Logical Methods in Computer Science
Vol. 13(4:17)2017, pp. 1–16
www.lmcs-online.org

Submitted Feb. 22, 2016
Published Nov. 28, 2017

BRANCHING BISIMILARITY OF NORMED BPA PROCESSES AS A RATIONAL MONOID

PETR JANČAR

Technical University Ostrava (FEI VŠB-TUO), Czech Rep.
e-mail address: petr.jancar@vsb.cz

Abstract. The paper presents an elaborated and simplified version of the structural result for branching bisimilarity on normed BPA (Basic Process Algebra) processes that was the crux of a conference paper by Czerwiński and Jančar (arxiv 7/2014 and LiCS 2015). That paper focused on the computational complexity, and a NEXPTIME-upper bound has been derived; the authors built on the ideas by Fu (ICALP 2013), and strengthened his decidability result. Later He and Huang announced the EXPTIME-completeness of this problem (arxiv 1/2015, and LiCS 2015), giving a technical proof for the EXPTIME membership. He and Huang indirectly acknowledge the decomposition ideas by Czerwiński and Jančar on which they also built, but it is difficult to separate their starting point from their new ideas.

One aim here is to present the previous decomposition result of Czerwiński and Jančar in a technically new framework, noting that branching bisimulation equivalence on normed BPA processes corresponds to a rational monoid (in the sense of [Sakarovitch, 1987]); in particular it is shown that the mentioned equivalence can be decided by normal-form computing deterministic finite transducers. Another aim is to provide a complete description, including an informal overview, that should also make clear how Fu's ideas were used, and to give all proofs in a form that should be readable and easily verifiable.

1. Introduction

Bisimulation equivalence (or bisimilarity) is a fundamental notion in theory of processes, and the respective decidability and complexity questions are a natural research topic; we can refer to [21] for an (updated) overview of the results in a specific area of process rewrite systems.

One basic model of infinite-state systems is called Basic Process Algebra (BPA), which can be naturally related to context-free grammars in Greibach normal form. Here the *processes* are identified with finite sequences of variables (nonterminals); a process $A\alpha$ can change by performing an *action*, denoted by $A\alpha \xrightarrow{a} \beta\alpha$, in which case its leftmost variable A is rewritten according to a grammar rule $A \longrightarrow a\beta$ (presented rather as $A \xrightarrow{a} \beta$ in our context).

Key words and phrases: branching bisimulation equivalence, basic process algebra.

The work was supported by the Grant Agency of the Czech Rep., project GAČR:15-13784S.

DOI:10.23638/LMCS-13(4:17)2017

A seminal paper by Baeten, Bergstra and Klop [1] showed the decidability of bisimilarity on nBPA, i.e. on the *normed* BPA processes, where each variable can be stepwise rewritten to the empty word; this decidability result was later extended to the whole class BPA [5]. Further exploration placed the problem on nBPA even in PTime [12] (this problem is thus PTime-complete [2]); we can refer to [6] for further references and a way towards the so far best known upper bound. The bisimilarity problem for the whole class BPA is known to be ExpTime-hard [17] and to belong to 2-ExpTime (claimed in [3] and explicitly proven in [15]).

When also internal (unobservable) actions of systems are taken into account, the most studied generalization of bisimilarity is *weak bisimilarity* [19] but the relevance of the finer equivalence called *branching bisimilarity* is also well argued [22].

The (un)decidability status of weak bisimilarity on BPA, as well as on nBPA, is still open, but we have the ExpTime-hardness result by Mayr [18] for weak bisimilarity on nBPA. Similarly, the decidability status of weak bisimilarity is still open in the case of (normed) Basic Parallel Processes, which is the parallel (or commutative) version of BPA.

The situation seems more favourable in the case of branching bisimilarity. It was first shown decidable for the normed Basic Parallel Processes [7], and then Fu [9] showed the decidability on nBPA. A later paper [23] shows that the mentioned decidability results for branching bisimilarity cannot be essentially extended, possibly with the exception of the full classes of BPA processes and of Basic Parallel Processes for which the decidability question remains open.

The case of *branching bisimilarity on nBPA* is the main topic of this paper. We first note that Fu's decidability result [9] is substantially stronger than the previous results dealing with so called *totally* normed BPA [14, 4] (where no variable can "disappear" by unobservable actions). In the case of totally normed BPA processes even a polynomial time algorithm is suggested in [10], building on the unique-decomposition results and techniques that were previously used in the case of (strong) bisimilarity on nBPA.

A crucial novel idea in Fu's decidability proof is a use of the notion that can be called the *class-change norm* (called the branching norm in [9]); while the standard norm counts all steps in rewriting a process to the empty word, the class-change norm only counts the steps that change the current equivalence-class. It is not clear how to compute this norm directly but equivalent processes $\alpha \sim \beta$ must have the same class-change norm. Another useful fact shown by Fu is that the relation of $\alpha\gamma$ and $\beta\gamma$ (either $\alpha\gamma \sim \beta\gamma$ or $\alpha\gamma \not\sim \beta\gamma$) is determined solely by the *redundant variables* w.r.t. γ, i.e. by those X for which $X\gamma \sim \gamma$, independently of the string γ itself.

This paper is based on the research reported on in [8], performed with W. Czerwiński (see Author's acknowledgements). The main new idea there was to use the *decompositions* of processes that are *relative* to a given set of (redundant) variables; the notion is also based on the (semantic) class-change norm. This structural result is here a bit reworked and presented in a technically new framework; it is shown that the quotient of branching bisimulation equivalence on nBPA is a rational monoid (in the sense of Sakarovitch [20]). In particular, for a given nBPA system $\mathcal{G}$ branching bisimilarity can be decided by a (canonical) normal-form computing deterministic finite-state transducer $\mathcal{T}^{\mathcal{G}}$; to each process α it computes the normal form $\mathcal{T}^{\mathcal{G}}(\alpha)$, which is a unique process in the equivalence-class $[\alpha]_\sim$, and we have $\alpha \sim \beta$ iff $\mathcal{T}^{\mathcal{G}}(\alpha) = \mathcal{T}^{\mathcal{G}}(\beta)$. The size of $\mathcal{T}^{\mathcal{G}}$ can be easily bounded by an exponential function of the size of $\mathcal{G}$.

We will not show a direct construction of $\mathcal{T}^{\mathcal{G}}$, but we will show a quickly verifiable consistency condition for any given transducer $\mathcal{T}$ that guarantees $\mathcal{T}(\alpha) = \mathcal{T}(\beta) \Rightarrow \alpha \sim \beta$; moreover, $\mathcal{T}^{\mathcal{G}}$ will be shown to satisfy this consistency condition w.r.t. $\mathcal{G}$. This immediately yields a nondeterministic exponential-time algorithm deciding branching bisimilarity on nBPA: given $\mathcal{G}$, α, β, guess a transducer $\mathcal{T}$ of at most exponential size (in the size of $\mathcal{G}$), check that $\mathcal{T}$ is consistent with $\mathcal{G}$, and verify that $\mathcal{T}(\alpha) = \mathcal{T}(\beta)$.

The problem for which Fu [9] showed the decidability (by an involved proof in a tableau framework) is thus placed in NExpTime. Regarding the question of a lower bound, Fu [9] noted that the problem is ExpTime-hard, which was later confirmed by Huang and Yin [13]. (More details about this interesting point are given in Section 3.) In Section 5 we also add some remarks on a possible construction of the canonical transducer $\mathcal{T}^{\mathcal{G}}$ in deterministic exponential time; this seems to be (at least implicitly) related to the paper by He and Huang [11] that announced ExpTime-completeness.

Structure of the paper. In Section 2 we define the used notions and make some simple observations. Section 3 gives an informal overview, which is then formalized in Section 4. Section 3 also contains a remark on the lower complexity bound, and Section 5 adds some further remarks.

2. Preliminaries

We put $\mathbb{N} = \{0, 1, 2, \dots\}$, and $[i, j] = \{i, i+1, \dots, j\}$ for $i, j \in \mathbb{N}$.

For a set M, by M^* we denote the set of finite sequences of elements of M, also called *words*, or *strings*, *over* M; by ε we denote the empty string. For $\alpha \in M^*$, by $|\alpha|$ we denote its length.

Labelled transition systems. A *labelled transition system*, an *LTS* for short, is a tuple

$$\mathcal{L} = (\mathcal{S}, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$$

where $\mathcal{S}$ is the set of *states*, $\mathcal{A}$ is the set of *actions* and $\xrightarrow{a} \subseteq \mathcal{S} \times \mathcal{S}$ is the set of *a-labelled transitions*. We reserve the symbol

τ for the (unique) *silent action*; the *visible actions* are the elements of $\mathcal{A} \smallsetminus \{\tau\}$.

We write $s \xrightarrow{a} t$ rather than $(s, t) \in \xrightarrow{a}$ (for $a \in \mathcal{A}$), and we define $s \xrightarrow{w} t$ for $w \in \mathcal{A}^*$ inductively: $s \xrightarrow{\varepsilon} s$; if $s \xrightarrow{a} s'$ and $s' \xrightarrow{u} t$, then $s \xrightarrow{au} t$. By $s \xrightarrow{w} t$ we sometimes also refer to a concrete respective path from s to t in $\mathcal{L}$. (We do not exclude cycles in the paths.)

Branching bisimilarity, i.e., branching bisimulation equivalence $\sim$. Given an LTS $\mathcal{L} = (\mathcal{S}, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$, a relation $\mathcal{B} \subseteq \mathcal{S} \times \mathcal{S}$ is a *branching bisimulation* in $\mathcal{L}$ if for any $(s, t) \in \mathcal{B}$ the following two conditions hold:

i) for any $a \in \mathcal{A}$, any *move* $s \xrightarrow{a} s'$ *can be matched from* t, i.e.,
   a) $a = \tau$ and $(s', t) \in \mathcal{B}$, or
   b) there is a path $t = t_0 \xrightarrow{\tau} t_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} t_k \xrightarrow{a} t'$ (for some $k \geq 0$) such that $(s', t') \in \mathcal{B}$ and $(s, t_i) \in \mathcal{B}$ for all $i \in [1, k]$;

ii) for any $a \in \mathcal{A}$, any *move* $t \xrightarrow{a} t'$ *can be matched from* s, i.e.,
   a) $a = \tau$ and $(s, t') \in \mathcal{B}$, or

b) there is a path $s = s_0 \xrightarrow{\tau} s_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} s_k \xrightarrow{a} s'$ (for some $k \geq 0$) such that $(s', t') \in \mathcal{B}$ and $(s_i, t) \in \mathcal{B}$ for all $i \in [1, k]$.

By $s \sim t$, to be read as "states s, t are *branching bisimilar*", we denote that there is a branching bisimulation containing (s, t). We can easily verify the standard facts that $\sim \subseteq \mathcal{S} \times \mathcal{S}$ is the union of all branching bisimulations (in $\mathcal{L}$), and thus the largest branching bisimulation in $\mathcal{L}$, and that $\sim$ is an equivalence relation.

Class-changing transitions, and class-change norm $\langle\!\langle s \rangle\!\rangle$. Assuming an LTS $\mathcal{L} = (\mathcal{S}, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$, we now introduce a few notions and make simple observations that turn out to be very useful for our aims. We say that

a *transition* $s \xrightarrow{a} s'$ is *class-changing* if $s \not\sim s'$.

Hence a class-changing transition leads from one equivalence class of $\sim$ to a different class. We note that $s \sim t$ and $s \xrightarrow{a} s'$ entails that either $a = \tau$ and $s \xrightarrow{a} s'$ is not class-changing (in which case $s' \sim t$), or there is a path $t = t_0 \xrightarrow{\tau} t_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} t_k \xrightarrow{a} t'$ (for some $k \geq 0$) where no transition in the path $t_0 \xrightarrow{\tau} t_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} t_k$ is class-changing (hence $t = t_0 \sim t_1 \sim \cdots \sim t_k$) and $s' \sim t'$; in the latter case, the transition $t_k \xrightarrow{a} t'$ is class-changing iff $s \xrightarrow{a} s'$ is class-changing.

We introduce the class-change norm $\langle\!\langle s \rangle\!\rangle$ as the "class-change distance" of s to the silent states. A *state* s is *silent* if $s \xrightarrow{w} s'$ entails $w \in \{\tau\}^*$. (Hence we can never perform a visible action when starting from a silent state.) Let S_{SIL} be the *set of silent states* (in the assumed LTS $\mathcal{L} = (S, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$); S_{SIL} is obviously a (maybe empty) equivalence class of $\sim$ (since the set $\{(s,t) \mid s, t \in S_{\text{SIL}}\}$ is a branching bisimulation, and $s \in S_{\text{SIL}}$, $t \notin S_{\text{SIL}}$ implies $s \not\sim t$).

By $\langle\!\langle s \rangle\!\rangle$ we denote the *class-change norm* of s, the *cc-norm* for short, which is the least $\ell \in \mathbb{N}$ such that there is a path $s = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_k} s_k \in S_{\text{SIL}}$ that has precisely ℓ class-changing transitions; we put $\langle\!\langle s \rangle\!\rangle = \omega$ if S_{SIL} is not reachable from s. Hence $\langle\!\langle s \rangle\!\rangle = 0$ iff $s \in S_{\text{SIL}}$. The previous discussion (of matching $s \xrightarrow{a} s'$ from t when $s \sim t$) easily yields the following fact:

Observation 2.1. If $s \sim t$, then $\langle\!\langle s \rangle\!\rangle = \langle\!\langle t \rangle\!\rangle$.

Remark 2.2. The cc-norm was introduced by Fu in [9], who used the name "branching norm" and a slightly different form; formally his norm also counts the visible transitions no matter if they are class-changing or not but this is no crucial difference, in fact.

BPA systems and processes. We view a *BPA system* (where BPA stands for Basic Process Algebra) as a context-free grammar in Greibach normal form, with no starting variable (nonterminal). We denote it as

$$\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$$

where $\mathcal{V}$ is a finite set of *variables* (or nonterminals), $\mathcal{A}$ is a finite set of *actions* (or terminals), which can contain the *silent action* τ, and $\mathcal{R}$ is a finite set of *rules* of the form $A \xrightarrow{a} \alpha$ where $A \in \mathcal{V}$, $a \in \mathcal{A}$, $\alpha \in \mathcal{V}^*$.

A BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$ has the associated LTS

$$\mathcal{L}_{\mathcal{G}} = (\mathcal{V}^*, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$$

where each rule $A \xrightarrow{a} \alpha$ in $\mathcal{R}$ induces the transitions $A\beta \xrightarrow{a} \alpha\beta$ for all $\beta \in \mathcal{V}^*$. The states of $\mathcal{L}_{\mathcal{G}}$, i.e. the strings of variables, are also called *processes*.

Standard (syntactic) norm $\|\alpha\|$, and normed BPA systems (nBPA). Given a BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$, the *norm* $\|\alpha\|$ of $\alpha \in \mathcal{V}^*$ is the length $|w|$ of a shortest $w \in \mathcal{A}^*$ such that $\alpha \xrightarrow{w} \varepsilon$; we put $\|\alpha\| = \omega$ when there is no such w (where ω stands for an "infinite amount"). We say that α is *normed* if $\|\alpha\| \in \mathbb{N}$ (i.e., if $\alpha \xrightarrow{w} \varepsilon$ for some w). The *BPA system* $\mathcal{G}$ is *normed*, an *nBPA system* for short, if each variable $A \in \mathcal{V}$ is normed.

A *transition* $\alpha \xrightarrow{a} \beta$ is *norm-reducing* if $\|\alpha\| > \|\beta\|$, in which case $\|\beta\| = \|\alpha\| - 1$, in fact. If $\|\alpha\| = \omega$, then there is no norm-reducing transition $\alpha \xrightarrow{a} \beta$. The facts captured by the next proposition are standard; they also entail that we can check in polynomial time whether a BPA system is normed.

Proposition 2.3. *Given a BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$, we have:*
(1) $\|\varepsilon\| = 0$.
(2) $\|\alpha\beta\| = \|\alpha\| + \|\beta\|$ *(assuming $\omega + z = z + \omega = \omega$ for any $z \in \mathbb{N} \cup \{\omega\}$).*
(3) $\|A\| = 1 + \|\alpha\|$ *for a norm-reducing rule $A \xrightarrow{a} \alpha$, if $\|A\| \in \mathbb{N}$.*
(4) *There is a polynomial-time algorithm that computes $\|A\|$ for each $A \in \mathcal{V}$.*
(5) *The finite values $\|A\|$ are at most exponential in the size of $\mathcal{G}$.*

We note in particular that the algorithm in the point (4) can naturally use dynamic programming: We first temporarily assume $\|A\| = \omega$ (the norm is infinite) for all variables; this also temporarily yields $\|\alpha\| = \omega$ for all rhs (right-hand sides) of the rules $A \xrightarrow{a} \alpha$, except of $\alpha = \varepsilon$ since we put $\|\varepsilon\| = 0$ definitively. Now we repeatedly look for a variable A with a temporary norm that has a rule $A \xrightarrow{a} \alpha$ with the least definitive $\|\alpha\| \in \mathbb{N}$; for such A we put $\|A\| = 1 + \|\alpha\|$ definitively (all variables in α have the definitive norms already), and we recompute the temporary norms of the right-hands sides of the rules in $\mathcal{R}$ accordingly. After this repeated process finishes, all temporary cases $\|A\| = \omega$ become also definitive.

Branching bisimilarity problem for nBPA. The *nBPA-bbis problem* asks, given an nBPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$ and two processes $\alpha, \beta \in \mathcal{V}^*$, if $\alpha \sim \beta$, i.e., if α and β are branching bisimilar as the states in $\mathcal{L}_{\mathcal{G}}$.

We add a remark on $\langle\!\langle \alpha \rangle\!\rangle$, which refers to the ("semantic") cc-norm of α in $\mathcal{L}_{\mathcal{G}}$. By Observation 2.1 we know that $\alpha \sim \beta$ implies $\langle\!\langle \alpha \rangle\!\rangle = \langle\!\langle \beta \rangle\!\rangle$. We have shown how to compute the ("syntactic") norm $\|\alpha\|$, but it is unclear how to compute $\langle\!\langle \alpha \rangle\!\rangle$. Nevertheless, since ε is a silent state in $\mathcal{L}_{\mathcal{G}}$, we can easily observe that $\langle\!\langle \alpha \rangle\!\rangle \leq \|\alpha\|$ (and $\langle\!\langle A \rangle\!\rangle$ is thus at most exponential by (5) in Prop. 2.3).

3. Informal overview

Here we sketch some informal ideas that are elaborated in Section 4. We also use a small, but important, example; the example is inspired by a recent work of Huang and Yin [13], which is further discussed below in an additional remark on the lower complexity bound for the nBPA-bbis problem.

Let us consider the following BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$ where

- $\mathcal{V} = \{A, B, C, E, F_B, [F_B, A]\}$ (hence $|\mathcal{V}| = 6$, since $[F_B, A]$ is one symbol),
- $\mathcal{A} = \{\tau\} \cup \{a, b, c, e, f_B^1, f_B^2\}$,

- $\mathcal{R} = \{A \xrightarrow{a} \varepsilon, B \xrightarrow{b} \varepsilon, C \xrightarrow{c} \varepsilon, E \xrightarrow{e} \varepsilon,$
  $F_B \xrightarrow{\tau} \varepsilon, F_B \xrightarrow{f_B^1} \varepsilon,$
  $A \xrightarrow{f_B^1} [F_B, A], B \xrightarrow{f_B^1} B, C \xrightarrow{f_B^1} E,$
  $[F_B, A] \xrightarrow{a} \varepsilon, [F_B, A] \xrightarrow{f_B^1} [F_B, A],$
  $[F_B, A] \xrightarrow{f_B^2} \varepsilon, A \xrightarrow{f_B^2} F_B\}.$

All variables are normed, we even have $\|X\| = 1$ for all $X \in \mathcal{V}$.

In the LTS $\mathcal{L}_\mathcal{G}$ we have, e.g., $F_BAABACCAB \sim AABACCAB$ but $F_BAACABCAB \not\sim AACABCAB$. More generally for any $\alpha \in \{A, B, C\}^*$ we have

$$F_B\alpha \sim \alpha \text{ iff } \alpha = \alpha' B \alpha'' \text{ where } \alpha' \in \{A\}^*. \tag{3.1}$$

We can view F_B as a "claim" that α satisfies "first B", which means that α contains B as the first (i.e., leftmost) occurrence of a symbol from $\{B, C\}$. We leave the verification of (3.1) as an interesting small exercise, since it is not crucial for us.

The example illustrates that a variable can be or not be "redundant" (w.r.t. $\sim$), depending on the "suffix"; formally we say that $R_\gamma = \{X \in \mathcal{V} \mid X\gamma \sim \gamma\}$ is the set of redundant variables w.r.t. γ. In the example, the condition characterizing the strings $\alpha \in \{A, B, C\}^*$ for which $F_B\alpha \sim \alpha$ is regular, i.e. checkable by a finite automaton. It turns out to be an important fact that each nBPA system $\mathcal{G}$ has an associated finite automaton $\mathcal{F}^\mathcal{G}$ that determines the set R_γ after reading γ. Moreover, it turns out possible, and convenient, to let the automaton $\mathcal{F}^\mathcal{G}$ read its input γ from right to left and use the respective sets $R \subseteq \mathcal{V}$ as its control states; the automaton starts in the initial state R_ε (which might be the empty set) and after reading γ (from right to left) it enters the state R_γ. Its transitions are thus of the form $R_{A\gamma} \xleftarrow{A} R_\gamma$, in the notation that visualizes reading from right to left. (In Section 4 we also show the soundness: $R_\gamma = R_\delta$ entails $R_{A\gamma} = R_{A\delta}$.) These ideas were already developed by Fu [9] (though he did not mention the automaton explicitly).

The above example can be generalized to show that the automaton $\mathcal{F}^\mathcal{G}$ can have exponentially many states R_γ (w.r.t. the size of the given nBPA system $\mathcal{G}$): we can add several other pairs $\{B', C'\}$ of variables, with the respective variables $F_{B'}, [F_{B'}, ..]$ and the respective actions and rules. The issue of exponentially many sets of redundant variables is dealt with in [13] in more detail.

Remark 3.1. In [9] there was also a note saying that the nBPA-bbis problem can be shown ExpTime-hard by a slight modification of Mayr's proof for weak bisimilarity [18]. Though this note was repeated in further works, no rigorous proof was given (as pointed out in the first version of this paper [16]). The mentioned "slight modification" has turned out to be not so obvious, but the ExpTime-hardness has been recently rigorously confirmed by Huang and Yin [13].

Mayr's ExpTime-hardness proof [18] uses a reduction from the ALBA problem (Alternating Linear Bounded Automata acceptance), a standard ExpTime-complete problem. Huang and Yin [13] decided to use the Hit-or-Run game for their reduction; this ExpTime-complete problem was used by Kiefer [17] to show the ExpTime-hardness of strong bisimilarity for (general) BPA systems. The ExpTime-hardness of the Hit-or-Run game was also established by a reduction from the ALBA problem. It is worth to note that it is also possible to modify Mayr's reduction [18] by the new idea of [13], to yield ExpTime-hardness of the nBPA-bbis problem by a direct reduction from the ALBA problem; the above example (inspired by [13]) captures the essence since it shows how it is possible to "remember" an ALBA configuration by the current set of redundant variables. (If α is a sequence of ALBA configurations, then R_α determines the leftmost configuration in the sequence; we use a special pair $\{[i,0],[i,1]\}$ of variables (like $\{B,C\}$ in the example), with the respective additional variables, actions and rules, to "remember" if the i-th position is 0 or 1.)

Hence Fu's remark in [9] can be viewed as correct in the end, though it has not been straightforward to come with the appropriate "slight" modification.

The contribution of this paper captures the decomposition ideas from [8]. The above discussed automaton $\mathcal{F}^{\mathcal{G}}$, satisfying $R_\alpha \xleftarrow{\alpha} R_\varepsilon$, can be enhanced to become a finite-state transducer $\mathcal{T}^{\mathcal{G}}$ (corresponding to a given nBPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$) that translates its input α into a string $\mathcal{T}^{\mathcal{G}}(\alpha)$, processing α from right to left; this is denoted $R_\alpha \xleftarrow[\beta]{\alpha} R_\varepsilon$ where $\beta = \mathcal{T}^{\mathcal{G}}(\alpha)$. For the uniqueness of the "canonical transducer" $\mathcal{T}^{\mathcal{G}}$ we use a linear order on $\mathcal{V}$ and take $\mathcal{T}^{\mathcal{G}}(\alpha)$ as the lexicographically smallest string among the longest redundancy-free strings from the equivalence class $[\alpha]_\sim$. (Here the lexicographic order of two different strings is determined by the first position *from the right* where the strings differ.) By the redundancy-freeness of a string β we mean that $\beta = \beta' A \beta''$ entails that $A\beta'' \not\sim \beta''$ (i.e., $A \notin R_{\beta''}$). We recall that $\beta \in [\alpha]_\sim$ entails $\langle\!\langle \beta \rangle\!\rangle = \langle\!\langle \alpha \rangle\!\rangle$ (by Observation 2.1), and we can observe that $|\beta| \leq \langle\!\langle \beta \rangle\!\rangle$ when β is redundancy-free (since any path $A\beta'' \xrightarrow{u} \beta''$ contains at least one class-changing transition when $A\beta'' \not\sim \beta''$).

We will verify the soundness of the above definition of the canonical transducer $\mathcal{T}^{\mathcal{G}}$ (for any *normed* BPA system $\mathcal{G}$). We thus get

$$\alpha \sim \beta \text{ (in } \mathcal{L}_{\mathcal{G}}) \text{ iff } \mathcal{T}^{\mathcal{G}}(\alpha) = \mathcal{T}^{\mathcal{G}}(\beta).$$

We also note the idempotency $\mathcal{T}^{\mathcal{G}}(\mathcal{T}^{\mathcal{G}}(\alpha)) = \mathcal{T}^{\mathcal{G}}(\alpha)$, and the fact that $\mathcal{T}^{\mathcal{G}}(\alpha)$ can be naturally viewed as the *normal form* (or the *prime decomposition*) of α; two strings α, β are equivalent (meaning branching bisimilar) iff they have the same normal forms (the same prime decompositions). We also note that generally we do not have $\mathcal{T}^{\mathcal{G}}(\alpha\gamma) = \mathcal{T}^{\mathcal{G}}(\alpha)\,\mathcal{T}^{\mathcal{G}}(\gamma)$, since the decomposition is more subtle: we have $\mathcal{T}^{\mathcal{G}}(\alpha\gamma) = \mathcal{T}^{\mathcal{G}}_{R_\gamma}(\alpha)\,\mathcal{T}^{\mathcal{G}}_{R_\varepsilon}(\gamma)$, where $\mathcal{T}^{\mathcal{G}}_R(\alpha)$ is the translation of α when the transducer starts from R instead of the initial state R_ε.

We will not show a direct construction of $\mathcal{T}^{\mathcal{G}}$, when given an nBPA system $\mathcal{G}$, but we will show a quickly verifiable "consistency" condition for any given transducer $\mathcal{T}$ that guarantees $\mathcal{T}(\alpha) = \mathcal{T}(\beta) \Rightarrow \alpha \sim \beta$; moreover, $\mathcal{T}^{\mathcal{G}}$ will be shown to satisfy this consistency condition w.r.t. $\mathcal{G}$.

The size of the canonical transducer $\mathcal{T}^{\mathcal{G}}$ is at most exponential in the size of $\mathcal{G}$ (since $|\mathcal{T}^{\mathcal{G}}_R(A)| \leq \|A\|$, as we will show easily), and we thus have a conceptually simple *nondeterministic exponential-time algorithm deciding the nBPA-bbis problem*:

> Given a normed BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$ and $\alpha, \beta \in \mathcal{V}^*$, guess a transducer $\mathcal{T}$ of at most exponential size (w.r.t. $\mathcal{G}$), check that $\mathcal{T}$ is consistent with $\mathcal{G}$, and verify that $\mathcal{T}(\alpha) = \mathcal{T}(\beta)$.

In Section 5 we add further remarks on the construction of $\mathcal{T}^{\mathcal{G}}$ and on the complexity of the nBPA-bbis problem.

4. Branching bisimilarity on nBPA via finite transducers

4.1. **Normal-form-computing transducers.** By a *transducer* we mean a tuple $\mathcal{T} = (Q, \mathcal{V}, \Delta, q_0)$ where Q is a finite set of (control) states, $\mathcal{V}$ is a finite (input and output) alphabet, Δ is a (transition and translation) function of the type $Q \times \mathcal{V} \longrightarrow Q \times \mathcal{V}^*$, and $q_0 \in Q$ is the initial state.

We view transducers as reading (and writing) *from right to left*; we write $q' \xleftarrow[\gamma]{A} q$ instead of $\Delta(q, A) = (q', \gamma)$ to visualize this fact. The function Δ is naturally extended to the type $Q \times \mathcal{V}^* \longrightarrow Q \times \mathcal{V}^*$ by the following inductive definition, which uses the "visual" notation:

- $q \xleftarrow[\varepsilon]{\varepsilon} q$ (for each $q \in Q$),
- if $q' \xleftarrow[\gamma]{A} q$ and $q'' \xleftarrow[\beta]{\alpha} q'$, then $q'' \xleftarrow[\beta\gamma]{\alpha A} q$.

By $\mathcal{T}_q(\alpha)$ we denote the translation of $\alpha \in \mathcal{V}^*$ when starting in $q \in Q$, i.e., the string β such that $q' \xleftarrow[\beta]{\alpha} q$ (for some q'); we also use the notation $\mathcal{T}(\alpha)$ for $\mathcal{T}_{q_0}(\alpha)$. For each $q \in Q$ we define the equivalence relation $\equiv^{\mathcal{T}}_q$ on $\mathcal{V}^*$ as follows:

$$\alpha \equiv^{\mathcal{T}}_q \beta \Leftrightarrow_{\mathrm{df}} \mathcal{T}_q(\alpha) = \mathcal{T}_q(\beta); \text{ we put } \equiv^{\mathcal{T}} = \equiv^{\mathcal{T}}_{q_0}.$$

We say that $A \in \mathcal{V}$ is a *q-prime* if $\mathcal{T}_q(A) = A$, hence if $q' \xleftarrow[A]{A} q$ for some q'. A string $\alpha \in \mathcal{V}^*$ is a *q-normal form* if $\alpha = \varepsilon$ or $\alpha = A_k A_{k-1} \cdots A_1$ for $k \geq 1$ where

$$q_k \xleftarrow[A_k]{A_k} q_{k-1} \xleftarrow[A_{k-1}]{A_{k-1}} q_{k-2} \cdots \xleftarrow[A_3]{A_3} q_2 \xleftarrow[A_2]{A_2} q_1 \xleftarrow[A_1]{A_1} q \text{ for some } q_1, q_2, \dots, q_k.$$

By $\mathrm{NF}^{\mathcal{T}}_q$ we denote the set of q-normal forms; hence $\varepsilon \in \mathrm{NF}^{\mathcal{T}}_q$, and $\beta A \in \mathrm{NF}^{\mathcal{T}}_q$ iff A is a q-prime and β is a q'-normal form for q' satisfying $q' \xleftarrow[A]{A} q$. We note that $\alpha \in \mathrm{NF}^{\mathcal{T}}_q$ entails $\mathcal{T}_q(\alpha) = \alpha$.

A transducer $\mathcal{T} = (Q, \mathcal{V}, \Delta, q_0)$ is a *normal-form-computing transducer*, an *nfc-transducer* for short, if $\mathcal{T}_q(A) \in \mathrm{NF}^{\mathcal{T}}_q$ for all $q \in Q$, $A \in \mathcal{V}$, and the "target states" are the same for both A and $\mathcal{T}_q(A)$, i.e.

$$q' \xleftarrow[\gamma]{A} q \text{ implies } q' \xleftarrow[\gamma]{\gamma} q \text{ (where } \gamma = \mathcal{T}_q(A)). \tag{4.1}$$

For nfc-transducers we thus have $\mathcal{T}_q(\mathcal{T}_q(\alpha)) = \mathcal{T}_q(\alpha)$ (idempotency), which also entails that $\alpha \equiv^{\mathcal{T}}_q \mathcal{T}_q(\alpha)$; moreover, the condition (4.1) also entails that $\mathcal{T}_q(\alpha\beta) = \mathcal{T}_q(\alpha\, \mathcal{T}_q(\beta))$.

We note that checking if a given transducer $\mathcal{T}$ is an nfc-transducer is straightforward.

4.2. **Nfc-transducers consistent with a BPA system.** In Section 4.3 we will define a canonical nfc-transducer $\mathcal{T}^{\mathcal{G}}$ for a *normed* BPA system $\mathcal{G}$; it will turn out that the branching bisimilarity $\sim$ in $\mathcal{L}_{\mathcal{G}}$ coincides with the equivalence $\equiv^{\mathcal{T}^{\mathcal{G}}}$.

Here we assume a fixed (general) BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$ and a fixed nfc-transducer $\mathcal{T} = (Q, \mathcal{V}, \Delta, q_0)$; we aim to find a suitable condition guaranteeing that the equivalence $\equiv^{\mathcal{T}}$ (on the set $\mathcal{V}^*$) is a branching bisimulation in the LTS $\mathcal{L}_{\mathcal{G}} = (\mathcal{V}^*, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$.

A natural idea is to require that for every action $a \in \mathcal{A}$ (including the case $a = \tau$) the processes α and $\mathcal{T}(\alpha)$ yield the same normal forms of the results of "long moves"

$\xrightarrow{\tau}\xrightarrow{\tau}\cdots\xrightarrow{\tau}\xrightarrow{a}$ where the (maybe empty) τ-prefix is bound to go inside the equivalence class $[\alpha]_{\equiv^{\mathcal{T}}}$ (which is the same as $[\mathcal{T}(\alpha)]_{\equiv^{\mathcal{T}}}$), and the final $\xrightarrow{a}$-step might be missing when $a=\tau$. We formalize this idea by Def. 4.1, after we introduce the "long moves" $\overset{a}{\rightsquigarrow}_q$, relativized w.r.t. the states $q \in Q$.

For our fixed $\mathcal{G}$ and $\mathcal{T}$ we write $\alpha \overset{a}{\rightsquigarrow}_q \beta$, where $\alpha, \beta \in \mathcal{V}^*$, $q \in Q$, and $a \in \mathcal{A}$, if

- either $a = \tau$ and $\beta = \mathcal{T}_q(\alpha)$,
- or there are $\alpha_1, \alpha_2, \ldots, \alpha_k$ (for some $k \geq 0$) and β' such that

$$\begin{array}{c} \alpha = \alpha_0 \xrightarrow{\tau} \alpha_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} \alpha_k \xrightarrow{a} \beta' \qquad \text{in } \mathcal{L}_{\mathcal{G}}, \\ \mathcal{T}_q(\alpha_0) = \mathcal{T}_q(\alpha_1) = \cdots = \mathcal{T}_q(\alpha_k) \qquad \text{and} \qquad \mathcal{T}_q(\beta') = \beta \ . \end{array}$$

Hence $\alpha \overset{a}{\rightsquigarrow}_q \beta$ entails that $\beta \in \mathrm{NF}_q^{\mathcal{T}}$ (β is a q-normal form). In particular we have $\varepsilon \overset{\tau}{\rightsquigarrow}_q \varepsilon$. We define the equivalences $\approx_q$ as follows:

$$\alpha_1 \approx_q \alpha_2 \ \Leftrightarrow_{\mathrm{df}}\ \forall a \in \mathcal{A} : \{\beta \mid \alpha_1 \overset{a}{\rightsquigarrow}_q \beta\} = \{\beta \mid \alpha_2 \overset{a}{\rightsquigarrow}_q \beta\}.$$

Now the announced definition follows; it also uses the fact that $q' \xleftarrow[\varepsilon]{A} q$ implies $q' = q$ for nfc-transducers (by the condition (4.1)).

Definition 4.1. An *nfc-transducer* $\mathcal{T} = (Q, \mathcal{V}, \Delta, q_0)$ is *consistent with a BPA system* $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$ if the following three conditions hold.

(1) $A \approx_{q_0} \varepsilon$ if $\mathcal{T}_{q_0}(A) = \varepsilon$ (i.e., if $q_0 \xleftarrow[\varepsilon]{A} q_0$);

(2) $A \approx_q \mathcal{T}_q(A)$ if $\mathcal{T}_q(A) \neq \varepsilon$ (hence $q' \xleftarrow[\beta]{A} q$ where $\beta \neq \varepsilon$ entails $A \approx_q \beta$);

(3) $AC \approx_q C$ if $\mathcal{T}_q(AC) = \mathcal{T}_q(C) = C$ (i.e., if $q' \xleftarrow[\varepsilon]{A} q' \xleftarrow[C]{C} q$ for some q').

Lemma 4.2.

(1) *There is a polynomial algorithm checking if a given nfc-transducer $\mathcal{T}$ is consistent with a given BPA system $\mathcal{G}$.*

(2) *If an nfc-transducer $\mathcal{T}$ is consistent with a BPA system $\mathcal{G}$, then $\equiv^{\mathcal{T}}$ is a branching bisimulation in $\mathcal{L}_{\mathcal{G}}$.*

Proof.

1. We assume an nfc-transducer $\mathcal{T} = (Q, \mathcal{V}, \Delta, q_0)$ and a BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$. For any $q \in Q$, we put $\mathcal{E}_q = \{X \in \mathcal{V} \mid \mathcal{T}_q(X) = \varepsilon\}$, and we define the set $\bar{\mathcal{E}}_q \subseteq \mathcal{V}$ (of *silently q-erasable variables*) inductively:

$$X \in \bar{\mathcal{E}}_q \text{ if } X \in \mathcal{E}_q \text{ and there is a rule } X \xrightarrow{\tau} \gamma \text{ in } \mathcal{R} \text{ where } \gamma \in (\bar{\mathcal{E}}_q)^*.$$

Using dynamic programming, the sets $\bar{\mathcal{E}}_q$ are quickly constructible for all $q \in Q$. (In the first step we find $X \in \mathcal{E}_q$ for which $X \xrightarrow{\tau} \varepsilon$ is a rule in $\mathcal{R}$; if there are no such X, then $\bar{\mathcal{E}}_q = \emptyset$.)

It is easy to verify that the following "axioms and deduction rules" i) – v) characterize when we have $\alpha \overset{a}{\rightsquigarrow}_q \beta$. (We omit $\mathcal{T}(\alpha)$ in the notation $q' \xleftarrow[\mathcal{T}(\alpha)]{\alpha} q$ when unimportant.)

i) $\alpha \overset{\tau}{\rightsquigarrow}_q \mathcal{T}_q(\alpha)$ (for all $\alpha \in \mathcal{V}^*$ and $q \in Q$);
ii) if $A \xrightarrow{a} \delta$ is a rule in $\mathcal{R}$, then $A \overset{a}{\rightsquigarrow}_q \mathcal{T}_q(\delta)$;
iii) if $A \xrightarrow{\tau} \delta$ is a rule in $\mathcal{R}$, $\mathcal{T}_q(A) = \mathcal{T}_q(\delta)$, and $\delta \overset{a}{\rightsquigarrow}_q \beta$, then $A \overset{a}{\rightsquigarrow}_q \beta$;

iv) if $q'' \xleftarrow{A} q' \xleftarrow[\delta]{\gamma} q$ and $A \overset{a}{\rightsquigarrow}_{q'} \beta$, then $A\gamma \overset{a}{\rightsquigarrow}_q \beta\delta$;

v) if $q' \xleftarrow[\varepsilon]{A} q' \xleftarrow{\gamma} q$, $A \in \bar{\mathcal{E}}_{q'}$, and $\gamma \overset{a}{\rightsquigarrow}_q \beta$, then $A\gamma \overset{a}{\rightsquigarrow}_q \beta$.

We say that a *string* $\alpha \in \mathcal{V}^*$ is *basic* if it is just one variable ($\alpha \in \mathcal{V}$) or it is a suffix of the right-hand side δ in a rule $A \xrightarrow{a} \delta$ in $\mathcal{R}$; hence the number of basic strings is no bigger than a standard size of $\mathcal{G}$. We say that $\alpha \overset{a}{\rightsquigarrow}_q \beta$ is a *basic move* if α is a basic string. Any basic move $\alpha \overset{a}{\rightsquigarrow}_q \beta$ can be derived in the "deduction system" i) – v) either by using an axiom i) or ii), or by using another basic move with a shorter derivation (in the rules iii) – v)). Hence if we apply i) – v) only to basic strings (i.e., we use i) only if the respective α is basic, and we use iv) or v) only if $A\gamma$ is basic) iteratively as long as new basic moves are being derived, we get all basic moves. Moreover, if $\alpha \overset{a}{\rightsquigarrow}_q \beta$ is a basic move, then β is either $\mathcal{T}_q(\alpha)$ or of the form $\mathcal{T}_{q'}(\delta)\,\gamma$ for some $q' \in Q$ where γ is a suffix of $\mathcal{T}_q(\alpha)$ and δ is the right-hand side of a rule in $\mathcal{R}$ (and $q' \xleftarrow[\gamma]{\gamma} q$); this claim also follows inductively, when inspecting the rules i)–v). There are thus only polynomially many basic moves (in the size of $\mathcal{G}$ and $\mathcal{T}$).

The above observations immediately yield a polynomial algorithm (in the size of $\mathcal{G}$ and $\mathcal{T}$) that constructs all basic moves. A polynomial check of consistency of $\mathcal{T}$ with $\mathcal{G}$ will be thus clear after we show that also non-basic moves of the type $\mathcal{T}_q(A) \overset{a}{\rightsquigarrow}_q \gamma$ and $AC \overset{a}{\rightsquigarrow}_q \gamma$ where $\mathcal{T}(AC) = \mathcal{T}(C) = C$ can be easily constructed, when basic moves are given. Let $\mathcal{T}_q(A) = B\beta$, where $q'' \xleftarrow[B]{B} q' \xleftarrow[\beta]{\beta} q$ (recall that $\mathcal{T}_q(A)$ is a q-normal form); then $B\beta \overset{a}{\rightsquigarrow}_q \gamma$ iff $\gamma = \gamma'\beta$ and $B \overset{a}{\rightsquigarrow}_{q'} \gamma'$. If $q' \xleftarrow[\varepsilon]{A} q' \xleftarrow[C]{C} q$ then $AC \overset{a}{\rightsquigarrow}_q \gamma$ iff $\gamma = \gamma' C$ and $A \overset{a}{\rightsquigarrow}_{q'} \gamma'$ or $A \in \bar{\mathcal{E}}_q$ and $C \overset{a}{\rightsquigarrow}_q \gamma$.

2. Let $\mathcal{T} = (Q, \mathcal{V}, \Delta, q_0)$ be an nfc-transducer that is consistent with a given BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$. We will first show that $\equiv^{\mathcal{T}}$ is a branching bisimulation in $\mathcal{L}_{\mathcal{G}}$ *when assuming* $\alpha \approx \mathcal{T}(\alpha)$ for all $\alpha \in \mathcal{V}^*$, where $\approx$ stands for $\approx_{q_0}$; this assumption will be proven afterwards.

Let us consider some $\alpha \equiv^{\mathcal{T}} \beta$ and a transition $\alpha \xrightarrow{a} \alpha'$; we thus have $\alpha \overset{a}{\rightsquigarrow} \mathcal{T}(\alpha')$ where $\overset{a}{\rightsquigarrow}$ stands for $\overset{a}{\rightsquigarrow}_{q_0}$. If $a = \tau$ and $\mathcal{T}(\alpha) = \mathcal{T}(\alpha')$, then $\alpha' \equiv^{\mathcal{T}} \beta$; so we further suppose that $a \neq \tau$ or $\mathcal{T}(\alpha) \neq \mathcal{T}(\alpha')$. Since $\alpha \approx \mathcal{T}(\alpha) = \mathcal{T}(\beta) \approx \beta$, we must also have $\beta \overset{a}{\rightsquigarrow} \mathcal{T}(\alpha')$. Hence we have $\beta = \beta_0 \xrightarrow{\tau} \beta_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} \beta_k \xrightarrow{a} \beta'$ for some $k \geq 0$ where $\mathcal{T}(\beta_0) = \mathcal{T}(\beta_1) = \cdots = \mathcal{T}(\beta_k)$ and $\mathcal{T}(\beta') = \mathcal{T}(\alpha')$. Therefore $\alpha \equiv^{\mathcal{T}} \beta_i$ for all $i \in [0, k]$, and $\alpha' \equiv^{\mathcal{T}} \beta'$. We have thus verified that $\equiv^{\mathcal{T}}$ is indeed a branching bisimulation.

It remains to prove that $\alpha \approx \mathcal{T}(\alpha)$. We proceed by induction on $|\alpha|$. If $\alpha = \varepsilon$, then the claim is trivial. We now assume $\alpha = A\beta$ where $q' \xleftarrow[\mathcal{T}_q(A)]{A} q \xleftarrow[\mathcal{T}(\beta)]{\beta} q_0$ and $\beta \approx \mathcal{T}(\beta)$. The fact $q \xleftarrow[\mathcal{T}(\beta)]{\mathcal{T}(\beta)} q_0$ (due to the properties of *nfc*-transducers) then implies $A\beta \approx A\,\mathcal{T}(\beta)$ (as can be verified by iv) and v) in the above "deduction system"); for establishing $A\beta \approx \mathcal{T}(A\beta)$ it thus suffices to show that $A\,\mathcal{T}(\beta) \approx \mathcal{T}_q(A)\,\mathcal{T}(\beta)$ (recall that $\mathcal{T}(A\beta) = \mathcal{T}_q(A)\,\mathcal{T}(\beta)$). If $\mathcal{T}_q(A) \neq \varepsilon$, then this follows from $A \approx_q \mathcal{T}_q(A)$ (cf. Def. 4.1(2)). Hence we further assume $\mathcal{T}_q(A) = \varepsilon$.

If $\mathcal{T}(\beta) = \varepsilon$ (hence $q_0 \xleftarrow[\varepsilon]{A} q_0 \xleftarrow[\varepsilon]{\beta} q_0$), then we need to show that $A \approx \varepsilon$ (i.e. $A \approx_{q_0} \varepsilon$); this holds by Def. 4.1(1). If $\mathcal{T}(\beta) = C\delta$, hence $q \xleftarrow[\varepsilon]{A} q \xleftarrow[C]{C} q'' \xleftarrow[\delta]{\delta} q_0$, then $AC\delta \approx C\delta$ follows from $AC \approx_{q''} C$, which holds by Def. 4.1(3). $\square$

4.3. **Canonical transducers.** Given a *normed* BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$, we now show a (non-effective) construction of a canonical transducer $\mathcal{T}^{\mathcal{G}}$. It will turn out that $\mathcal{T}^{\mathcal{G}}$ is an nfc-transducer that is consistent with $\mathcal{G}$ (hence $\equiv^{\mathcal{T}^{\mathcal{G}}} \subseteq \sim$) and for which $\sim \subseteq \equiv^{\mathcal{T}^{\mathcal{G}}}$; hence $\alpha \sim \beta$ in $\mathcal{L}_{\mathcal{G}}$ iff $\mathcal{T}^{\mathcal{G}}(\alpha) = \mathcal{T}^{\mathcal{G}}(\beta)$. We will also get an exponential bound on the size of $\mathcal{T}^{\mathcal{G}}$ (in the size of $\mathcal{G}$). These facts will immediately entail a NExpTime upper bound for the branching bisimilarity problem for normed BPA systems. (We have already touched on this in Section 3, and some further remarks on the complexity are in Section 5.)

In the definition of the transducer $\mathcal{T}^{\mathcal{G}}$ we also use the following technical notions. For $\gamma \in \mathcal{V}^*$ we put

$$R_\gamma = \{X \in \mathcal{V} \mid X\gamma \sim \gamma\}.$$

Each $X \in R_\gamma$ is called a *redundant variable w.r.t.* γ. We say that the prefix α of $\alpha\gamma \in \mathcal{V}^*$ is *redundancy-free* if it cannot be written as $\alpha = \delta X \beta$ where $X\beta\gamma \sim \beta\gamma$.

To make $\mathcal{T}^{\mathcal{G}}$ unique (though this is not crucial), we also assume a linear order on the set $\mathcal{V}$; we say that $\alpha \in \mathcal{V}^*$ is *lexicographically smaller* than $\beta \in \mathcal{V}^*$ if α is a proper suffix of β, or if $\alpha = \alpha' A \gamma$, $\beta = \beta' B \gamma$ and A is less than B in the order on $\mathcal{V}$. (We reflect our right-to-left transducers in this definition.)

We first state the following definition and then we discuss its soundness, which is based on the assumption that $\mathcal{G}$ is normed.

Definition 4.3. For a normed BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ is linearly ordered, we define the *canonical transducer* $\mathcal{T}^{\mathcal{G}} = (Q, \mathcal{V}, \Delta, q_0)$ by the following three points.

i) $Q = \{R_\gamma \mid \gamma \in \mathcal{V}^*\}$. (Hence each state is the set of redundant variables w.r.t. some γ.)
ii) The initial state q_0 is the set R_ε (i.e. the set $\{X \in \mathcal{V} \mid X \sim \varepsilon\}$, which might be empty).
iii) For each $R_\gamma \in Q$ and each $A \in \mathcal{V}$ we put $\Delta(R_\gamma, A) = (R_{A\gamma}, \alpha)$, which is denoted as $R_{A\gamma} \xleftarrow[\alpha]{A} R_\gamma$, where α satisfies the conditions
  a) $\alpha\gamma \sim A\gamma$,
  b) α is a redundancy-free prefix of $\alpha\gamma$,
  and is lexicographically smallest among the longest strings satisfying a) and b).

The soundness of the definition can be shown by the facts established already in [9]; a crucial fact is that $R_\gamma = R_\delta$ implies $\alpha\gamma \sim \beta\gamma \;\Leftrightarrow\; \alpha\delta \sim \beta\delta$. To be self-contained, we also prove these facts (by Prop. 4.5), and then we show the soundness (as a part of Theorem 4.6).

We fix a normed BPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$, and we first define the "relative" equivalences $\alpha \sim_R \beta$ and the "relative" cc-norms $\langle\!\langle \alpha \rangle\!\rangle_R$ for all $R \subseteq \mathcal{V}$, via the LTSs $\mathcal{L}_{\mathcal{G},R}$; for a fixed set $R \subseteq \mathcal{V}$ we stipulate:

- The LTS $\mathcal{L}_{\mathcal{G},R}$ arises from $\mathcal{L}_{\mathcal{G}} = (\mathcal{V}^*, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$ by declaring all $\alpha \in R^*$ to be silent states; technically we simply remove all their outgoing transitions. Hence $\alpha \in R^*$ satisfies $\alpha \sim \varepsilon$ in $\mathcal{L}_{\mathcal{G},R}$.
- $\alpha \sim_R \beta \;\Leftrightarrow_{\mathrm{df}}\; \alpha \sim \beta$ in $\mathcal{L}_{\mathcal{G},R}$.
- $\langle\!\langle \alpha \rangle\!\rangle_R$ is equal to $\langle\!\langle \alpha \rangle\!\rangle$ in $\mathcal{L}_{\mathcal{G},R}$.

Remark 4.4. Unlike in [9], the definition is not restricted to $R = R_\gamma$ for $\gamma \in \mathcal{V}^*$, and the claims that we derive for R_γ can be naturally extended to the general cases $R \subseteq \mathcal{V}$. Similarly we could define the states of $\mathcal{T}^{\mathcal{G}}$ to be all sets $R \subseteq \mathcal{V}$ (not only those reachable from R_ε). Additional remarks are given in Section 5.

Now we note a few facts that already appeared in [9].

Proposition 4.5. *For any nBPA system $\mathcal{G}$ the following claims hold:*
(1) $\gamma \sim \delta$ *implies* $\alpha\gamma \sim \alpha\delta$*;*
(2) $\alpha\gamma \sim \gamma$ *iff* $\alpha \in (R_\gamma)^*$*;*
(3) $\alpha\gamma \sim \beta\gamma$ *iff* $\alpha \sim_{R_\gamma} \beta$*.*

Proof. We assume an nBPA system $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$.

(1) If $\gamma \sim \delta$, then the set $\mathcal{B} = \sim \cup \{(\alpha\gamma, \alpha\delta) \mid \alpha \in \mathcal{V}^*\}$ can be easily verified to be a branching bisimulation (and thus $\mathcal{B} = \sim$ in $\mathcal{L}_\mathcal{G}$).

(2) If $\alpha = \alpha' Y$ where $Y \in R_\gamma = \{X \mid X\gamma \sim \gamma\}$, then $\alpha\gamma = \alpha' Y \gamma \sim \alpha'\gamma$ (by 1.); using this fact repeatedly, $\alpha \in (R_\gamma)^*$ entails $\alpha\gamma \sim \gamma$.

Suppose $\alpha \notin (R_\gamma)^*$, hence $\alpha = \alpha' Y \alpha''$ where $\alpha'' \in (R_\gamma)^*$ and $Y \notin R_\gamma$; thus $\alpha\gamma \sim \alpha' Y \gamma$. Since $Y\gamma \not\sim \gamma$, we have $\langle\!\langle Y\gamma \rangle\!\rangle > \langle\!\langle \gamma \rangle\!\rangle$ (since any path $Y\gamma \xrightarrow{u} \gamma$ contains at least one class-changing transition); this entails $\langle\!\langle \alpha' Y \gamma \rangle\!\rangle \geq \langle\!\langle Y\gamma \rangle\!\rangle > \langle\!\langle \gamma \rangle\!\rangle$, and thus $\alpha' Y \gamma \not\sim \gamma$ (by Observation 2.1). Since $\alpha\gamma \sim \alpha' Y \gamma$, we get $\alpha\gamma \not\sim \gamma$.

(3) a) We first show the implication $\alpha\gamma \sim \beta\gamma \;\Rightarrow\; \alpha \sim_{R_\gamma} \beta$. This will be clear when we show that for any $\gamma \in \mathcal{V}^*$ the set
$$\mathcal{B} = \{(\alpha, \beta) \mid \alpha\gamma \sim \beta\gamma\}$$
is a branching bisimulation in $\mathcal{L}_{\mathcal{G},R_\gamma}$. Let $(\alpha, \beta) \in \mathcal{B}$ and $\alpha \xrightarrow{a} \alpha'$ in $\mathcal{L}_{\mathcal{G},R_\gamma}$; we will show that the move $\alpha \xrightarrow{a} \alpha'$ can be matched from β in $\mathcal{L}_{\mathcal{G},R_\gamma}$. We note that $\alpha \notin (R_\gamma)^*$ (since it has an outgoing transition in $\mathcal{L}_{\mathcal{G},R_\gamma}$) and thus $\alpha\gamma \not\sim \gamma$ (by 2.); this also entails $\beta\gamma \not\sim \gamma$ since $\alpha\gamma \sim \beta\gamma$. We also have the move $\alpha\gamma \xrightarrow{a} \alpha'\gamma$ in $\mathcal{L}_\mathcal{G}$.

- If $a = \tau$ and $\alpha'\gamma \sim \alpha\gamma$, hence also $\alpha'\gamma \sim \beta\gamma$, then $(\alpha', \beta) \in \mathcal{B}$.
- If $a \neq \tau$ or $\alpha'\gamma \not\sim \alpha\gamma$, then in $\mathcal{L}_\mathcal{G}$ we must have $\beta\gamma = \delta_0 \xrightarrow{\tau} \delta_1 \cdots \xrightarrow{\tau} \delta_k \xrightarrow{a} \delta$ where $\alpha\gamma \sim \beta\gamma \sim \delta_i$ for all $i \in [0, k]$ and $\alpha'\gamma \sim \delta$. Since $\beta\gamma \not\sim \gamma$, we have $\delta_i \not\sim \gamma$ for all $i \in [0, k]$; this entails that for each $i \in [0, k]$ we have $\delta_i = \beta_i\gamma$ where $\beta_i \notin (R_\gamma)^*$, $\delta = \beta'\gamma$, and $\beta = \beta_0 \xrightarrow{\tau} \beta_1 \cdots \xrightarrow{\tau} \beta_k \xrightarrow{a} \beta'$ is a path in $\mathcal{L}_{\mathcal{G},R_\gamma}$. Hence $\alpha\gamma \sim \beta_i\gamma$ for all $i \in [0, k]$, and $\alpha'\gamma \sim \beta'\gamma$; therefore $(\alpha, \beta_i) \in \mathcal{B}$ for all $i \in [0, k]$, and $(\alpha', \beta') \in \mathcal{B}$.

b) Now we show the implication $\alpha \sim_{R_\gamma} \beta \;\Rightarrow\; \alpha\gamma \sim \beta\gamma$. This will be clear when we show that for any $\gamma \in \mathcal{V}^*$ the set
$$\mathcal{B} = \sim \cup \{(\alpha\gamma, \beta\gamma) \mid \alpha \sim_{R_\gamma} \beta\}$$
is a branching bisimulation in $\mathcal{L}_\mathcal{G}$ (which implies $\mathcal{B} = \sim$ in $\mathcal{L}_\mathcal{G}$). It suffices to show that for any $(\delta_1, \delta_2) \in \mathcal{B}$ any move $\delta_1 \xrightarrow{a} \delta$ can be matched from δ_2. If $\delta_1 \sim \delta_2$, then this follows from the definition of $\sim$. Hence it suffices to consider the case $\delta_1 = \alpha\gamma$ and $\delta_2 = \beta\gamma$ where $\alpha \sim_{R_\gamma} \beta$.

- If $\alpha \in (R_\gamma)^*$, then $\alpha\gamma \sim \gamma$ and $\alpha \sim_{R_\gamma} \varepsilon \sim_{R_\gamma} \beta$; both α and β are silent states in $\mathcal{L}_{\mathcal{G},R_\gamma}$. This entails that either $\beta \in (R_\gamma)^*$, or any move $\beta \xrightarrow{b} \beta'$ in $\mathcal{L}_{\mathcal{G},R_\gamma}$ satisfies that $b = \tau$ and that β' is silent in $\mathcal{L}_{\mathcal{G},R_\gamma}$; since $\mathcal{G}$ is normed, we must also have $\beta \xrightarrow{u} \beta'$ where $u \in \{\tau\}^*$ and $\beta' \in (R_\gamma)^*$. It follows that $\beta\gamma \sim \gamma$ (since the set $\sim \cup \{(\beta'\gamma, \gamma) \mid \beta' \text{ is silent in } \mathcal{L}_{\mathcal{G},R_\gamma}\}$ is a branching bisimulation); hence $\beta \in (R_\gamma)^*$, in fact (by 2.). We thus have the case $\delta_1 \sim \delta_2$ (since $\alpha\gamma \sim \gamma \sim \beta\gamma$); the move $\delta_1 \xrightarrow{a} \delta$ can be thus matched from δ_2.
- If $\alpha \notin (R_\gamma)^*$, then the move $\delta_1 \xrightarrow{a} \delta$, i.e. $\alpha\gamma \xrightarrow{a} \delta$, can be presented as $\alpha\gamma \xrightarrow{a} \alpha'\gamma$ where $\alpha \xrightarrow{a} \alpha'$ in $\mathcal{L}_{\mathcal{G},R_\gamma}$. If $a = \tau$ and $\alpha' \sim_{R_\gamma} \alpha$, then we have

$(\delta, \delta_2) = (\alpha'\gamma, \beta\gamma) \in \mathcal{B}$. Now assume $a \neq \tau$ or $\alpha' \not\sim_{R_\gamma} \alpha$. Since $\alpha \sim_{R_\gamma} \beta$, in $\mathcal{L}_{\mathcal{G},R_\gamma}$ we must have $\beta = \beta_0 \xrightarrow{\tau} \beta_1 \cdots \xrightarrow{\tau} \beta_k \xrightarrow{a} \beta'$ where $\alpha \sim_{R_\gamma} \beta_i$ for all $i \in [0,k]$ and $\alpha' \sim_{R_\gamma} \beta'$. But then in $\mathcal{L}_{\mathcal{G}}$ we have $\delta_2 = \beta_0\gamma \xrightarrow{\tau} \beta_1\gamma \cdots \xrightarrow{\tau} \beta_k\gamma \xrightarrow{a} \beta'\gamma$ where $(\alpha\gamma, \beta_i\gamma) \in \mathcal{B}$ for all $i \in [0,k]$ and $(\alpha'\gamma, \beta'\gamma) \in \mathcal{B}$. Hence the move $\delta_1 \xrightarrow{a} \delta$ (i.e. $\alpha\gamma \xrightarrow{a} \alpha'\gamma$) can be matched from $\delta_2 = \beta\gamma$. $\square$

We now prove the announced properties of $\mathcal{T}^{\mathcal{G}}$ (from Def. 4.3).

Theorem 4.6. *For any normed BPA system $\mathcal{G}$, the canonical transducer $\mathcal{T}^{\mathcal{G}}$ has the following properties:*
(1) *$\mathcal{T}^{\mathcal{G}}$ is an nfc-transducer that is consistent with $\mathcal{G}$.*
(2) *$\equiv^{\mathcal{T}^{\mathcal{G}}} = \sim$ (i.e., $\mathcal{T}^{\mathcal{G}}(\alpha) = \mathcal{T}^{\mathcal{G}}(\beta)$ iff $\alpha \sim \beta$ in $\mathcal{L}_{\mathcal{G}}$).*
(3) *The size of $\mathcal{T}^{\mathcal{G}}$ is bounded by an exponential function of the size of $\mathcal{G}$.*

Proof. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$ be an nBPA system and let $\mathcal{T}^{\mathcal{G}} = (Q, \mathcal{V}, \Delta, q_0)$ be as in Def. 4.3.

(1) (First part.) We now show that $\mathcal{T}^{\mathcal{G}}$ is an nfc-transducer; the consistency with $\mathcal{G}$ is shown in the second part, after the point (2) is established.

We first need to show that the function Δ, presented by four-tuples $R_{A\gamma} \xleftarrow[\alpha]{A} R_\gamma$, is defined soundly. Let us assume $R_\gamma = R_\delta$, hence $\sim_{R_\gamma} = \sim_{R_\delta}$. By Prop. 4.5(3) we deduce that $XA\gamma \sim A\gamma$ iff $XA\delta \sim A\delta$; hence $R_{A\gamma} = R_{A\delta}$. Similarly we deduce that $\alpha\gamma \sim A\gamma$ iff $\alpha\delta \sim A\delta$, and that α is a redundancy-free prefix of $\alpha\gamma$ iff α is a redundancy-free prefix of $\alpha\delta$. Hence the strings α satisfying a) and b) in Def. 4.3(iii) are determined by the set R_γ. The set of such strings is nonempty (since it contains $\alpha = A$ or $\alpha = \varepsilon$); once we show that this set is finite, the soundness of Δ is clear. The finitiness follows from the fact that $\alpha\gamma \sim A\gamma$ entails $\langle\!\langle \alpha\gamma \rangle\!\rangle = \langle\!\langle A\gamma \rangle\!\rangle$ (Observation 2.1), and from the obvious fact that $\langle\!\langle \alpha\gamma \rangle\!\rangle \geq |\alpha| + \langle\!\langle \gamma \rangle\!\rangle$ when α is a redundancy-free prefix of $\alpha\gamma$. (We have already observed that $X\beta \not\sim \beta$ entails that any path $X\beta \xrightarrow{u} \beta$ has at least one class-changing transition.)

Hence $\mathcal{T}^{\mathcal{G}}$ is indeed a transducer. We show that it is an nfc-transducer, i.e., $R' \xleftarrow[\alpha]{A} R$ implies that α is an R-normal form and $R' \xleftarrow[\alpha]{\alpha} R$.

Let us consider $R_{A\gamma} \xleftarrow[\alpha]{A} R_\gamma$. By definition of $\mathcal{T}^{\mathcal{G}}$, and by Prop. 4.5(3), the string α is lexicographically smallest among the longest strings that satisfy $\alpha \sim_{R_\gamma} A$ and are R_γ-redundancy free, by which we mean that $\alpha = \delta X\beta$ entails $X\beta \not\sim_{R_\gamma} \beta$. This obviously entails that $\alpha = \varepsilon$ iff $A \in R_\gamma$ (by recalling Prop. 4.5(2)); in this case we have $R_\gamma \xleftarrow[\varepsilon]{A} R_\gamma$.

We thus further suppose $A \notin R_\gamma$. Since $A\gamma \sim \alpha\gamma$, we have $XA\gamma \sim X\alpha\gamma$ (recall Prop. 4.5(1)); therefore $XA\gamma \sim A\gamma$ iff $X\alpha\gamma \sim \alpha\gamma$. Hence $R_{\alpha\gamma} = R_{A\gamma}$, and we thus have $R_{A\gamma} \xleftarrow{\alpha} R_\gamma$. It remains to show that α is an R_γ-normal form. For the sake of contradiction we suppose that it is not the case; hence we have $\alpha = \alpha' B A_\ell A_{\ell-1} \cdots A_2 A_1$ for some $\ell \geq 0$ where

$$R_{\alpha\gamma} \xleftarrow[\beta']{\alpha'} R_{\ell+1} \xleftarrow[\beta]{B} R_\ell \xleftarrow[A_\ell]{A_\ell} \cdots R_2 \xleftarrow[A_2]{A_2} R_1 \xleftarrow[A_1]{A_1} R_\gamma \text{ and } \beta \neq B.$$

By the definition of $\mathcal{T}^{\mathcal{G}}$ we have $B \sim_{R_\ell} \beta$ and $\beta \neq \varepsilon$ (we have $B \notin R_\ell$ since α is R_γ-redundancy free), which entails that $|\alpha'\beta A_\ell \cdots A_2 A_1| > |\alpha|$ or $|\alpha'\beta A_\ell \cdots A_2 A_1| = |\alpha|$ and $\alpha'\beta A_\ell \cdots A_2 A_1$ is lexicographically smaller than α. The fact $\beta \sim_{R_\ell} B$ entails

$\beta A_\ell \cdots A_1\gamma \sim BA_\ell \cdots A_1\gamma$, and thus $\alpha'\beta A_\ell \cdots A_1\gamma \sim \alpha' BA_\ell \cdots A_1\gamma = \alpha\gamma$ (using Prop. 4.5(1,3)). This forces us to conclude that $\alpha'\beta A_\ell \cdots A_1$ is not R_γ-redundancy free (due to the choice of α in $\mathcal{T}^{\mathcal{G}}$). But this is impossible, since β is R_ℓ-redundancy free, α' is $R_{\ell+1}$-redundancy free, and $R_{\ell+1} \xleftarrow{\beta} R_\ell$ (since $\beta \sim_{R_\ell} B$ entails $X\beta \sim_{R_\ell} \beta$ iff $XB \sim_{R_\ell} B$).

(2) We will show the following (more general) claim for the nfc-transducer $\mathcal{T}^{\mathcal{G}}$:

$$\alpha \sim_R \beta \text{ iff } \mathcal{T}^{\mathcal{G}}_R(\alpha) = \mathcal{T}^{\mathcal{G}}_R(\beta) \text{ (for any } R \in Q). \tag{4.2}$$

We first note the fact that

$$\alpha \sim_R \mathcal{T}^{\mathcal{G}}_R(\alpha) \text{ (for all } \alpha \in \mathcal{V}^* \text{ and } R \in Q), \tag{4.3}$$

using an induction on $|\alpha|$. For $\alpha = \varepsilon$ the fact ($\varepsilon \sim_R \varepsilon$) is trivial. For $\alpha = \alpha' A$ we have $R'' \xleftarrow[\mathcal{T}^{\mathcal{G}}_{R'}(\alpha')]{\alpha'} R' \xleftarrow[\mathcal{T}^{\mathcal{G}}_R(A)]{A} R$ where $A \sim_R \mathcal{T}^{\mathcal{G}}_R(A)$ by the definition of $\mathcal{T}^{\mathcal{G}}$ and $\alpha' \sim_{R'} \mathcal{T}^{\mathcal{G}}_{R'}(\alpha')$ by the induction hypothesis. Hence $\alpha' A \sim_R \mathcal{T}^{\mathcal{G}}_R(\alpha' A)$, by applying Prop. 4.5(3).

The "if-direction" of (4.2) thus follows (since $\alpha \sim_R \mathcal{T}^{\mathcal{G}}_R(\alpha) = \mathcal{T}^{\mathcal{G}}_R(\beta) \sim_R \beta$ implies $\alpha \sim_R \beta$).

We now show the "only-if-direction" of (4.2). For the sake of contradiction, suppose there are $\alpha \sim_R \beta$ for which $\mathcal{T}^{\mathcal{G}}_R(\alpha) \neq \mathcal{T}^{\mathcal{G}}_R(\beta)$. By (4.3) we deduce that there are two different R-normal forms α, β such that $\alpha \sim_R \beta$; let us consider such α, β. We thus have $\alpha = \alpha' A\gamma$, $\beta = \beta' B\gamma$ where $A \neq B$; hence $\alpha' A \sim_{R'} \beta' B$ where $R' \xleftarrow[\gamma]{\gamma} R$, and $\alpha' A$, $\beta' B$ are R'-normal forms. Hence we immediately choose some $R \in Q$ and two R-normal forms αA, βB where $A \neq B$ and $\alpha A \sim_R \beta B$; w.l.o.g. we assume $\langle\!\langle A \rangle\!\rangle_R \geq \langle\!\langle B \rangle\!\rangle_R$.

We now consider a path $\alpha A \xrightarrow{u} \gamma A$ in $\mathcal{L}_{\mathcal{G},R}$ where $\langle\!\langle \gamma A \rangle\!\rangle_R = \langle\!\langle A \rangle\!\rangle_R$ and the R-cc-norms $\langle\!\langle . \rangle\!\rangle_R$ of all processes on this path before γA are bigger than $\langle\!\langle A \rangle\!\rangle_R$. (We can have $u = \varepsilon$ and $\gamma = \alpha$; in this case $\alpha = \varepsilon$ since otherwise $\langle\!\langle \alpha A \rangle\!\rangle_R > \langle\!\langle A \rangle\!\rangle_R$ due to R-redundancy freeness of αA, which is an R-normal form.)

We recall that $\gamma_1 \sim_R \gamma_2$ implies $\langle\!\langle \gamma_1 \rangle\!\rangle_R = \langle\!\langle \gamma_2 \rangle\!\rangle_R$ (by Observation 2.1). Since $\alpha A \sim_R \beta B$, the path $\alpha A \xrightarrow{u} \gamma A$ must have a matching path $\beta B \xrightarrow{v} \delta'$ where $\gamma A \sim_R \delta'$ (hence $\langle\!\langle \delta' \rangle\!\rangle_R = \langle\!\langle \gamma A \rangle\!\rangle_R = \langle\!\langle A \rangle\!\rangle_R \geq \langle\!\langle B \rangle\!\rangle_R$) and all processes on the path before δ' have the R-cc-norms bigger than $\langle\!\langle A \rangle\!\rangle_R$; necessarily $\delta' = \delta B$ for some δ.

We now derive a contradiction. Since $\langle\!\langle \gamma A \rangle\!\rangle_R = \langle\!\langle A \rangle\!\rangle_R$, we have $\gamma A \sim_R A$ (there must be a path $\gamma A \xrightarrow{w} A$ with no class-changing transition), and thus $A \sim_R \delta B$. We have $\mathcal{T}^{\mathcal{G}}_R(A) = A$ (A is an R-prime since αA is an R-normal form) and $A \sim_R \delta B \sim_R \mathcal{T}^{\mathcal{G}}_R(\delta B) = \delta'' B$ (for some δ''; B is also an R-prime since βB is an R-normal form). If $\delta'' \neq \varepsilon$, then A is not a longest R-redundancy-free string from the class $[A]_{\sim_R}$ (which contradicts with $\mathcal{T}^{\mathcal{G}}_R(A) = A$); if $\delta'' = \varepsilon$, then $A \sim_R B$ and one of $\mathcal{T}^{\mathcal{G}}_R(A) = A$, $\mathcal{T}^{\mathcal{G}}_R(B) = B$ violates the "lexicographically smallest" condition.

(1) (Second part.) We show that $\mathcal{T}^{\mathcal{G}}$ is consistent with $\mathcal{G}$. We have to show $A \approx_{R_\varepsilon} \varepsilon$, $A \approx_R \mathcal{T}^{\mathcal{G}}_R(A)$, and $AC \approx_R C$ in the cases specified in Def. 4.1. Since in these cases we have $A \sim_{R_\varepsilon} \varepsilon$, $A \sim_R \mathcal{T}^{\mathcal{G}}_R(A)$, and $AC \sim_R C$ (as follows from (4.3)), it suffices to show that $\alpha \sim_R \beta$ implies $\alpha \approx_R \beta$.

So let us assume $\alpha \sim_R \beta$ (hence $\mathcal{T}^{\mathcal{G}}_R(\alpha) = \mathcal{T}^{\mathcal{G}}_R(\beta)$ by (4.2)), and suppose $\alpha \overset{a}{\rightsquigarrow}_R \alpha'$; we will be done if we show that $\beta \overset{a}{\rightsquigarrow}_R \alpha'$. If $a = \tau$ and $\alpha' = \mathcal{T}^{\mathcal{G}}_R(\alpha)$, then indeed $\beta \overset{a}{\rightsquigarrow}_R \alpha'$ since it is $\beta \overset{\tau}{\rightsquigarrow}_R \mathcal{T}^{\mathcal{G}}_R(\beta)$ in this case. Otherwise we have $\alpha = \alpha_0 \xrightarrow{\tau} \alpha_1 \cdots \xrightarrow{\tau} \alpha_k \xrightarrow{a} \alpha''$

where $\mathcal{T}^{\mathcal{G}}_R(\alpha) = \mathcal{T}^{\mathcal{G}}_R(\beta) = \mathcal{T}^{\mathcal{G}}_R(\alpha_0) = \cdots = \mathcal{T}^{\mathcal{G}}_R(\alpha_k)$ and $\alpha' = \mathcal{T}^{\mathcal{G}}_R(\alpha'')$. By (4.2) we have $\alpha \sim_R \alpha_0 \sim_R \cdots \sim_R \alpha_k$, hence also $\alpha_k \sim_R \beta$. Since $\alpha_k \xrightarrow{a} \alpha''$, we must have $\beta = \beta_0 \xrightarrow{\tau} \beta_1 \cdots \xrightarrow{\tau} \beta_{k'} \xrightarrow{a} \beta''$ where $\beta \sim_R \beta_0 \sim_R \cdots \sim_R \beta_{k'}$ and $\beta'' \sim_R \alpha''$. By (4.2) we thus have $\mathcal{T}^{\mathcal{G}}_R(\beta) = \mathcal{T}^{\mathcal{G}}_R(\beta_0) = \cdots = \mathcal{T}^{\mathcal{G}}_R(\beta_{k'})$ and $\mathcal{T}^{\mathcal{G}}_R(\beta'') = \alpha'$; hence $\beta \overset{a}{\rightsquigarrow}_R \alpha'$.

(3) The number of states of $\mathcal{T}^{\mathcal{G}} = (Q, \mathcal{V}, \Delta, q_0)$ is bounded by the number of subsets of $\mathcal{V}$ (hence $|Q| \leq 2^{|\mathcal{V}|}$). Function Δ can be presented by $|Q| \cdot |\mathcal{V}|$ expressions $\Delta(R, A) = (R', \alpha)$ where α (i.e., $\mathcal{T}^{\mathcal{G}}_R(A)$) is an R-normal form satisfying $A \sim_R \alpha$, and thus also $\langle\!\langle A \rangle\!\rangle_R = \langle\!\langle \alpha \rangle\!\rangle_R$. It is straightforward to note that $|\alpha| \leq \langle\!\langle \alpha \rangle\!\rangle_R = \langle\!\langle A \rangle\!\rangle_R \leq \|A\|$, and $\|A\|$ is at most exponential in the size of $\mathcal{G}$ (by Prop. 2.3(5)). The overall size of $\mathcal{T}^{\mathcal{G}}$ is thus indeed at most exponential in the size of $\mathcal{G}$. □

5. Additional remarks

The main result of the paper is captured by Theorem 4.6. Together with Lemma 4.2 it places the nBPA-bbis problem in NExpTime, as was discussed in Section 3.

In the arxiv-version of [8] we (Czerwiński and Jančar) mentioned that a natural way for a further research is to look for a deterministic exponential algorithm that would compute the decompositions (or a *base* in the terminology of [8]) by proceeding via a certain series of decreasing over-approximations. In the transducer framework, this suggests to build the canonical transducer $\mathcal{T}^{\mathcal{G}}$ by a series of stepwise refined over-approximations.

We mentioned in Section 4 that the relative equivalences $\sim_R$ (defined via the LTSs $\mathcal{L}_{\mathcal{G},R}$) make sense also for general $R \subseteq \mathcal{V}$, not only for R_γ, so we could think of constructing such a more general transducer; its (exponentially many) control states are thus given. It is then natural to use *nondeterministic* transducers $\mathcal{T}$ as the over-approximations of $\mathcal{T}^{\mathcal{G}}$, and to try to find a method of some safe successive decreasing of the nondeterminism by finding where the current $\mathcal{T}$ violates the consistency and other conditions satisfied by $\mathcal{T}^{\mathcal{G}}$. (An example of one such condition that has not been mentioned explicitly is that $A \in R_\gamma$ entails $A \xrightarrow{u} \varepsilon$ for $u \in \{\tau\}^*$.)

Here we do not pursue such a task further; it would be interesting to clarify if the approach by He and Huang [11] can be seen as accomplishing it.

Author's acknowledgements. I would like to thank especially to Wojciech Czerwiński, with whom we performed the research reported on in [8]. I felt appropriate to elaborate this new version but Wojciech could not participate because of other duties, and he suggested that I do this alone. Nevertheless he commented a preliminary version of this paper, for which I also warmly thank him.

I also thank the anonymous reviewers for their helpful comments.

References

[1] J. C. M. Baeten, J. A. Bergstra, and J. W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *J. ACM*, 40(3):653–682, 1993.

[2] J. L. Balcázar, J. Gabarró, and M. Santha. Deciding bisimilarity is P-complete. *Formal Asp. Comput.*, 4(6A):638–648, 1992.

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

[4] D. Caucal, D. T. Huynh, and L. Tian. Deciding branching bimiliarity of normed context-free processes is in Σ_2^p. *Inf. Comput.*, 118(2):306–315, 1995.

[5] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. *Inf. Comput.*, 121(2):143–148, 1995.

[6] W. Czerwiński. *Partially-commutative context-free graphs.* PhD thesis, University of Warsaw, 2012. Available at www.mimuw.edu.pl/˜wczerwin.

[7] W. Czerwiński, P. Hofman, and S. Lasota. Decidability of branching bisimulation on normed commutative context-free processes. *Theory Comput. Syst.*, 55(1):136–169, 2014.

[8] W. Czerwiński and P. Jančar. Branching bisimilarity of normed BPA processes is in NEXPTIME. In *Proc. LiCS'15*, pages 168–179. IEEE, 2015. A preliminary version at arxiv.org/abs/1407.0645.

[9] Y. Fu. Checking equality and regularity for normed BPA with silent moves. In *Proc. ICALP'13*, volume 7966 of *Lecture Notes in Computer Science*, pages 238–249. Springer, 2013.

[10] C. He. A polynomial time algorithm for deciding branching bisimilarity on totally normed BPA. *CoRR*, abs/1411.4157, 2014.

[11] C. He and M. Huang. Branching bisimilarity on normed BPA is EXPTIME-complete. In *Proc. LiCS'15*, pages 180–191. IEEE, 2015. A preliminary version at arxiv.org/abs/1501.04748.

[12] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theor. Comput. Sci.*, 158(1&2):143–159, 1996.

[13] M. Huang and Q. Yin. Two lower bounds for BPA. In *Proc. Concur'17*, volume 85 of *LIPIcs*, pages 20:1–20:16. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017.

[14] H. Hüttel. Silence is golden: Branching bisimilarity is decidable for context-free processes. In *Proc. CAV'91*, volume 575 of *LNCS*, pages 2–12. Springer, 1992.

[15] P. Jančar. Bisimilarity on basic process algebra is in 2-EXPTIME (an explicit proof). *Logical Methods in Computer Science*, 9(1:10):1–19, 2013.

[16] P. Jančar. Branching bisimilarity of normed BPA processes as a rational monoid. *CoRR*, abs/1602.05151v1, 2016.

[17] S. Kiefer. BPA bisimilarity is EXPTIME-hard. *Inf. Process. Lett.*, 113(4):101–106, 2013.

[18] R. Mayr. Weak bisimilarity and regularity of context-free processes is EXPTIME-hard. *Theor. Comput. Sci.*, 330(3):553–575, 2005.

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

[20] J. Sakarovitch. Easy multiplications. I. The realm of Kleene's theorem. *Inf. Comput.*, 74(3):173–197, 1987.

[21] J. Srba. Roadmap of infinite results. In *Current Trends In Theoretical Computer Science, The Challenge of the New Century*, volume 2, pages 337–350. World Scientific Publishing Co., 2004. An updated version at http://users-cs.au.dk/srba/roadmap/.

[22] R. J. van Glabbeek and W. P. Weijland. Branching time and abstraction in bisimulation semantics. *J. ACM*, 43(3):555–600, 1996.

[23] Q. Yin, Y. Fu, C. He, M. Huang, and X. Tao. Branching bisimilarity checking for PRS. In *Proc. ICALP'14*, volume 8573 of *Lecture Notes in Computer Science*, pages 363–374. Springer, 2014.