Checking Equality and Regularity for Normed BPA with Silent Moves*

Yuxi Fu

BASICS, Department of Computer Science, Shanghai Jiao Tong University
MOE-MS Key Laboratory for Intelligent Computing and Intelligent Systems

Abstract. The decidability of weak bisimilarity on normed BPA is a long standing open problem. It is proved in this paper that branching bisimilarity, a standard refinement of weak bisimilarity, is decidable for normed BPA and that the associated regularity problem is also decidable.

1 Introduction

In [BBK87] Baeten, Bergstra and Klop proved a surprising result that strong bisimilarity between context free grammars without empty production is decidable. The decidability is in sharp contrast to the well known fact that language equivalence between these grammars is undecidable. After [BBK87] decidability and complexity issues of equivalence checking of infinite systems *à la* process algebra have been intensively investigated. As regards BPA, Hüttel and Stirling [HS91] improved Baeten, Bergstra and Klop's proof by a more straightforward one using tableau system. Hüttel [Hüt92] then repeated the tableau construction for branching bisimilarity on totally normed BPA processes. Later Hirshfeld [Hir96] applied the tableau method to the weak bisimilarity on the totally normed BPA. An affirmative answer to the decidability of the strong bisimilarity on general BPA is given by Christensen, Hüttel and Stirling by applying the technique of bisimulation base [CHS92].

The complexity aspect of BPA has also been investigated over the years. Balcazar, Gabarro and Santha [BGS92] pointed out that strong bisimilarity is P-hard. Huynh and Tian [HT94] showed that the problem is in Σ_2^p, the second level of the polynomial hierarchy. Hirshfeld, Jerrum and Moller [HJM96] completed the picture by offering a remarkable polynomial algorithm for the strong bisimilarity of normed BPA. For the general BPA, Burkart, Caucal and Steffen [BCS95] showed that the strong bisimilarity problem is elementary. They claimed that their algorithm can be optimized to get a 2-EXPTIME upper bound. A further elaboration of the 2-EXPTIME upper bound is given in [Jan12] with the introduction of infinite regular words. The current known best lower bound of the problem, EXPTIME, is obtained by Kiefer [Kie13], improving both the PSPACE lower bound result and its proof of Srba [Srb02]. Much less is known about the weak bisimilarity on BPA. Stříbrná's PSPACE lower bound [Stř98] is subsumed

* F.V. Fomin et al. (Eds.): ICALP 2013, Part II, LNCS 7966, pp. 244-255, 2013.

by both the result of Srba [Srb02] and that of Mayr [May03], all of which are subsumed by Kiefer's recent result. A slight modification of Mayr's proof shows that the EXPTIME lower bound holds for the branching bisimilarity as well.

It is generally believed that weak bisimilarity, as well as branching bisimilarity, on BPA is decidable. There has been however a lack of technique to resolve the difficulties caused by silent transitions. This paper aims to advance our understanding of the decidability problems of BPA in the presence of silent transitions. The main contributions of the paper are as follows:

- We introduce branching norm, which is the least number of nontrivial actions a process has to do to become an empty process. With the help of this concept one can carry out a much finer analysis on silent actions than one would have using weak norm. Branching norm turns out to be crucial in our approach.
- We reveal that in normed BPA the length of a state preserving silent transition sequence can be effectively bounded. As a consequence we show that branching bisimilarity on normed BPA processes can be approximated by a sequence of finite branching bisimulations.
- We establish the decidability of branching bisimilarity on normed BPA by constructing a sound and complete tableau system for the equivalence.
- We demonstrate how to derive the decidability of the associated regularity problem from the decidability of the branching bisimilarity of normed BPA.

The result of this paper is significantly stronger than previous decidability results on the branching bisimilarity of totally normed BPA [Hüt92,CHT95]. It is easy to derive effective size bound for totally normed BPA since a totally normed BPA process with k variable occurrences has a norm at least k. For the same reason right cancellation property holds. Hence the decidability. The totality condition makes the branching bisimilarity a lot more like strong bisimilarity.

2 Branching Bisimilarity for BPA

A *basic process algebra* (BPA for short) Γ is a triple $(\mathcal{V},\mathcal{A},\Delta)$ where $\mathcal{V} = \{X_1,\ldots.X_n\}$ is a finite set of *variables*, $\mathcal{A} = \{a_1,\ldots.a_m\} \cup \{\tau\}$ is a finite set of *actions* ranged over by ℓ, and Δ is a finite set of *transition rules*. The special symbol τ denotes a *silent* action. A *BPA process* defined in Γ is an element of the set $\mathcal{V}^*$ of finite string of element of $\mathcal{V}$. The set $\mathcal{V}$ will be ranged over by capital letters and $\mathcal{V}^*$ by lower case Greek letters. The empty string is denoted by ϵ. We will use $=$ for the grammar equality on $\mathcal{V}^*$. A transition rule is of the form $X \stackrel{\ell}{\longrightarrow} \alpha$, where ℓ ranges over $\mathcal{A}$. The transitional semantics is closed under *composition* in the sense that $X\gamma \stackrel{\ell}{\longrightarrow} \alpha\gamma$ for all γ whenever $X \stackrel{\ell}{\longrightarrow} \alpha$. We shall assume that every variable of a BPA is defined by at least one transition rule and every action in $\mathcal{A}$ appears in some transition rule. Accordingly we sometimes refer to a BPA by its set of transition rules. We write $\longrightarrow$ for $\stackrel{\tau}{\longrightarrow}$ and $\Longrightarrow$ for the reflexive transitive closure of $\stackrel{\tau}{\longrightarrow}$. The set $\mathcal{A}^*$ will be ranged over by ℓ^*. If $\ell^* = \ell_1 \ldots \ell_k$ for some $k \geq 0$, then $\alpha \stackrel{\ell^*}{\longrightarrow} \alpha'$ stands for $\alpha \stackrel{\ell_1}{\longrightarrow} \alpha_1 \ldots \stackrel{\ell_{k-1}}{\longrightarrow} \alpha_{k-1} \stackrel{\ell_k}{\longrightarrow} \alpha'$ for some $\alpha_1,\ldots,\alpha_{k-1}$. We say that α' is a *descendant* of α if $\alpha \stackrel{\ell^*}{\longrightarrow} \alpha'$ for some ℓ^*.

A BPA process α is *normed* if there are some actions $\ell_1, \dots \ell_j$ such that $\alpha \xrightarrow{\ell_1} \dots \xrightarrow{\ell_j} \epsilon$. A process is *unnormed* if it is not normed. The *norm* of a BPA process α, denoted by $\|\alpha\|$, is the least k such that $\alpha \xrightarrow{\ell_1} \dots \xrightarrow{\ell_k} \epsilon$ for some $\ell_1, \dots \ell_k$. A *normed BPA*, or *nBPA*, is one in which every variable is normed.

For each given BPA Δ, we introduce the following notations:

- m_Δ is the number of transition rules; and n_Δ is the number of variables.
- r_Δ is max $\left\{ |\gamma| \,\middle|\, X \xrightarrow{\lambda} \gamma \in \Delta \right\}$, where $|\gamma|$ denotes the length of γ.
- $\|\Delta\|$ is max $\{\|X_i\| \mid 1 \le i \le n_\Delta$ and X_i is normed$\}$.

Each of m_Δ, n_Δ, r_Δ and $\|\Delta\|$ can be effectively calculated from Δ.

2.1 Branching Bisimilarity

The idea of the branching bisimilarity of van Glabbeek and Weijland [vGW89] is that not all silent actions can be ignored. What can be ignored are those that do not change system states irreversibly. For BPA we need to impose additional condition to guarantee congruence. In what follows $x\mathcal{R}y$ stands for $(x, y) \in \mathcal{R}$.

Definition 1. *A symmetric relation $\mathcal{R}$ on BPA processes is a* branching bisimulation *if the following statements are valid whenever $\alpha\mathcal{R}\beta$:*

1. *If $\beta\mathcal{R}\alpha \xrightarrow{\ell} \alpha'$ then one of the following statements is valid:*
   (i) $\ell = \tau$ and $\alpha'\mathcal{R}\beta$.
   (ii) $\beta \Longrightarrow \beta''\mathcal{R}\alpha$ for some β'' such that $\beta'' \xrightarrow{\ell} \beta'\mathcal{R}\alpha'$ for some β'.
2. *If $\alpha = \epsilon$ then $\beta \Longrightarrow \epsilon$.*

The branching bisimilarity $\simeq$ *is the largest branching bisimulation.*

The branching bisimilarity $\simeq$ satisfies the standard property of observational equivalence stated in the next lemma [vGW89].

Lemma 1. *Suppose $\alpha_0 \xrightarrow{\tau} \alpha_1 \xrightarrow{\tau} \dots \xrightarrow{\tau} \alpha_k \simeq \alpha_0$. Then $\alpha_0 \simeq \alpha_1 \simeq \dots \simeq \alpha_k$.*

Using Lemma 1 it is easy to show that $\simeq$ is a congruence and that whenever $\beta \simeq \alpha \xrightarrow{\ell} \alpha'$ is simulated by $\beta \xrightarrow{\tau} \beta_1 \xrightarrow{\tau} \beta_2 \dots \xrightarrow{\tau} \beta_k \xrightarrow{\ell} \beta'$ such that $\beta_k \simeq \alpha$ and $\beta' \simeq \alpha'$ then $\beta \simeq \beta_1 \simeq \dots \simeq \beta_k$.

Having defined an equality for BPA, we can formally draw a line between the silent actions that change the capacity of systems and those that do not. We say that a silent action $\alpha \xrightarrow{\tau} \alpha'$ is *state preserving* if $\alpha \simeq \alpha'$; it is a *change-of-state* if $\alpha \not\simeq \alpha'$. We will write $\alpha \to \alpha'$ if $\alpha \xrightarrow{\tau} \alpha'$ is state preserving and $\alpha \xrightarrow{\iota} \alpha'$ if it is a change-of-state. The reflexive and transitive closure of $\to$ is denoted by $\to^*$. Since both external actions and change-of-state silent actions must be explicitly bisimulated, we let $\jmath$ range over the set $(\mathcal{A} \setminus \{\tau\}) \cup \{\iota\}$. So $\alpha \xrightarrow{\jmath} \alpha'$ means either $\alpha \xrightarrow{a} \alpha'$ for some $a \neq \tau$ or $\alpha \xrightarrow{\iota} \alpha'$.

Let's see an example.

Example 1. The BPA Γ_1 is defined by the following transition rules:

$$A \xrightarrow{a} A,\ A \xrightarrow{\tau} \epsilon,\ B \xrightarrow{b} B,\ B \xrightarrow{\tau} \epsilon,\ C \xrightarrow{a} C,\ C \xrightarrow{b} C,\ C \xrightarrow{\tau} \epsilon.$$

Clearly $AC \simeq BC$, although $A \not\simeq B$. In this example all variables are normed.

2.2 Bisimulation Base

An *axiom system* $\mathcal{B}$ is a finite set of equalities on nBPA processes. An element $\alpha = \beta$ of $\mathcal{B}$ is called an *axiom*. Write $\mathcal{B} \vdash \alpha = \beta$ if the equality $\alpha = \beta$ can be derived from the axioms of $\mathcal{B}$ by repetitive use of any of the three equivalence rules and two congruence rules. For our purpose the most useful axiom systems are those that generate branching bisimulations. These are bisimulation bases originally due to Caucal. The following definition is Hüttel's adaptation to the branching scenario [Hüt92].

Definition 2. *A finite axiom system $\mathcal{B}$ is a* bisimulation base *if the following bisimulation base property hold for every axiom (α_0, β_0) of $\mathcal{B}$:*

1. *If $\beta_0 \longrightarrow \beta_1 \longrightarrow \ldots \longrightarrow \beta_n \xrightarrow{\ell} \beta'$ then there are $\alpha_1, \ldots, \alpha_n, \alpha'$ such that $\mathcal{B} \vdash \beta_1 = \alpha_1$, $\ldots$, $\mathcal{B} \vdash \beta_n = \alpha_n$, $\mathcal{B} \vdash \beta' = \alpha'$ and the following hold:*
   (i) For each i with $0 \leq i < n$, either $\alpha_i = \alpha_{i+1}$, or $\alpha_i \longrightarrow \alpha_{i+1}$, or there are $\alpha_i^1, \ldots, \alpha_i^{k_i}$ such that $\alpha_i \longrightarrow \alpha_i^1 \longrightarrow \ldots \longrightarrow \alpha_i^{k_i} \longrightarrow \alpha_{i+1}$ and $\mathcal{B} \vdash \beta_i = \alpha_i^1$, $\ldots$, $\mathcal{B} \vdash \beta_i = \alpha_i^{k_i}$.
   (ii) Either $\ell = \tau$ and $\alpha_n = \alpha'$, or $\alpha_n \xrightarrow{\ell} \alpha'$, or there are $\alpha_n^1, \ldots, \alpha_n^{k_n}$ such that $\alpha_n \longrightarrow \alpha_n^1 \longrightarrow \ldots \longrightarrow \alpha_n^{k_n} \xrightarrow{\ell} \alpha'$ and $\mathcal{B} \vdash \beta_n = \alpha_n^1$, $\ldots$, $\mathcal{B} \vdash \beta_n = \alpha_n^{k_n}$.
2. *If $\beta_0 = \epsilon$ then either $\alpha_0 = \epsilon$ or $\alpha_0 \longrightarrow \alpha_1 \longrightarrow \ldots \longrightarrow \alpha_k \longrightarrow \epsilon$ for some $\alpha_1, \ldots, \alpha_k$ with $k \geq 0$ such that $\mathcal{B} \vdash \alpha_1 = \epsilon$, $\ldots$, $\mathcal{B} \vdash \alpha_k = \epsilon$.*
3. *The conditions symmetric to 1 and 2.*

The next lemma justifies the above definition [Hüt92].

Lemma 2. *If $\mathcal{B}$ is a bisimulation base then $\mathcal{B}^{\vdash} = \{(\alpha, \beta) \mid \mathcal{B} \vdash \alpha = \beta\} \subseteq \simeq$.*

Proof. If $\mathcal{B} \vdash \alpha = \beta$, then an inductive argument shows that there exist $\gamma_1\delta_1\lambda_1$, $\gamma_2\delta_2\lambda_2$, $\gamma_3\delta_3\lambda_3, \ldots, \gamma_{k-1}\delta_{k-1}\lambda_{k-1}$, $\gamma_k\delta_k\lambda_k$ and $\delta'_1, \ldots, \delta'_k$ for $k \geq 1$ such that $\alpha = \gamma_1\delta_1\lambda_1$, $\gamma_k\delta'_k\lambda_k = \beta$ and $\gamma_1\delta_1\lambda_1 \ \mathcal{B} \ \gamma_1\delta'_1\lambda_1 = \gamma_2\delta_2\lambda_2 \ \mathcal{B} \ \gamma_2\delta'_2\lambda_2 = \ldots \gamma_{k-1}\delta'_{k-1}\lambda_{k-1} = \gamma_k\delta_k\lambda_k \ \mathcal{B} \ \gamma_k\delta'_k\lambda_k$. The transitive closure makes it easy to see that $\mathcal{B}^{\vdash}$ satisfies the bisimulation base property. Consequently it is a branching bisimulation. $\square$

3 Approximation of Branching Bisimilarity

To look at the algebraic property of the branching bisimilarity $\simeq$ more closely, we introduce a notion of normedness appropriate for the equivalence.

Definition 3. *The* branching norm *of an nBPA process α is the least number k such that $\exists \jmath_1 \ldots \jmath_k . \exists \alpha_1 \ldots \alpha_k . \alpha \rightarrow^* \xrightarrow{\jmath_1} \alpha_1 \rightarrow^* \xrightarrow{\jmath_2} \ldots \alpha_{k-1} \rightarrow^* \xrightarrow{\jmath_k} \alpha_k \rightarrow^* \epsilon$. The branching norm of α is denoted by $\|\alpha\|_b$.*

For example the branching norm of B defined by $\{B \xrightarrow{a} B, B \xrightarrow{\tau} \epsilon\}$ is 1. It is easy to prove that if $\alpha \simeq \beta$ then $\|\alpha\|_b = \|\beta\|_b$ and that if $\|\alpha\|_b = 0$ then $\alpha \simeq \epsilon$. It follows that $\|\alpha'\|_b = \|\alpha\|_b$ whenever $\alpha \rightarrow^* \alpha'$. Also notice that $\|\alpha\|_b \leq \|\alpha\|$.

An important property of branching norm is stated next.

Lemma 3. *Suppose α is normed. Then $\alpha \simeq \delta\alpha$ if and only if $\|\alpha\|_b = \|\delta\alpha\|_b$.*

Proof. If $\|\alpha\|_b = \|\delta\alpha\|_b$ then every silent action sequence from $\delta\alpha$ to α must contain only state preserving silent transitions according to Lemma 1. Moreover there must exist such a silent action path for otherwise $\|\alpha\|_b < \|\delta\alpha\|_b$. $\square$

It does not follow from $\alpha \simeq \delta\alpha$ that $\delta \simeq \epsilon$. A counter example is given by the BPA defined in Example 1. One has $AC \simeq C \simeq BC$. But clearly $\epsilon \not\simeq A \not\simeq B \not\simeq \epsilon$. To deal with situations like this we need the notion of relative norm.

Definition 4. *The* relative norm $\|\alpha\|_b^\sigma$ *of α with respect to σ is the least k such that $\alpha\sigma \rightarrow^* \xrightarrow{\jmath_1} \alpha_1\sigma \ldots \alpha_{k-1}\sigma \rightarrow^* \xrightarrow{\jmath_k} \alpha_k\sigma \rightarrow^* \sigma$ for some $\jmath_1, \ldots, \jmath_k, \alpha_1, \ldots, \alpha_k$.*

Obviously $0 \leq \|\alpha\|_b^\sigma \leq \|\alpha\|_b$. Returning to the BPA Γ_1 defined in Example 1, we see that $\|A\|_b^B = 1$ and $\|A\|_b^C = 0$. Using the notion of relative norm we may introduce the following terminologies:

- A transition $X\sigma \xrightarrow{\ell} \eta\sigma$ is *norm consistent* if either $\|\eta\|_b^\sigma = \|X\|_b^\sigma$ and $\ell = \tau$ or $\|\eta\|_b^\sigma = \|X\|_b^\sigma - 1$ and $\ell \neq \tau \vee \ell = \iota$.
- If $X\sigma \longrightarrow \eta\sigma$ is norm consistent with $\|X\|_b^\sigma > 0$, then it is *norm splitting* if at least two variables in η have (smaller) nonzero relative norms in $\eta\sigma$.

For an nBPA Δ no silent transition sequence contains more than $\|\Delta\|_b$ norm splitting transitions, where $\|\Delta\|_b$ is $\max\{\|X_i\|_b \mid 1 \leq i \leq n_\Delta \text{ and } X_i \text{ is normed}\}$.

The crucial property about relative norm is described in the following lemma.

Lemma 4. *Let $\alpha, \beta, \delta, \gamma$ be normed with $\|\alpha\|_b^\gamma = \|\beta\|_b^\delta$. If $\alpha\gamma \simeq \beta\delta$ then $\gamma \simeq \delta$.*

Proof. Suppose $\|\alpha\|_b^\gamma = \|\beta\|_b^\delta$. Now $\|\alpha\|_b^\gamma + \|\gamma\|_b = \|\alpha\gamma\|_b = \|\beta\delta\|_b = \|\beta\|_b^\delta + \|\delta\|_b$. Therefore $\|\gamma\|_b = \|\delta\|_b$. A norm consistent action sequence $\alpha\gamma \rightarrow^* \xrightarrow{\jmath_1} \ldots \rightarrow^* \xrightarrow{\jmath_k} \rightarrow^* \gamma$ must be matched up by $\beta\delta \rightarrow^* \xrightarrow{\jmath_1} \ldots \rightarrow^* \xrightarrow{\jmath_k} \beta'\delta$ for some β'. Clearly $\|\beta'\delta\|_b = \|\gamma\|_b = \|\delta\|_b$. It follows from Lemma 3 that $\delta \simeq \beta'\delta \simeq \gamma$. $\square$

Lemma 4 describes a weak form of left cancelation property. The general left cancelation property fails. Fortunately there is a nice property of nBPA that allows us to control the size of common suffix of a pair of bisimilar processes.

Definition 5. *A process α is* irredundant *over γ if $\|\alpha\|_b^\gamma > 0$. It is* redundant *over γ if $\|\alpha\|_b^\gamma = 0$. A process α is* head irredundant *if either $\alpha = \epsilon$ or $\alpha = X\alpha'$ for some X, α' such that $\alpha \not\simeq \alpha'$. It is head redundant otherwise. We write $Hirred(\alpha)$ to indicate that α is head irredundant. A process α is* completely irredundant *if every suffix of α is head irredundant. We write $Cirred(\alpha)$ to mean that α is completely irredundant.*

If α is normed, then α is irredundant over γ if and only if $\alpha\gamma \not\simeq \gamma$. In nBPA a redundant process consists solely of redundant variables.

Lemma 5. *Suppose $X_1, \ldots, X_k, \sigma$ are normed. Then $X_1 \ldots X_k$ is redundant over σ if and only if X_i is redundant over σ for every $X_i \in \{X_1, \ldots, X_k\}$.*

Proof. Suppose $X_1, \ldots, X_k, \sigma$ are normed and $X_1 \ldots X_k$ is redundant over σ. Then $X_1 \ldots X_k\sigma \Longrightarrow X_2 \ldots X_k\sigma \Longrightarrow \ldots \Longrightarrow X_k\sigma \Longrightarrow \sigma \simeq X_1 \ldots X_k\sigma$. It follows from Lemma 1 that $X_1 \ldots X_k\sigma \simeq X_2 \ldots X_k\sigma \simeq \ldots \simeq X_k\sigma \simeq \sigma$. We are done by using the congruence property. $\square$

For each σ, let the *redundant set* $\mathcal{R}_\sigma$ of σ be $\{X \mid X\sigma \simeq \sigma\}$. Let $\mathcal{V}(\alpha)$ be the set of variables appearing in α. We have two useful corollaries.

Corollary 1. *Suppose α, σ are normed. Then $\alpha\sigma \simeq \sigma$ if and only if $\mathcal{V}(\alpha) \subseteq \mathcal{R}_\sigma$.*

Corollary 2. *Suppose $\alpha, \beta, \sigma_0, \sigma_1$ are defined in an nBPA and $\mathcal{R}_{\sigma_0} = \mathcal{R}_{\sigma_1}$. Then $\alpha\sigma_0 \simeq \beta\sigma_0$ if and only if $\alpha\sigma_1 \simeq \beta\sigma_1$.*

Proof. Suppose $\mathcal{R}_{\sigma_0} = \mathcal{R}_{\sigma_1}$. Let $\mathcal{S}$ be $\{(\alpha\sigma_0, \beta\sigma_0) \mid \alpha\sigma_1 \simeq \beta\sigma_1\}$. It is not difficult to see that $\mathcal{S} \cup \simeq$ is a branching bisimulation. $\square$

We now take a look at the state preserving transitions of nBPA processes. We are particularly interested in knowing if the quotient set $\{\theta \mid \alpha \rightarrow^* V\theta\}/\simeq$ of the equivalence classes is finite for every nBPA process α and every variable V. It turns out that all such sets are finite with effective size bound.

Lemma 6. *For each nBPA process $\alpha = X\omega$, there is an effective bound H_α, uniformly computable from α, satisfying the following: If $\alpha \rightarrow^* V\theta$ then $\alpha \rightarrow^* V\eta$ for some η such that $\theta \simeq \eta$ and the length of $\alpha \rightarrow^* V\eta$ is no more than H_α.*

Proof. The basic idea is to show that in an effectively bounded number of steps α can reach, via norm consistent and norm splitting silent transitions, terms $V\theta$ with all possible variable V and all possible relative norm of V. We then apply Lemma 4. The bound H_α is computed from $|\alpha|$ and the transition system. $\square$

Under the assumption $\gamma \not\simeq \beta\gamma$ we can repeat the proof of Lemma 6 for $\beta\gamma$ in a way that γ is not affected. Hence the next corollary.

Corollary 3. *Suppose $\alpha, \beta\gamma$ are nBPA processes and $\gamma \not\simeq \beta\gamma$. If $\beta\gamma \simeq \alpha \xrightarrow{\jmath} \alpha'$, then there is a transition sequence $\beta\gamma \rightarrow^* \beta''\gamma \xrightarrow{\jmath} \beta'\gamma$ with its length bounded by H_β such that $\beta''\gamma \simeq \alpha$ and $\beta'\gamma \simeq \alpha'$.*

We are now in a position to prove the following.

Proposition 1. *The relation $\not\simeq$ on nBPA processes is semi-decidable.*

Proof. We define $\simeq_k$, the branching bisimilarity up to depth k, by exploiting Corollary 3. The inductive definition is as follows:

- $\alpha \simeq_0 \beta$ for all α, β.
- $\alpha \simeq_{i+1} \beta$ if the following condition and its symmetric version hold: If $\alpha \simeq_i \beta \xrightarrow{\ell} \beta'$ then one of the following statements is valid:
  (i) $\ell = \tau$ and $\alpha \simeq_i \beta'$.
  (ii) $\alpha \Longrightarrow \alpha'' \simeq_i \beta$ for some α'' such that $\alpha'' \xrightarrow{\ell} \alpha' \simeq_i \beta'$ for some α' and the length of $\alpha \Longrightarrow \alpha''$ is bounded by H_α.

Each $\simeq_k$ is decidable. Using Corollary 3 one easily sees that $\simeq \subseteq \bigcap_{k\in\omega} \simeq_k$. The proof of the converse inclusion is standard. $\square$

4 Equality Checking

A straightforward approach to proving an equality between two processes is to construct a finite bisimulation tree for the equality. A tree of this kind has been called a tableau system [HS91,Hüt92]. To apply this approach we need to make sure that the following properties are satisfied: (i) Every tableau for an equality $\alpha = \beta$ is finite. (ii) The set of tableaux for an equality $\alpha = \beta$ is finite. We can achieve (i) by using Corollary 2 and Corollary 3. This is because if σ is long enough then according to Corollary 2 it can be decomposed into some $\sigma_0\sigma_1\sigma_2$ such that $\mathcal{R}_{\sigma_1\sigma_2} = \mathcal{R}_{\sigma_2}$. Then $\lambda\sigma_0\sigma_1\sigma_2 \simeq \gamma\sigma_0\sigma_1\sigma_2$ can be simplified to $\lambda\sigma_0\sigma_2 \simeq \gamma\sigma_0\sigma_2$. The equivalence provides a method to control the size of labels of a tableau. Now (ii) is a consequence of (i), Corollary 3 and König lemma.

The building blocks for tableaux are matches. Suppose $\alpha_0\alpha \not\simeq \alpha$ and $\beta_0\beta \not\simeq \beta$. A *match* for the equality $\alpha_0\alpha = \beta_0\beta$ *over* (α, β) is a finite symmetric relation $\{\gamma_i\alpha = \lambda_i\beta\}_{i=1}^k$ containing only those equalities accounted for in the following condition: For each transition $\alpha_0\alpha \xrightarrow{\ell} \alpha'\alpha$, one of the following holds:

- $\ell = \tau$ and $\alpha'\alpha = \beta_0\beta \in \{\gamma_i\alpha = \lambda_i\beta\}_{i=1}^k$;
- there is a sequence $\beta_0\beta \xrightarrow{\tau} \beta_1\beta \xrightarrow{\tau} \ldots \xrightarrow{\tau} \beta_n\beta \xrightarrow{\ell} \beta'\beta$, for $n < H_{\beta_0}$, such that $\{\alpha_0\alpha = \beta_1\beta, \ldots, \alpha_0\alpha = \beta_n\beta, \alpha'\alpha = \beta'\beta\} \subseteq \{\gamma_i\alpha = \lambda_i\beta\}_{i=1}^k$.

If $\alpha_0\sigma \not\simeq \sigma \not\simeq \beta_0\sigma$, a match for $\alpha_0\sigma = \beta_0\sigma$ over (σ, σ) is said to be a match for $\alpha_0\sigma = \beta_0\sigma$ over σ. The computable bound H_{β_0}, given by Corollary 3, guarantees that the number of matches for $\alpha_0\alpha = \beta_0\beta$ is effectively bounded.

Suppose α_0, β_0 are nBPA processes. A *tableau* for $\alpha_0 = \beta_0$ is a tree with each of its nodes labeled by an equality between nBPA processes. The root is labeled by $\alpha_0 = \beta_0$. We shall distinguish between *global tableau* and *local tableau.* The global tableau is the overall tableau whose root is labeled by the goal $\alpha_0 = \beta_0$. It is constructed from the rules given in Fig. 1. Decmp rule decomposes a goal into several subgoals. We shall find it useful to use SDecmp, which is a stronger version of Decmp. The side condition of SDecmp ensures that it is unnecessary to apply it consecutively. When applying Decmp rule we assume that an equality $\gamma\sigma = \sigma$, respectively $\sigma = \gamma\sigma$, is always decomposed in the following manner

$$\frac{\gamma\sigma = \sigma}{\sigma = \sigma \qquad \{V\sigma = \sigma\}_{V\in\mathcal{V}(\gamma)}} \text{ respectively } \frac{\sigma = \gamma\sigma}{\sigma = \sigma \qquad \{V\sigma = \sigma\}_{V\in\mathcal{V}(\gamma)}}.$$

Accordingly $\gamma = \epsilon$, respectively $\epsilon = \gamma$, is decomposed in the following fashion

$$\frac{\gamma = \epsilon}{\epsilon = \epsilon \qquad \{V = \epsilon\}_{V\in\mathcal{V}(\gamma)}} \text{ respectively } \frac{\epsilon = \gamma}{\epsilon = \epsilon \qquad \{V = \epsilon\}_{V\in\mathcal{V}(\gamma)}}.$$

SubstL and SubstR allow one to create common suffix for the two processes in an equality. ContrL and ContrR are used to remove a redundant variable inside a process. In the side conditions of these two rules, α_0, β_0 are the processes appearing in the root of the global tableau. ContrC deletes redundant variables from the common suffix of a node label whenever the size of the common suffix

Decmp	$\dfrac{\gamma\alpha = \lambda\beta}{\alpha = \beta \qquad \{U\alpha = \alpha\}_{U \in \mathcal{V}(\gamma)} \qquad \{V\beta = \beta\}_{V \in \mathcal{V}(\lambda)}}$	$\lvert\gamma\rvert + \lvert\lambda\rvert > 0$, $\forall U \in \mathcal{V}(\gamma).U \Longrightarrow \epsilon$, $\forall V \in \mathcal{V}(\lambda).V \Longrightarrow \epsilon$.
SDecmp	$\dfrac{\gamma\alpha = \lambda\beta}{\alpha = \beta \qquad \{U\alpha = \alpha\}_{U \in \mathcal{V}(\gamma)} \qquad \{V\beta = \beta\}_{V \in \mathcal{V}(\lambda)}}$	$\lvert\gamma\rvert + \lvert\lambda\rvert > 0$, $Hirred(\alpha),\ Hirred(\beta)$, $\forall U \in \mathcal{V}(\gamma).U \Longrightarrow \epsilon$, $\forall V \in \mathcal{V}(\lambda).V \Longrightarrow \epsilon$.
Match	$\dfrac{\gamma\alpha = \lambda\beta}{\alpha_1\alpha = \beta_1\beta \ \ldots \ \alpha_k\alpha = \beta_k\beta}$	$\gamma\alpha \not\simeq \alpha,\ \lambda\beta \not\simeq \beta,$ and $\{\alpha_i\alpha = \beta_i\beta\}_{i=1}^k$ is a match for $\gamma\alpha = \lambda\beta$ over (α, β).
SubstL	$\dfrac{\gamma\alpha = \lambda\beta}{\gamma\delta\beta = \lambda\beta}$	$\alpha = \delta\beta$ is the residual.
SubstR	$\dfrac{\gamma\alpha = \lambda\beta}{\gamma\alpha = \lambda\delta\alpha}$	$\delta\alpha = \beta$ is the residual.
ContrL	$\dfrac{\gamma Z\delta = \lambda}{\gamma\delta = \lambda \qquad Z\delta = \delta}$	$Hirred(\delta),\ Z \Longrightarrow \epsilon$ and $\lvert\gamma Z\delta\rvert > \max\{\lvert\alpha_0\rvert, \lvert\beta_0\rvert\}\lVert\Delta\rVert$.
ContrR	$\dfrac{\gamma = \lambda Z\delta}{\gamma = \lambda\delta \qquad Z\delta = \delta}$	$Hirred(\delta),\ Z \Longrightarrow \epsilon$ and $\lvert\lambda Z\delta\rvert > \max\{\lvert\alpha_0\rvert, \lvert\beta_0\rvert\}\lVert\Delta\rVert$.
ContrC	$\dfrac{\gamma\sigma'\sigma_0\sigma_1 = \lambda\sigma'\sigma_0\sigma_1}{\gamma\sigma'\sigma_1 = \lambda\sigma'\sigma_1 \qquad \{V\sigma_1 = \sigma_1\}_{V \in \mathcal{V}(\sigma_0)}}$	$\lvert\sigma'\sigma_0\sigma_1\rvert > 2^{n_\Delta},\ \lvert\sigma_0\rvert > 0$, $Hirred(\sigma_1)$, $\forall V \in \mathcal{V}(\sigma_0).V \Longrightarrow \epsilon$.

Fig. 1. Rules for Global Tableaux

is over limit. Notice that all the side conditions on the rules are semi-decidable due to the semi-decidability of $\not\simeq$. So we can effectively enumerate tableaux.

In what follows a node $Z\eta = W\kappa$ to which Match rule is applied with the condition $Z\eta \not\simeq \eta \wedge W\kappa \not\simeq \kappa$ is called an *M-node*. A node of the form $Z\sigma = \sigma$ with σ being head irredundant is called a *V-node*. We now describe how a global tableau for $\alpha_0 = \beta_0$ is constructed. Assuming $\alpha_0 = \gamma X\alpha_1$ and $\beta_0 = \lambda Y\beta_1$ such that $X\alpha_1 \not\simeq \alpha_1$ and $Y\beta_1 \not\simeq \beta_1$, we apply the following instance of SDecmp rule:

$$\frac{\gamma X\alpha_1 = \lambda Y\beta_1}{X\alpha_1 = Y\beta_1 \qquad \{UX\alpha_1 = X\alpha_1\}_{U \in \mathcal{V}(\gamma)} \qquad \{VY\beta_1 = Y\beta_1\}_{V \in \mathcal{V}(\lambda)}}.$$

By definition $X\alpha_1 = Y\beta_1$ is an M-node and $\{UX\alpha_1 = X\alpha_1\}_{U \in \mathcal{V}(\gamma)} \cup \{VY\beta_1 = Y\beta_1\}_{V \in \mathcal{V}(\lambda)}$ is a set of V-nodes. These nodes are the roots of new subtableaux. Starting from $X\alpha_1 = Y\beta_1$ we apply Match rule under the condition that neither α_1 nor β_1 is affected. The application of Match rule is repeated to grow the subtableau rooted at $X\alpha_1 = Y\beta_1$. The construction of the tree is done in a breadth first fashion. So the tree grows level by level. At some stage we apply Decmp rule to all the current leaves. This particular application of Decmp must meet the following conditions: (i) Both α_1 and β_1 must be kept intact in all the current leaves; (ii) Either α_1 or β_1 is exposed in at least one current leaf. Choose a leaf labeled by either $\alpha_1 = \delta_1\beta_1$ for some δ_1 or by $\delta_1'\alpha_1 = \beta_1$ for some δ_1' and call it the *residual node* or *R-node*. Suppose the residual node is $\alpha_1 = \delta_1\beta_1$. All the other current leaves, the *non-residual nodes*, must be labeled by an equality of the form $\gamma_1\alpha_1 = \lambda_1\beta_1$. A non-residual node with label $\gamma_1\alpha_1 = \lambda_1\beta_1$ is then

Localization	$\dfrac{\gamma\sigma'\sigma_0\sigma_1 = \lambda\sigma'\sigma_0\sigma_1}{\begin{array}{l}\gamma\sigma'\sigma_1 = \lambda\sigma'\sigma_1 \\ \{X_i\sigma_1 = \sigma_1\}_{i\in I} \\ \{X_i\sigma_0\sigma_1 = \sigma_0\sigma_1\}_{i\in I}\end{array}}$	$\lvert\gamma\rvert > 0$ and $\lvert\lambda\rvert > 0$; $\lvert\sigma'\sigma_0\sigma_1\rvert > 2^{n_\Delta}$, $2^{n_\Delta} \geq \lvert\sigma_1\rvert > 0$ and $\lvert\sigma_0\rvert > 0$; $Cirred(\sigma'\sigma_0\sigma_1)$ and $Cirred(\sigma'\sigma_1)$; $\gamma\sigma'\sigma_0\sigma_1 \not\approx \sigma'\sigma_0\sigma_1,\ \gamma\sigma'\sigma_1 \not\approx \sigma'\sigma_1$; $\lambda\sigma'\sigma_0\sigma_1 \not\approx \sigma'\sigma_0\sigma_1,\ \lambda\sigma'\sigma_1 \not\approx \sigma'\sigma_1$; $I \cap J = \emptyset,\ I \cup J = \{1,\ldots,n_\Delta\}$; $\forall j \in J.\ X_j\sigma_0\sigma_1 \not\approx \sigma_0\sigma_1$ and $X_j\sigma_1 \not\approx \sigma_1$; $X_i \Longrightarrow \epsilon$ for all $i \in I$.

Fig. 2. Rule for Local Tableaux

attached with a single child labeled by $\gamma_1\delta_1\beta_1 = \lambda_1\beta_1$. This is an application of SubstL rule. Now we can recursively apply the global tableau construction to $\gamma_1\delta_1\beta_1 = \lambda_1\beta_1$ to produce a new subtableau. The treatment of a V-node child, say $UX\alpha_1 = X\alpha_1$, is similar. We keep applying Match rule over α_1 as long as the side condition is met. At certain stage we apply Decmp rule to all the leaves. The application should meet the following conditions: (i) No occurrence of α_1 is affected; (ii) There is an application of Decmp that takes the following shape

$$\frac{\gamma_1\alpha_1 = \lambda_1\alpha_1}{\alpha_1 = \alpha_1 \qquad \{V\alpha_1 = \alpha_1\}_{V\in\mathcal{V}(\gamma_1)} \qquad \{V\alpha_1 = \alpha_1\}_{V\in\mathcal{V}(\lambda_1)}}.$$

We then recursively apply the tableau construction to create new subtableaux.

In the above construction the R-node $\alpha_1 = \delta_1\beta_1$ can be the root of a new subtableau, which might contain another R-node. In fact a chain of R-nodes is possible. ContrL/ContrR is used to control the size of R-nodes.

After an application of SubstL/SubstR rule we may get a *C-node* $\alpha'\sigma'\sigma_0\sigma_1 = \beta'\sigma'\sigma_0\sigma_1$ if ContrC rule is applicable. Once a C-node appears, we immediately apply ContrC rule to reduce the size of its common suffix. Intuitively we should apply ContrC rule sufficiently often so that the common suffix becomes completely irredundant. Eventually either the length of the common suffix has become no more than 2^{n_Δ}, in which case we continue to build up the global tableau, or Localization rule as defined in Fig. 2 is applicable, in which case we get an *L-node*. The soundness of Localization rule is guaranteed by Corollary 2.

Suppose Localization rule is applied to an L-node $\alpha'\sigma'\sigma_0\sigma_1 = \beta'\sigma'\sigma_0\sigma_1$:

$$\frac{\alpha'\sigma'\sigma_0\sigma_1 = \beta'\sigma'\sigma_0\sigma_1}{\{X_i\sigma_1 = \sigma_1\}_{i\in I} \qquad \alpha'\sigma'\sigma_1 = \beta'\sigma'\sigma_1 \qquad \{X_i\sigma_0\sigma_1 = \sigma_0\sigma_1\}_{i\in I}}.$$

The node $\alpha'\sigma'\sigma_1 = \beta'\sigma'\sigma_1$ is a new L-node. We call $\{X_i \mid i \in I\}$ the *R-set* of the new L-node. If the size of the common suffix of $\alpha'\sigma'\sigma_1 = \beta'\sigma'\sigma_1$ is still larger than 2^{n_Δ}, we continue to apply Localization rule. Otherwise we get an *L-root*, which is the root of a local tableau. Now suppose $\alpha'\sigma'\sigma_1 = \beta'\sigma'\sigma_1$ is an L-root. The construction of the local tableau should stick to two principles described as follows: (I) *Locality.* No application of Decmp, SDecmp, SubstL, SubsR and ContrC should ever affect $\sigma'\sigma_1$ or any suffix of $\sigma'\sigma_1$. Notice that applications of

SubstL or SubstR can never affect $\sigma'\sigma_1$ or any suffix of $\sigma'\sigma_1$. (II) *Consistency.* Suppose $\gamma\alpha = \lambda\beta$ is a node to which Match rule is applied using a match over (α, β). Then either $\sigma'\sigma_1$ is a suffix of both α and β, or $\alpha = \beta = \sigma''\sigma_1$ for some σ'' satisfying the following: (i) σ'' is a proper suffix of σ'; (ii) $\gamma = UZ$ and $\lambda = Z$ such that $Z\sigma''$ is a suffix of σ'; and (iii) the match is over $\sigma''\sigma_1$. The locality and consistency conditions basically say that choices made in the construction of the local tableau should not contradict to the fact that $\sigma'\sigma_1$ is completely irredundant.

The construction of a path in a tableau ends with a leaf. A *successful leaf* is either a node labeled by $\varsigma = \varsigma$ for some ς, or a node labeled by $\epsilon = V$ ($V = \epsilon$) with $V \simeq \epsilon$, or a node that has the same label as one of its ancestors. An *unsuccessful leaf* is produced if the node is either labeled by $\epsilon = V$ ($V = \epsilon$) with $V \not\simeq \epsilon$, or labeled by some $\varsigma = \varsigma'$ with distinct ς, ς' such that no rule is applicable to $\varsigma = \varsigma'$. A local tableau has additionally two new kind of successful/unsuccessful leaves: (i) An L-root is a successful leaf if it shares the same label with one of its ancestors that is also an L-root. (ii) Suppose $\alpha'\sigma'\sigma_0\sigma_1 = \beta'\sigma'\sigma_0\sigma_1$ is an L-node and its child $\alpha'\sigma'\sigma_1 = \beta'\sigma'\sigma_1$ is an L-root. In the local tableau rooted at $\alpha'\sigma'\sigma_1 = \beta'\sigma'\sigma_1$, a node of the form $Z\sigma_1 = \sigma_1$ is deemed as a leaf. It is a successful leaf if Z is in the R-set of the L-root; it is an unsuccessful leaf otherwise.

Tableau constructions always terminate. In fact we have the following.

Lemma 7. *The size of every tableau for an equality is effectively bounded. The number of tableaux for an equality is effectively bounded.*

A tableau is *successful* if all of its leaves are successful. Successful tableaux generate bisimulation bases.

Proposition 2. *Suppose $X\alpha, Y\beta$ are nBPA processes. Then $X\alpha \simeq Y\beta$ if and only if there is a successful tableau for $X\alpha = Y\beta$.*

Proof. If $X\alpha \simeq Y\beta$ we can easily construct a tableau using the bisimulation property, Corollary 2 and Corollary 3. Conversely suppose there is a successful tableau $\mathfrak{T}$ for $X\alpha = Y\beta$. Let $\mathcal{A} = \mathcal{A}_b \cup \mathcal{A}_z \cup \mathcal{A}_l$. The set $\mathcal{A}_b$ of basic axioms is given by $\{\gamma = \lambda \mid \gamma = \lambda \text{ is a label of a node in } \mathfrak{T}\}$. The set $\mathcal{A}_z$ is defined by

$$\mathcal{A}_z = \left\{ V\sigma = \theta\sigma, \theta\sigma = \sigma \;\middle|\; \begin{array}{l} V\sigma = \sigma \text{ is in } \mathcal{A}_b, \text{ and } V \stackrel{\tau}{\Longrightarrow} \theta \stackrel{\tau}{\Longrightarrow} \epsilon \\ \text{is a chosen shortest path from } V \text{ to } \epsilon. \end{array} \right\}.$$

Suppose $\gamma\sigma'\sigma_1 = \lambda\sigma'\sigma_1$ is an L-root and $\gamma\sigma'\sigma_0\sigma_1 = \lambda\sigma'\sigma_0\sigma_1$ is its parent. A node $\eta\sigma'\sigma_1 = \kappa\sigma'\sigma_1$ in the local tableau rooted at $\gamma\sigma'\sigma_1 = \lambda\sigma'\sigma_1$ must be lifted to $\eta\sigma'\sigma_0\sigma_1 = \kappa\sigma'\sigma_0\sigma_1$ in order to show that $\gamma\sigma'\sigma_0\sigma_1 = \lambda\sigma'\sigma_0\sigma_1$ satisfies the bisimulation base property. Since local tableaux may be nested, the node might have several lifted versions. The set $\mathcal{A}_l$ is defined to be the collection of all such lifted pairs. We can prove by induction on the nodes of the tableau, starting with the leaves, that $\mathcal{A}$ is a bisimulation base. Hence $X\alpha \simeq Y\beta$ by Lemma 2. $\square$

Our main result follows from Proposition 1, Lemma 7 and Proposition 2.

Theorem 1. *The branching bisimilarity on nBPA processes is decidable.*

5 Regularity Checking

Regularity problem asks if a process is bisimilar to a finite state process. For strong regularity problem of nBPA, Kučera [Kuč96] showed that it is decidable in polynomial time. Srba [Srb02] observed that it is actually NL-complete. The decidability of strong regularity problem for the general BPA was proved by Burkart, Caucal and Steffen [BCS95,BCS96]. It was shown to be PSPACE-hard by Srba [Srb02]. The decidability of almost all weak regularity problems of process rewriting systems [May00] are unknown. The only exception is Jancar and Esparza's undecidability result of weak regularity problem of Petri Net and its extension [JE96]. Srba [Srb03] proved that weak regularity is both NP-hard and co-NP-hard for nBPA. Using a result by Srba [Srb03], Mayr proved that weak regularity problem of nBPA is EXPTIME-hard [May03].

The present paper improves our understanding of the issue by the following.

Theorem 2. *The regularity problem of $\simeq$ on nBPA is decidable.*

Proof. One proves by a combinatorial argument that, in the transition tree of an infinite state BPA process, (i) a path $V_0\sigma_0 \xrightarrow{\ell_1^*} V_1\sigma_1 \xrightarrow{\ell_2^*} V_2\sigma_2 \ldots \xrightarrow{\ell_m^*} V_m\sigma_m$ exists such that (ii) $|\sigma_0| < |\sigma_1| < |\sigma_2| < \ldots < |\sigma_m|$ and (iii) $\|V_0\sigma_0\|_b < \|V_1\sigma_1\|_b < \|V_2\sigma_2\|_b < \ldots < \|V_m\sigma_m\|_b$. We can choose m large enough such that $0 \leq i < j \leq m$ for some i, j satisfying $V_i = V_j$ and $\mathcal{R}_{\sigma_i} = \mathcal{R}_{\sigma_j}$. Let $\sigma_j = \sigma\sigma_i$ for some σ. Clearly $\|\sigma_i\|_b < \|\sigma_j\|_b$. Using Corollary 2 one can prove by induction that $\sigma^i\sigma_i \not\simeq \sigma^j\sigma_i$ whenever $i \neq j$. It is semi-decidable to find (i) with properties (ii,iii). The converse implication is proved by a tree construction using Theorem 1. $\Box$

6 Remark

For parallel processes (BPP/PN) with silent actions, the only known decidability result on equivalence checking is due to Czerwiński, Hofman and Lasota [CHL11]. This paper provides the analogous decidability result for the sequential processes (BPA/PDA) with silent actions. For further research one could try to apply the technique developed in this paper to general BPA and normed PDA.

Acknowledgement. I am indebted to He, Huang, Long, Shen, Tao, Yang, Yin and the anonymous referees. The support from NSFC (60873034, 61033002, ANR 61261130589) and STCSM (11XD1402800) is gratefully acknowledged.

References

[BBK87] J. Baeten, J. Bergstra, and J. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. In *PARLE'87*, pages 94–113. Lecture Notes in Computer Science 259, 1987.

[BCS95] O. Burkart, D. Caucal, and B. Steffen. An elementary bisimulation decision procedure for arbitrary context free processes. In *MFCS'95*, pages 423–433. Lecture Notes in Computer Science 969, Springer, 1995.

[BCS96] O. Burkart, D. Caucal, and B. Steffen. Bisimulation collapse and the process taxonomy. In *CONCUR'96*, pages 247–262. Lecture Notes in Computer Science 1119, Springer, 1996.

[BGS92] J. Balcazar, J. Gabarro, and M. Santha. Deciding bisimilarity is p-complete. *Formal Aspects of Computing*, 4:638–648, 1992.

[CHL11] W. Czerwiński, P. Hofman, and S. Lasota. Decidability of branching bisimulation on normed commutative context-free processes. In *CONCUR'11*, pages 528–542. Lecture Notes in Computer Science 6901, Springer, 2011.

[CHS92] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. In *CONCUR'92*, pages 138–147. Lecture Notes in Computer Science 630, Springer, 1992.

[CHT95] D. Caucal, D. Huynh, and L. Tian. Deciding branching bisimilarity of normed context-free processes is in σ_2^p. *Information and Computation*, 118:306–315, 1995.

[Hir96] Y. Hirshfeld. Bisimulation trees and the decidability of weak bisimulations. *Electronic Notes in Theoretical Computer Science*, 5:2–13, 1996.

[HJM96] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context free processes. *Theoretical Computer Science*, 158(1-2):143–159, 1996.

[HS91] H. Hüttel and C. Stirling. Actions speak louder than words: Proving bisimilarity for context-free processes. In *LICS'91*, pages 376–386, 1991.

[HT94] T. Huynh and L. Tian. Deciding bisimilarity of normed context free processes is in σ_2^p. *Theoretical Computer Science*, 123:83–197, 1994.

[Hüt92] H. Hüttel. Silence is golden: Branching bisimilarity is decidable for context free processes. In *CAV'91*, pages 2–12. Lecture Notes in Computer Science 575, Springer, 1992.

[Jan12] P. Jančar. Bisimilarity on basic process algebra is in 2-exptime. 2012.

[JE96] P. Jančar and J. Esparza. Deciding finiteness of petri nets up to bisimulation. In *ICALP'96*, pages 478–489. Lecture Notes in Computer Science 1099, Springer, 1996.

[Kie13] S. Kiefer. Bpa bisimilarity is exptime-hard. *Information Processing Letters*, 113:101–106, 2013.

[Kuč96] A. Kučera. Regularity is decidable for normed bpa and normed bpp processes in polynomial time. In *SOFSEM'96*, pages 377–384. Lecture Notes in Computer Science 1175, Springer, 1996.

[May00] R. Mayr. Process rewrite systems. *Information and Computation*, 156:264–286, 2000.

[May03] R. Mayr. Weak bisimilarity and regularity of bpa is exptime-hard. In *EXPRESS'03*, 2003.

[Srb02] J. Srba. Strong bisimilarity and regularity of basic process algebra is pspace-hard. In *ICALP'02*, pages 716–727. Lecture Notes in Computer Science 2380, Springer, 2002.

[Srb03] J. Srba. Complexity of weak bisimilarity and regularity for BPA and BPP. *Mathematical Structures in Computer Science*, 13:567–587, 2003.

[Stř98] J. Stříbrná. Hardness results for weak bisimilarity of simple process algebras. *Electronic Notes in Theoretical Computer Science*, 18:179–190, 1998.

[vGW89] R. van Glabbeek and W. Weijland. Branching time and abstraction in bisimulation semantics. In *Information Processing'89*, pages 613–618. North-Holland, 1989.