Silence is Golden: Branching Bisimilarity is Decidable for Context-Free Processes

Hans Hüttel *
hans@lfcs.ed.ac.uk

Laboratory for Foundations of Computer Science
James Clerk Maxwell Building
University of Edinburgh
Edinburgh EH9 3JZ
SCOTLAND

Summer 1991

Abstract

We show that the branching bisimulation equivalence introduced by Rob van Glabbeek is decidable for the class of normed, recursively defined BPA processes with silent actions, thus generalizing the decidability result for strong bisimilarity by Baeten, Bergstra, and Klop.

1 Introduction

In their paper [BBK87a] Baeten, Bergstra, and Klop showed that strong bisimulation equivalence is decidable for normed recursively defined BPA processes [BK88], a class of processes corresponding to that of irredundant context-free grammars without empty productions. In this paper we generalize this result by showing that the branching bisimilation equivalence introduced by van Glabbeek and Weijland in [vGW89] is decidable for the class of normed recursively defined BPA processes with silent actions. The proof generalizes that of the decidability of strong bisimulation equivalence given by Colin Stirling and the present author in [HS91], relying as it does on a similar tableau-based decision method.

*Partially supported by the Danish Research Academy.

This tableau method is related to the equivalence algorithms for certain classes of context-free grammars introduced by Korenjak and Hopcroft [KH66], and it directly reflects usual intuitions about determining the equivalence of processes by means of successive matchings of transitions.

Since the class of processes considered allows infinite state spaces, the present result further underpins the fact that the decidability of behavioural equivalences can extend beyond the finite-state case, and that decidability should be seen as a criterion for determining the relative merits and deficiencies of behavioural equivalences.

Section 2 introduces the notion of branching bisimilarity and the class of normed $\mathrm{BPA}^{\tau}_{\mathrm{rec}}$ processes. In Section 3 we describe the tableau system, prove its soundness and completeness, give a complexity measure and establish the decidability result of branching bisimilarity. Section 4 sums up conclusions and gives suggestions for further work.

2 Preliminaries

2.1 Branching bisimilarity

The process that we will be looking at have their behavioural semantics given by labelled transition system with silent actions $\Gamma = (Pr, Act \cup \{\tau\}, \rightarrow)$. Here the weak transition relations $\{\stackrel{s}{\Longrightarrow} \mid s \in Act \cup \{\epsilon\}\}$ are given by $\stackrel{a}{\Longrightarrow} \; = \; \stackrel{\tau}{\rightarrow}^{*} \stackrel{a}{\rightarrow} \stackrel{\tau}{\rightarrow}^{*}$ for $a \in Act$ and $\stackrel{\epsilon}{\Longrightarrow} \; = \; \stackrel{\tau}{\rightarrow}^{*}$, respectively.

The notion of branching bisimilarity, put forward by van Glabbeek and Weijland in [vGW89], is closely related to the notion of stuttering equivalence on Kripke structures considered in e.g. [BCG88]. The difference between branching bisimilarity and weak bisimilarity [Mil89] is that changes in branching properties caused by individual τ-transitions must always be taken into account.

Definition 2.1 *A* branching bisimulation (bb) *on Γ is a symmetric relation $R \subseteq Pr \times Pr$ such that whenever pRq for any $a \in Act \cup \{\tau\}$ we have that $p \stackrel{a}{\rightarrow} p'$ implies*

- *$a = \tau$ and $p'Rq$ or*
- *there exist q_1', q' such that $q \stackrel{\epsilon}{\Longrightarrow} q_1' \stackrel{a}{\rightarrow} q'$ with $pRq_1', p'Rq'$*

If pRq for some bb R we write $p \approx_b q$.

An equivalent definition which reflects this *stuttering property* better is the one below which we will be using in the tableau system presented in Section 3. For a proof that this definition is indeed equivalent, see [vG90].

Proposition 2.1 *A branching bisimulation on Γ is a symmetric relation $R \subseteq Pr \times Pr$ such that whenever pRq for any $a \in Act \cup \{\tau\}$ we have that if $p \stackrel{a}{\rightarrow} p'$ then either*

- *$a = \tau$ and $p' R q$ or*
- *there exist $q'_0, \ldots, q'_n, q'$ such that $q = q'_0 \xrightarrow{\tau} q'_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} q'_n \xrightarrow{a} q'$ with pRq'_i for $0 \leq i \leq n$ and $p'Rq'$.*

2.2 Normed $\mathbf{BPA}^{\tau}_{\mathrm{rec}}$

In what follows we look at the class of processes $\mathrm{BPA}^{\tau}_{\mathrm{rec}}$ [BK88]. These are processes given as systems of defining equations $\Delta = \{X_i \overset{\mathrm{def}}{=} E_i \mid 1 \leq i \leq m\}$ where the E_i's are given by the syntax

$$E ::= a \mid \tau \mid EE \mid E + E \mid X$$

where a ranges over some set Act of actions and X ranges over a set $Var = \{X_i \mid 1 \leq i \leq n\}$ of variables. $+$ is nondeterministic choice, and juxtaposition is sequential composition. Throughout, elements of Var^* will be denoted by Greek letters: $\alpha, \beta, \ldots$. In particular we let ϵ denote the empty string of variables, corresponding to the empty process, and have the convention that ϵF is the same as F. The operational semantics given by the transition relations $\{ \xrightarrow{a} \mid a \in Act \cup \{\tau\}\}$ is defined by [BK88]:

$$\frac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'} \qquad \frac{F \xrightarrow{a} F'}{E + F \xrightarrow{a} F'}$$

$$\frac{E \xrightarrow{a} E'}{EF \xrightarrow{a} E'F} \qquad a \xrightarrow{a} \epsilon \quad a \in Act \cup \{\tau\}$$

$$\frac{E \xrightarrow{a} E'}{X \xrightarrow{a} E'} \qquad X \overset{\mathrm{def}}{=} E \in \Delta$$

We restrict our attention to *weakly normed* families.

Definition 2.2 *The* weak norm *of any $X \in Var$ is defined as*

$$||X|| = \min\{length(w) \mid X \overset{w}{\Longrightarrow} \epsilon, w \in Act^*\}$$

A system of defining equations Δ is weakly normed *if for any variable $X \in Var$ $0 < ||X|| < \infty$. The maximal norm of any variable in Δ is defined by $m_\Delta = \max\{||X|| \mid X \in Var\}$.*

The weak norm is additive under sequential composition:

Proposition 2.2 *For $\alpha, \beta \in Var^*$ $||\alpha\beta|| = ||\alpha|| + ||\beta||$.*

Since norms must be strictly positive, all variables must eventually perform an observable action. This is in analogy with the requirement in [BBK87a] that the grammars considered do not contain empty productions.

Because weak norms are assumed strictly positive, we have a simple relationship between lengths and norms:

Proposition 2.3 *For $\alpha \in Var^*$ we have $length(\alpha) \leq ||\alpha||$ and $||\alpha|| \leq m_\Delta length(\alpha)$.*

Finally, for weakly normed systems $\alpha \approx_b \beta$ implies that $||\alpha|| = ||\beta||$.

We also need the notion of *strong norm*:

Definition 2.3 *The* strong norm *of any $X \in Var$ is defined as*

$$|X| = \min\{length(w) \mid X \stackrel{w}{\rightarrow} \epsilon, w \in (Act \cup \{\tau\})^*\}$$

A system of defining equations Δ is strongly normed *if for any variable $X \in Var$ $|X| < \infty$.*

Clearly, Δ is strongly normed if it is weakly normed.

Finally, we restrict our attention to systems of defining equations given in 3-Greibach Normal Form (3-GNF). Δ is in 3-GNF if it is of the form $\{X_i \stackrel{\text{def}}{=} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij} \mid 1 \leq i \leq m\}$ where each variable sequence α_{ij} has length < 3. This is actually no real restriction, since any family Δ in $\text{BPA}^\tau_{\text{rec}}$ can effectively be rewritten to a family Δ' which is *strongly* bisimilar to Δ and therefore normed iff Δ is [BBK87b]. The restriction to variable sequences of length at most 2 guarantees limited growth when determining single transitions. When applying a defining equation to the leftmost variable in a string α the length of the derivative does not increase by more than 1.

Note that there is an obvious correspondence between defining systems given in 3-GNF and context-free grammars: variables correspond to non-terminals and actions correspond to terminal symbols. Equations $X_i \stackrel{\text{def}}{=} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij}$ correspond to the family of productions $\{X_i \rightarrow a_{ij}\alpha_{ij} \mid 1 \leq j \leq n_i\}$.

The class of normed $\text{BPA}^\tau_{\text{rec}}$ does not include all regular processes. However, we can now define processes with an infinite number of states; an example is the process given in the following example.

Example 2.1 Consider the processes given by $\Delta_1 = \{A \stackrel{\text{def}}{=} a + bBC;\ B \stackrel{\text{def}}{=} \tau A;\ C \stackrel{\text{def}}{=} c\}$ (as shown in Figure 1) and $\Delta_2 = \{X \stackrel{\text{def}}{=} a + bXY;\ Y \stackrel{\text{def}}{=} c\}$, respectively. Then $X \approx_b A$ because the relation $\{(XY^n, AC^n) | n \geq 0\} \cup \{(XY^n, BC^n) | n \geq 1\} \cup \{(Y^n, C^n) \mid n \geq 1\} \cup \{(\epsilon, \epsilon)\}$. $\square$

For the tableau system we need two important results on the relationship between branching bisimilarity and sequential composition. (The proofs utilize the notion of bb from Definition 2.1.) First of all, $\approx_b$ is a congruence w.r.t. sequential composition:

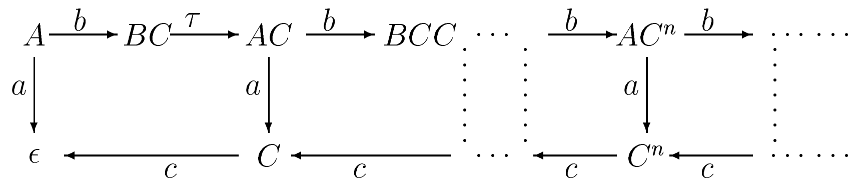

Figure 1: Transition graph for $A \stackrel{\text{def}}{=} a + bBC$; $B \stackrel{\text{def}}{=} \tau A$; $C \stackrel{\text{def}}{=} c$

Proposition 2.4 *If $\alpha_1 \approx_b \beta_1$ and $\alpha_2 \approx_b \beta_2$ then $\alpha_1\alpha_2 \approx_b \beta_1\beta_2$*

The next proposition allows us to discard identical tails from branching bisimilar expressions.

Proposition 2.5 *If $\alpha_1\alpha \approx_b \alpha_2\alpha$ then $\alpha_1 \approx_b \alpha_2$.*

Note that this does *not* hold for weak bisimulation. A counterexample is the family $\Delta = \{X = aY, Y = a + \tau X, A = a + aB, B = a\}$ which clearly obeys all requirements stated above. It is easily seen that $X \approx BY$ and that $A \not\approx B$. However, we have that $AY \approx BY$.

3 The tableau system

3.1 Building tableaux

A tableau for determining branching bisimilarity is a maximal proof tree built using the proof rules in Table 1. Tableaux are built from a series of subtableaux. A subtableau is in turn built from successive applications of the STEP rule . STEP is applicable iff there is a possibility of matching transitions. A *possible match* is any set of equations whose sides are the results of successful matching transitions according to the definition of branching bisimilarity in Proposition 2.1:

Definition 3.1 *A set of equations M is a* possible match *for $p = q$ if for any $a \in Act$ we have that if $p \stackrel{a}{\to} p'$ then either*

- *$a = \tau$ and $p' = q \in M$ or*
- *there exist $q_0' = q, \ldots, q_n', q'$ such that $q_0 \stackrel{\tau}{\to} q_1' \stackrel{\tau}{\to} \cdots \stackrel{\tau}{\to} q_n' \stackrel{a}{\to} q'$ with $p = q_i' \in M$ for $0 \le i \le n$ and $p' = q' \in M$.*

and similarly for any $q \stackrel{a}{\to} q'$.

This definition appears to allow *infinitely* many possible matches, since no bound is made on n above. However, since we are assuming 3-GNF, whenever $X\alpha \approx_b Y\beta$ and $X\alpha \xrightarrow{a} \alpha'\alpha$ is matched by $Y\beta \stackrel{\epsilon}{\Longrightarrow} \beta'_1\beta \xrightarrow{a} \beta'\beta$ the intermediate states in $Y\beta \stackrel{\epsilon}{\Longrightarrow} \beta'_1\beta$ must all have the same norm as $X\alpha$, so by Proposition 2.3 their lengths can increase by at most m_Δ. Similarly, the length of β' can increase at most m_Δ. Thus, all equations in any possible match for $X\alpha = Y\beta$ compare expressions whose lengths are bounded by $2m_\Delta + \max(length(\alpha), length(\beta))$. Let v be the cardinality of Var. Then there are at most $v^{2m_\Delta+\max(length(\alpha),length(\beta))+1}$ different equations for any transition $X\alpha \xrightarrow{a} \alpha'\alpha$ in a possible match. So if there is a possible match for $X\alpha = Y\beta$ we can always find one with at most $t_{X\alpha=Y\beta}v^{2m_\Delta+\max(length(\alpha),length(\beta))+1}$ equations, where $t_{X\alpha=Y\beta}$ is the cardinality of $\{\alpha'\alpha \mid X\alpha \xrightarrow{a} \alpha'\alpha\} \cup \{\beta'\beta \mid Y\beta \xrightarrow{a} \beta'\beta\}$.

Clearly, STEP is forwards sound in the sense that if the premise is valid, then there is a possible match all of whose equations are valid.

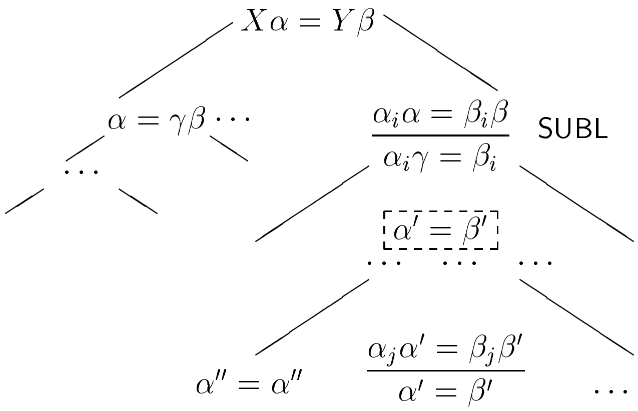

Figure 2: A tableau for $X\alpha = Y\beta$; some successful leaves are shown

An eliminating subtableau for $X\alpha = Y\beta$ consists of such attempted matches to the depth where an equation of the form $\alpha = \gamma\beta$ is reached. When $|X| \leq |Y|$ each leaf of an eliminating subtableau for $X\alpha = Y\beta$ is either labelled $\alpha = \gamma\beta$ (a *residual* of the subtableau) as X is eliminated, or $\alpha_i\alpha = \beta_i\beta$. Because the number of successive attempted matches is $|X|$ there must be at least one residual and α and β must persist as suffixes throughout the subtableau because all norms are strictly positive. For any such subtableau we pick one residual node and call it *the* residual. If instead $|Y| < |X|$ similar remarks apply, only then the residual has the form $\gamma\alpha = \beta$.

Unless a subtableau leaf is a successful terminal (see below) it is used as the basis of a new subtableau. However, before a new subtableau is constructed, for every leaf one of the SUB rules is used to trim the length of the expressions in the new subtableau root. See Figure 2. These rules are forward sound in that if

$\alpha_i\alpha \approx_b \beta_i\beta$ and $\alpha \approx_b \gamma\beta$ then we must have $\alpha_i\gamma \approx_b \beta_i$ (and similarly for SUBR). This follows from Propositions 2.4 and 2.5.

Rule within a subtableau

STEP $$\frac{\alpha = \beta}{\alpha_1 = \beta_1 \ldots \alpha_k = \beta_k}$$ where $\{\alpha_1 = \beta_1 \ldots \alpha_k = \beta_k\}$ is a possible match for $\alpha = \beta$

Rules for new subtableaux

SUBL $$\frac{\alpha_i\alpha = \beta_i\beta}{\alpha_i\gamma = \beta_i}$$ where $\alpha = \gamma\beta$ is the residual

SUBR $$\frac{\alpha_i\alpha = \beta_i\beta}{\alpha_i = \beta_i\gamma}$$ where $\gamma\alpha = \beta$ is the residual

---

Table 1: The tableau rules

The rules are only applied to nodes that are not *terminal*. Terminal nodes can either be *successful* or *unsuccessful*.

Definition 3.2 *A tableau node is an* unsuccessful terminal *if it has one of the forms*

1. $\alpha = \beta$ *with* $||\alpha|| \neq ||\beta||$
2. $\alpha = \beta$ *with* $\alpha \neq \epsilon, \beta \neq \epsilon$ *and no possible match exists (i.e.* STEP *is inapplicable).*

Clearly, in both of the above cases we cannot have that the expressions compared are bisimilar. Thus, if we see an unsuccessful terminal the whole tableau construction aborts.

The nodes that can be successful terminals are those that are potential root of eliminating subtableaux:

Definition 3.3 *A residual or consequent of an application of a* SUB *rule is a* successful terminal *if it has one of the forms*

1. *$\alpha = \beta$ where there is another subtableau root above it on the path from the root also labelled $\alpha = \beta$ or $\beta = \alpha$.*

2. *$\alpha = \alpha$*

$$\dfrac{X = A}{\epsilon = \epsilon \qquad \dfrac{\dfrac{XY = BC}{XY = BC}\ \mathsf{SUBL}}{\dfrac{XY = AC}{X = A}\ \mathsf{SUBL} \quad \dfrac{XYY = BCC}{XY = BC}\ \mathsf{SUBL} \quad \dfrac{Y = C}{\epsilon = \epsilon}\ \mathsf{STEP}}\ \mathsf{STEP}}\ \mathsf{STEP}$$

Figure 3: A successful tableau for $X = A$

Example 3.0 (cont.) The tableau in Figure 3 is a successful tableau for $X = A$. $\square$

3.2 Termination, completeness, and soundness

It is important for our decidability result that all tableaux are finite. Since our tableaux are finitely branching, by König's Lemma an infinite tableau would have an infinite path. This would then be caused by the combined absence of unsuccessful termination and the successful termination condition 1 along that path.

Since we have assumed 3-GNF and normedness, there is a uniform bound on the total length of the consequent of a SUB rule. Assume wlog that we have a SUBL rule:

$$\dfrac{X\alpha_1\alpha = Y\beta_1\beta}{X\alpha_1\gamma = Y\beta_1}\quad \mathsf{SUBL}$$

Since $|X| \leq |Y|$ we have that α_1 has its total length bounded by $m_\Delta + 1$. Since $||\gamma|| = ||X|| - ||Y||$, by Proposition 2.3 the length of γ is bounded by $m_\Delta - 1$. As pointed out in the previous section, whenever $X\alpha \xrightarrow{a} \alpha'\alpha$ is matched by $Y\beta \xRightarrow{\epsilon} \beta_1'\beta \xrightarrow{a} \beta'\beta$, the intermediate states in $Y\beta \xrightarrow{\epsilon} \beta_1'\beta$ must have the same norm as $X\alpha$, so their lengths can increase by at most m_Δ and similarly, the length of β' can increase by at most m_Δ. Thus, in the above we have $length(\beta_1) \leq 2m_\Delta(m_\Delta{+}1)$ and the total length of a SUB consequent is bounded by $4m_\Delta^2{+}6m_\Delta$.

Thus there is no infinite path through infinitely many SUB applications since there are of course only finitely many different equations of any given length.

Nor can an infinite path pass through infinitely many residuals. For if a residual $\alpha_0 = \beta_0$ is above the residual $\alpha_1 = \beta_1$ we have that $||\alpha_0|| = ||\beta_0|| < ||\alpha_1|| = ||\beta_1||$. By Proposition 2.3, any subsequence of residuals therefore has a uniform bound on the total lengths of expressions compared, again ensuring termination.

In other words

Theorem 3.1 *For any equation $\alpha = \beta$ all tableaux are finite.*

It is easily seen that the tableau system is complete in that valid equations always give rise to some successful tableau, because of the forwards soundness of the SUB rules and because STEP then always is applicable and can be used in such a way that only valid consequents arise.

Theorem 3.2 *If $\alpha \approx_b \beta$, $\alpha = \beta$ has a successful tableau.*

Finally we must show soundness of the tableau system, namely that the existence of a successful tableau for $\alpha = \beta$ indicates that $\alpha \approx_b \beta$. This follows from the fact that the tableau system tries to construct an 'essential part of a branching bisimulation', namely a *branching bisimulation up to sequential congruence* that contains the root, a notion originating in work by Caucal [Cau88]. If the tableau is successful this relation exists and is *finite*, consisting of all pairs compared in the tableau. Here the notion of the least congruence under sequential composition becomes important:

Definition 3.4 *The* least precongruence *under sequential composition of a relation R, $\underset{R}{\rightarrow}$, is defined by $p \underset{R}{\rightarrow} q$ whenever $p = sp't, q = sq't$ and $p'Rq'$. The* symmetric closure *of $\underset{R}{\rightarrow}$ is denoted by $\underset{R}{\longleftrightarrow}$. The* least congruence under sequential composition *$\underset{R}{\longleftrightarrow}^*$ is the reflexive and transitive closure of $\underset{R}{\longleftrightarrow}$.*

We need a simple rephrasing of Proposition 2.1 here:

Proposition 3.1 *A branching bisimulation on Γ is a symmetric relation $R \subseteq Pr \times Pr$ such that whenever pRq for any $a \in Act \cup \{\tau\}$ we have that if $p = p_0 \xrightarrow{\tau} p_1 \xrightarrow{\tau} \cdots p_m \xrightarrow{a} p'$ then there exist $q_0, q_1, \ldots, q_m, q'$ such that $q_0 = q$ and $p_i R q_i$ for $1 \leq i \leq m$, $p'Rq'$ and for $i < m$*

- – *$q_i = q_{i+1}$ or*
  – *there exist $q_{i_1}, \ldots, q_{i_{n(i)}}$ such that $q_i \xrightarrow{\tau} q_{i_1} \xrightarrow{\tau} \cdots q_{i_{n(i)}} \xrightarrow{\tau} q_{i+1}$ with $p_i R q_{ij}$ for $1 \leq j \leq n(i)$*

and either

- $a = \tau$ *and* $q_m = q'$ *or*
- *there exist* $q_{m_1}, \ldots, q_{m_{n(m)}}$ *such that* $q_m \xrightarrow{\tau} q_{m_1} \xrightarrow{\tau} \cdots q_{m_{n(m)}} \xrightarrow{a} q'$

PROOF:Induction in n. $\square$

We now have

Definition 3.5 *A* branching bisimulation up to sequential congruence (sbb) *on* Γ *is a symmetric relation* $R \subseteq Pr \times Pr$ *such that whenever* pRq *for any* $a \in Act \cup \{\tau\}$ *we have that if* $p = p_0 \xrightarrow{\tau} p_1 \xrightarrow{\tau} \cdots p_m \xrightarrow{a} p'$ *then there exist* $q_0, q_1, \ldots, q_m, q'$ *such that* $q_0 = q$ *and* $p_i \underset{R}{\longleftrightarrow}^* q_i$ *for* $1 \leq i \leq m$*,* $p' \underset{R}{\longleftrightarrow}^* q'$ *and for* $i < m$

-
  - $q_i = q_{i+1}$ *or*
  - *there exist* $q_{i_1}, \ldots, q_{i_{n(i)}}$ *such that* $q_i \xrightarrow{\tau} q_{i_1} \xrightarrow{\tau} \cdots q_{i_{n(i)}} \xrightarrow{\tau} q_{i+1}$ *with* $p_i \underset{R}{\longleftrightarrow}^* q_{ij}$ *for* $1 \leq j \leq n(i)$

and either

- $a = \tau$ *and* $q_m = q'$ *or*
- *there exist* $q_{m_1}, \ldots, q_{m_{n(i)}}$ *such that* $q_m \xrightarrow{\tau} q_{m_1} \xrightarrow{\tau} \cdots q_{m_{n(m)}} \xrightarrow{a} q'$ *with* $p_m \underset{R}{\longleftrightarrow}^* q_{mj}$ *for* $1 \leq j \leq n(m)$*.*

The reason why a bisimulation up to sequential congruence can be said to be an essential part of a bisimulation lies in the following result which is our version of a result used by Caucal [Cau88]:

Lemma 3.1 *If* R *is an sbb then* $\underset{R}{\longleftrightarrow}^*$ *is a bb.*

Corollary 3.1 $\alpha \approx_b \beta$ *iff there is an sbb* R *such that* $\alpha R \beta$*.*

PROOF: From the above and from the fact that any bb is an sbb. $\square$

We now have

Theorem 3.3 *If* $\alpha = \beta$ *has a successful tableau* $\mathbf{T}$ *then*

$$R_{\mathbf{T}} = \{(\alpha', \beta') \mid \alpha' = \beta' \text{ or } \beta' = \alpha' \text{ is an equation in } \mathbf{T}\}$$

is an sbb.

So we now get the soundness of the tableau system as

Corollary 3.2 *If* $\alpha = \beta$ *has a successful tableau then* $\alpha \approx_b \beta$*.*

3.3 Complexity of the tableau system and decidability of $\approx_b$

The complexity of the tableau system can be measured in terms of the maximal depth of a tableau, i.e. the length w.r.t. STEP applications of the longest possible path in a successful tableau for an equation $X\alpha = Y\beta$. Let v be the cardinality of Var. By the discussion preceding Theorem 3.1 we have that any SUB consequent has a length of at most $4m^2+6m$, so there can be at most $(2m_\Delta^2+3m_\Delta)v^{4m_\Delta^2+6m_\Delta}$ distinct SUB consequents along any tableau path. Between any two SUB consequents there are at most $(4m_\Delta^2+6m_\Delta)$ residuals, so any path that contains SUB consequents can have at most $(4m_\Delta^2+6m_\Delta)(2m_\Delta^2+3m_\Delta)v^{4m_\Delta^2+6m_\Delta}$ subtableau roots. As for the leftmost path, all of whose subtableau roots are residuals, there can be at most $\max(||\alpha||, ||\beta||)$ residuals, since the norm of the residuals is strictly decreasing. So, since a subtableau can have a depth w.r.t. STEP applications of at most $m_\Delta T_\Delta$, any path can have a length of at most

$$\max(||\alpha||, ||\beta||, (4m_\Delta^2+6m_\Delta)(2m_\Delta^2+3m_\Delta)v^{4m_\Delta^2+6m_\Delta}m_\Delta T_\Delta$$

STEPs.

Since we thus have an upper bound on the depth of a tableau, and because we have assumed image-finiteness, we see that any equation has finitely many possible tableaux. A naive decision procedure for $\approx_b$ consists in constructing all these finitely many tableaux, answering 'yes' if a successful tableau occurs and 'no' otherwise. Thus

Theorem 3.4 *For any normed, convergent, and image-finite Δ it is decidable whether or not $\alpha \approx_b \beta$ for $\alpha, \beta \in Var^*$.*

4 Conclusions and directions for further work

We have here shown that the branching bisimilarity of [vGW89] is decidable for the class of normed BPA processes with silent actions by giving a tableau system. This system has exponential complexity in terms of the longest possible path of a generated tableau; however, in the case of a successful tableau we get additional information in the form of a finite relation whose congruence closure w.r.t. sequential composition is a bisimulation containing the initial equation.

There are still several interesting open problems.

The restriction to processes with strictly positive norms is rather strong, as it rules out the possibility of a process terminating silently. A problem with having nullary norms in the tableau system is that we no longer are guaranteed that α and β persist throughout an eliminating subtableau for $X\alpha = Y\beta$, since a match for $X\alpha \xrightarrow{a}$ may require access to observable actions inside β. So the question is: is there a way of introducing nullary norms ? Moreover, it would be nice if we could get rid of the restriction of normedness of any kind. In particular: What happens if we allow infinite norms ? This problem is still open for strong bisimilarity, so it seems that progress must first be made here before we can give any answer for the branching bisimulation case.

Moreover, it might be interesting if we could give a syntax-directed version of our tableau system since this may give us an equational theory of $\approx_b$ along the lines of [HS91].

Last, but not least the questions for weak bisimilarity all remain open. As we saw, Proposition 2.5 does not hold for this equivalence so a different approach must be used in that case.

References

[BBK87a] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. In *LNCS 259*, pages 93–114. Springer-Verlag, 1987.

[BBK87b] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. Technical Report CS-R8632, CWI, September 1987.

[BCG88] M.C. Browne, E.M. Clarke, and O. Grumberg. Characterizing Kripke structures in temporal logic. *Theoretical Computer Science*, 51:115–131, 1988.

[BK88] J.A. Bergstra and J.W. Klop. Process theory based on bisimulation semantics. In J.W. de Bakker, W.P de Roever, and G. Rozenberg, editors, *LNCS 354*, pages 50–122. Springer-Verlag, 1988.

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

[HS91] H. Hüttel and C. Stirling. Actions speak louder than words: Proving bisimilarity for context-free processes. In *Proceedings of 6th Annual Symposium on Logic in Computer Science (LICS 91)*, pages 376–386. IEEE Computer Society Press, 1991.

[KH66] A.J. Korenjak and J.E. Hopcroft. Simple deterministic languages. In *Proceedings of Seventh Annual IEEE Symposium on Switching and Automata Theory*, pages 36–46, 1966.

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

[vG90] R.J. van Glabbeek. *Comparative Concurrency Semantics and Refinement of Actions*. PhD thesis, CWI/Vrije Universiteit, 1990.

[vGW89] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics (extended abstract). In G.X. Ritter, editor, *Information Processing 89*, pages 613–618. North-Holland, 1989.