 1 contains at most $|V| \cdot |P| \leq n^2$ markings. $\square$

Proposition 10. *If $\alpha \sim M$ for M from Class 2 then $\alpha = A$ for some $A \in V$. Thus the number of markings in Class 2 is at most $|V| \leq n$.*

Proof. For the sake of contradiction, suppose $A\alpha \sim M$ where $\alpha \in V^+$ and M is from Class 2. We take a counterexample with the minimal length ℓ of a sequence v such that $M \xrightarrow{v} M'$ where $M'(p_1) \geq 1$, $M'(p_2) \geq 1$ for two different p_1, p_2 with norm 1. We note that $\|A\| \geq 2$ by Proposition 6, and first suppose $\ell > 0$. It is easy to verify that there is a move $M \longrightarrow M''$, matched by $A\alpha \longrightarrow B\beta\alpha$, $B\beta\alpha \sim M''$, where $|Car(M'')| \geq 2$ and the respective length ℓ decreased; this would be a contradiction with the assumed minimality. Thus $\ell = 0$, which means $M(p_1) \geq 1$, $M(p_2) \geq 1$. But then M certainly allows $M \longrightarrow_R^* M_1$, $M \longrightarrow_R^* M_2$ where $\|M_1\| = \|M_2\| = \|\alpha\| \geq 1$ and $M_1 \neq M_2$, and thus $M_1 \not\sim M_2$. On the other hand, $A\alpha$ can offer only α as the result of matching such sequences; hence $A\alpha \not\sim M$. $\square$

Proposition 11. *If $A\alpha \sim M$ for $\alpha \in V^+$ and M from Class 3 or 4 then $M \longrightarrow_R^* p^{\|\alpha\|}$ where p is the sink place. Thus $\alpha \sim p^{\|\alpha\|}$.*

Proof. We prove the claim by induction on the norm $\|A\|$. Suppose $A\alpha \sim M$ as in the statement. Proposition 6 implies $\|A\| \geq 2$. M necessarily has a token in a place $p' \neq p$ with the least norm greater than 1. Performing a norm-reducing transition with this token corresponds to some $M \longrightarrow_R M'$, and this must be matched by $A\alpha \longrightarrow_R B\beta\alpha$, $B\beta\alpha \sim M'$, where $\|B\| < \|A\|$. Either $|Car(M')| = 1$, in which case necessarily $M' = p^{\|B\beta\alpha\|}$, or $|Car(M')| \geq 2$, and then $M' \longrightarrow_R^* p^{\|\beta\alpha\|}$ due to the induction hypothesis. Since obviously $p^{\|B\beta\alpha\|} \longrightarrow_R^* p^{\|\beta\alpha\|} \longrightarrow_R^* p^{\|\alpha\|}$, we are done. $\square$

Proposition 12. *If $A\alpha \sim M$ where $\|\alpha\| \geq 2$ and M is from Class 3 or 4 then the sink place p is an SF-place. Hence M is from Class 4.*

Proof. For the sake of contradiction, suppose $A\alpha \sim M$ with $\|\alpha\| \geq 2$, M from Class 3, the sink place p thus being a non-SF-place, and assume $\|A\|$ minimal possible; $\|A\| \geq 2$ by Proposition 6.

If there was a step $M \longrightarrow_R M'$ with $|Car(M')| \geq 2$, the matching $A\alpha \longrightarrow_R B\beta\alpha$ would lead to a contradiction with minimality of $\|A\|$. Since $|Car(M)| \geq 2$, the only remaining possibility is the following: $Tok(M) = 2$, $M(p) = 1$ and $M(p') = 1$ where $p' \longrightarrow_R p^k$ for $k = \|A\| + \|\alpha\| - 2 \geq 2$.

Since the sink place p is a non-SF-place, it must be in a cycle C with at least two places. Moving a token along C cannot generate new tokens, due to Corollary 8, so p' is not in C. On the other hand, C contains some p'' with $\|p''\| = 2$. Starting in M, we can move the token from p to p'', the norm being greater than $\|M\| = \|A\alpha\|$ along the way. For the resulting M' we obviously have $M' \longrightarrow_R^* M''$ for M'' satisfying $M''(p'') = 1$ and $\|M''\| = \|\alpha\|$. $A\alpha$ can match this only by reaching α but $\alpha \sim p^{\|\alpha\|}$ according to Proposition 11 and thus $\alpha \not\sim M''$. □

We can thus have $A\alpha \sim M$ for M from Class 3 only when $\|\alpha\| \leq 1$, and it is thus easy to derive the following corollary.

Corollary 13. *The number of markings in Class 3 is at most* $|V|^2 \leq n^2$.

Proposition 14. *The number of markings in Class 4 is at most* $|V| \cdot |P| \leq n^2$.

Proof. Let $A\alpha \sim M$ for M from Class 4, p being the respective SF-sink place. Using Proposition 11, we derive $\alpha \sim I^k$ where $k = \|\alpha\|$ and $I \in V$, $I \sim p$ (such I must exist since $M \longrightarrow^* p$). Thus $AI^k \sim M$ but $AI^k \not\sim I^m$ for any m since $I^m \sim p^m$ and $p^m \not\sim M$ (note that $p^m \neq M$ and Δ is in the prime form).

Since $M \longrightarrow_R^* p^m$ for some m, there must be a (shortest) norm-reducing sequence $A \xrightarrow{w} B\beta$ where $\beta \sim I^{\|\beta\|}$, $B \not\sim I^{\|B\|}$ but all norm-reducing transitions $B \xrightarrow{a} \gamma$ satisfy $\gamma \sim I^{\|\gamma\|}$. The sequence $A\alpha \xrightarrow{w} B\beta\alpha$ (where $B\beta\alpha \sim BI^{\|\beta\alpha\|}$) must be matched by some $M \xrightarrow{v} M'$ where M' does not have all tokens in p but every norm-reducing transition from M' results in M'' with all tokens in p; it follows that M' has a single token (so we have at most $|P|$ possibilities for M').

This easily implies that there are at most $|V| \cdot |P| \leq n^2$ markings in Class 4.

□

5 Problem nBPA-nBPP-BISIM is in PTIME

We first note that if moving a token along a cycle C in a BPP system Δ generates new tokens in a place p and C is reachable (markable) from M_0 then p is *primarily unbounded* (in M_0). Any place which is unbounded is either primarily unbounded, or *secondarily unbounded*, which means reachable from a primarily unbounded place. Thus any unbounded place has at least one corresponding *pumping cycle*.

We now characterize when there is no nBPA bisimilar with a given nBPP. We say that SF-place p is *growing* if there is a transition $p \xrightarrow{a} p^k$ for $k \geq 2$.

Lemma 15. *For* (M_0, Δ)*,* Δ *being a normed BPP in the prime form, there is no normed BPA process* (α_0, Σ) *such that* $\alpha_0 \sim M_0$*, iff one of the following conditions holds:*

1. *a non-SF-place is unbounded,*

2. *$M_0 \longrightarrow^* M$ with $|Car(M)| \geq 2$ and $M(p) \geq 1$ for some growing* SF*-place p,*
3. *a non-growing* SF*-place p is unbounded.*

Proof. If 1. is satisfied then we cannot have $\alpha \sim M_0$ (for any Σ with a [finite] variable set V) due to Corollary 8. If 2. or 3. is satisfied then, for any $c \in \mathbb{N}$, $M_0 \longrightarrow^* M$ with $|Car(M)| \geq 2$ and $Tok(M) > c$. (Any pumping cycle for p in 3. contains $p' \neq p$.) Hence we cannot have $\alpha \sim M_0$ due to Proposition 7.

If none of 1.,2.,3. is satisfied, an appropriate (α, Σ) can be constructed as described below. □

We note that the conditions in Lemma 15 can be checked by straightforward standard algorithms, linear in the size of Δ.

5.1 Construction

Suppose now that a given (M_0, Δ) satisfies none of the conditions 1., 2., 3. in Lemma 15. Thus only growing SF-places can be unbounded. Moreover, if some growing SF-place is reachable from M_0 then $Tok(M_0) = 1$ and each transition sequence reaching p just moves the token into p without creating new tokens on the way.

We can construct the usual reachability graph for M_0, with the exception that the "all-in-one-SF" markings p^k are taken as "frozen" – we construct no successors for them. The thus arising *basic LTS* is necessarily finite, and we can view its states as BPA-variables; each unfrozen marking M is viewed as a variable A_M, with the obvious rewriting rules.

To finish the construction, we introduce a variable I_p for each SF-place p together with appropriate rewriting rules.

More formally, for (M_0, Δ) we could construct nBPA system $\Sigma' = (\mathcal{F} \cup \mathcal{I}, \mathcal{A}, \Gamma')$ where $\mathcal{F} = \{A_M \mid M \in \mathcal{M}_{uf}\}$ (where $\mathcal{M}_{uf} = \{M_1, M_2, \ldots, M_m\}$ is the set of unfrozen markings reachable from M_0), $\mathcal{I} = \{I_p \mid p \in P_{SF}\}$ (where $P_{SF} = \{p_1, p_2, \ldots, p_\ell\}$ is the set of SF-places of Δ), and Γ' contains corresponding rewriting rules.

Note that each rule in Γ' is of one of the following three forms: $A_M \xrightarrow{a} A_{M'}$, $A_M \xrightarrow{a} (I_p)^k$, or $I_p \xrightarrow{a} (I_p)^k$ where $A_M, A_{M'} \in \mathcal{F}$, $I_p \in \mathcal{I}$, and $k \in \mathbb{N}$ (this includes also rules of the form $A_M \xrightarrow{a} \varepsilon$ and $I_p \xrightarrow{a} \varepsilon$). Configuration α'_0 corresponding to M_0 will be A_{M_0} (or $(I_{p_0})^k$ when all k tokens in M_0 are in one SF-place p_0). Note that each configuration α reachable from α'_0 is either of the form A_M or $(I_p)^k$, and we have $(\alpha'_0, \Sigma') \sim (M_0, \Delta)$.

Note that the size of (α'_0, Σ') can be exponential with respect to the size of (M_0, Δ), so we will not construct it explicitly in the algorithm.

Assume an instance of nBPA-nBPP-BISIM, i.e., nBPA (α_0, Σ) and nBPP (M_0, Δ). The polynomial algorithm for nBPA-nBPP-BISIM works as follows.

It first transforms (M_0, Δ) to bisimilar (M'_0, Δ') where Δ' is in the prime form; recall Theorem 4. Then it starts to build the nBPA Σ' for (M', Δ') as described above by building the set $\mathcal{M}_{uf}$ of unfrozen states. If it founds out that the number of elements of $\mathcal{M}_{uf}$ exceeds $4n^2$, where n is the maximum of $\{|V_\Sigma|, |P_{\Delta'}|\}$, then the algorithm stops with the answer $\alpha_0 \not\sim M_0$; this is correct due to Theorem 5.

If the number of elements of $\mathcal{M}_{uf}$ does not exceed $4n^2$, the algorithm finishes the construction of Σ'. However, it does not construct Σ' explicitly but rather a succinct representation of it where right hand sides of rules of the form $(I_p)^k$ are represented as pairs (I_p, k) where k is written in binary. (It can be easily shown that the number of bits of every possible k is in $O(n^2)$ where n is the size of (M'_0, Δ').)

Our aim is to apply the polynomial time algorithm from [8] or [9] to decide if $\alpha_0 \sim \alpha'_0$. However, there is a small technical difficulty since this algorithm expects "usual" nBPA, not nBPA in the succinct form described above. This can be handled by adding special variables I_p^1, I_p^2, I_p^4, I_p^8, $\ldots$ $I_p^{2^m}$ for each $I_p \in \mathcal{I}$ and sufficiently large m (in $O(n^2)$); the rules are adjusted in a straightforward way (note that there will be at most $O(m)$ variables on the right hand side of each rewriting rule after this transformation).

The size of the constructed nBPA is surely polynomial with respect to the size of the original instance of the problem and the algorithm from [8] or [9] can be applied.

So we obtained our main theorem:

Theorem 16. *There is a polynomial-time algorithm deciding whether* $(\alpha_0, \Sigma) \sim (M_0, \Delta)$ *where Σ is a normed BPA and Δ a normed BPP.*

Since (α'_0, Σ') is in a very special form (it is a finite state system (FS) extended with "SF-tails"), it is in fact not necessary to use the above mentioned general algorithms. Instead we can use a specialized (and probably more efficient) algorithm sketched in the next subsection.

5.2 Specialized Algorithm

The presented algorithm is an adaptation of the standard technique for deciding bisimilarity for a given BPA (or PDA) and a finite-state system used for example in [14, 15].

Assume we have nBPAs (α_0, Σ) and (α'_0, Σ') where (α_0, Σ) is the nBPA from the instance of nBPA-nBPP-BISIM and (α'_0, Σ') is the nBPA described in the previous subsection (with $V_{\Sigma'} = \mathcal{F} \cup \mathcal{I}$) stored using the succinct representation described above (right hand sides of the form $(I_p)^k$ are stored as pairs (I_p, k) with k represented in binary). Let $V_{all} = V_\Sigma \cup V_{\Sigma'}$.

At first we note that the set of configurations from V_{all}^* bisimilar with $(I_p)^k$ where $I_p \in \mathcal{I}$ can be easily characterized. For each $I_p \in \mathcal{I}$ we construct a set $\mathit{Class}(I_p)$

as the maximal subset of V_{all} such that each $X \in \mathit{Class}(I_p)$ can perform exactly the same actions with the same changes on norm as I_p, and can be rewritten only to variables from $\mathit{Class}(I_p)$ (i.e., $X \xrightarrow{a} \beta$ implies $\beta \in (\mathit{Class}(I_p))^*$, and $I_p \xrightarrow{a} (I_p)^k$ iff $X \xrightarrow{a} \beta$ for some $\beta \in (\mathit{Class}(I_p))^*$ such that $\|\beta\| - \|X\| = k-1$).

The classes $\mathit{Class}(I_p)$ for $I_p \in \mathcal{I}$ can be easily computed in polynomial time. It is not difficult to show that for any $\alpha \in V_{all}^*$ and $I_p \in \mathcal{I}$ we have $\alpha \sim (I_p)^k$ iff $\alpha \in \mathit{Class}(I)^*$ and $\|\alpha\| = k$. Using this fact and precomputed classes $\mathit{Class}(I_p)$, we have a fast (polynomial) test for $\alpha \sim (I_p)^k$.

The crucial observation used in the algorithm is the following. Suppose we want to check if $\alpha \sim A_M$ for some $\alpha \in V_{all}^*$ and $A_M \in \mathcal{F}$ where $\alpha = X\alpha'$ for some $X \in V_{all}$. If $X\alpha' \sim A_M$ then any norm reducing sequence $X\alpha' \longrightarrow_R^* \alpha'$ must be matched by some norm reducing sequence $A_M \longrightarrow_R^* \beta$ such that $\alpha' \sim \beta$. Obviously, β is either of the form $A_{M'}$ (for some $A_{M'} \in \mathcal{F}$) or $(I_p)^k$ (for some $I_p \in \mathcal{I}$). Suppose $\beta = A_{M'}$ (the case $\beta = (I_p)^k$ is similar). Then $\alpha' \sim A_{M'}$. Since $\sim$ is a congruence, we have $XA_{M'} \sim A_M$. On the other hand, if we know that $XA_{M'} \sim A_M$ and $\alpha' \sim A_{M'}$, we know that $X\alpha' \sim A_M$.

By repeating the same approach we can reduce the problem if $\alpha \sim A_M$ to subproblems of testing if $XA_{M'} \sim A_M$, resp. $X(I_p)^k \sim A_M$. Since $\|\alpha\| \neq \|\beta\|$ implies $\alpha \not\sim \beta$, in testing if $X(I_p)^k \sim A_M$ we can consider only those cases where $k = \|A_M\| - \|X\|$. It is obvious that the total number of such subproblems is polynomial with respect to the size of the instance.

The algorithm works by computing the solution for all these subproblems. It approximates from above the set of all such pairs (α, β), where $\alpha \sim \beta$, by computing a fixpoint. It starts with the set of all possible pairs where $\|\alpha\| = \|\beta\|$ and refines it by checking for each pair if it satisfies expansion, i.e., if each transition possible in α is matched by the corresponding transition in β (with respect to the current approximation) and vice versa. (In this checking it also uses the above mentioned test for $\alpha \sim (I_p)^k$.)

Obviously the fixed point is reached after polynomial number of iterations. It is not difficult to check that the resulting fixpoint represents the correct set of pairs which then can be used for computing the answer to the original question if $\alpha_0 \sim A_{M_0}$.

References

1. Burkart, O., Caucal, D., Moller, F., Steffen, B.: Verification on infinite structures. In Bergstra, J., Ponse, A., Smolka, S., eds.: Handbook of Process Algebra. Elsevier Science (2001) 545–623
2. Srba, J.: Roadmap of infinite results. In: Current Trends In Theoretical Computer Science, The Challenge of the New Century. Volume 2: Formal Models and Semantics., World Scientific Publishing Co. (2004) 337–350 (See an updated version at http://www.brics.dk/˜srba/roadmap/).

3. Hirshfeld, Y., Jerrum, M.: Bisimulation equivalence is decidable for normed process algebra. In: Proceedings of 26th International Colloquium on Automata, Languages and Programming (ICALP'99). LNCS 1644, Springer-Verlag (1999) 412–421
4. Burkart, O., Caucal, D., Steffen, B.: An elementary decision procedure for arbitrary context-free processes. In: Proceedings of the 20th International Symposium on Mathematical Foundations of Computer Science (MFCS'95). LNCS 969, Springer-Verlag (1995) 423–433
5. Srba, J.: Strong bisimilarity and regularity of Basic Process Algebra is PSPACE-hard. In: Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP'02). LNCS 2380, Springer (2002) 716–727
6. Jančar, P.: Strong bisimilarity on Basic Parallel Processes is PSPACE-complete. In: Proc. 18th LiCS, IEEE Computer Society (2003) 218–227
7. Srba, J.: Strong bisimilarity and regularity of Basic Parallel Processes is PSPACE-hard. In: Proc. STACS'02. LNCS 2285, Springer (2002) 535–546
8. Hirshfeld, Y., Jerrum, M., Moller, F.: A polynomial algorithm for deciding bisimilarity of normed context-free processes. Theoretical Computer Science **158** (1996) 143–159
9. Lasota, S., Rytter, W.: Faster algorithm for bisimulation equivalence of normed context-free processes. In: Proc. MFCS'06. LNCS 4162, Springer-Verlag (2006) 646–657
10. Hirshfeld, Y., Jerrum, M., Moller, F.: A polynomial-time algorithm for deciding bisimulation equivalence of normed Basic Parallel Processes. Mathematical Structures in Computer Science **6** (1996) 251–259
11. Jančar, P., Kot, M.: Bisimilarity on normed Basic Parallel Processes can be decided in time $O(n^3)$. In Bharadwaj, R., ed.: Proceedings of the Third International Workshop on Automated Verification of Infinite-State Systems – AVIS 2004. (2004)
12. Černá, I., Křetínský, M., Kučera, A.: Comparing expressibility of normed BPA and normed BPP processes. Acta Informatica **36** (1999) 233–256
13. Jančar, P., Kučera, A., Moller, F.: Deciding bisimilarity between BPA and BPP processes. In: Proceedings of CONCUR 2003. LNCS 2761, Springer-Verlag (2003) 159–173
14. Kučera, A., Mayr, R.: Weak bisimilarity between finite-state systems and BPA or normed BPP is decidable in polynomial time. Theoretical Computer Science **270** (2002) 667–700
15. Kučera, A., Mayr, R.: A generic framework for checking semantic equivalences between pushdown automata and finite-state automata. In: IFIP TCS, Kluwer (2004) 395–408