Bisimulation Equivalence is Decidable for all Context-Free Processes

Søren Christensen* Hans Hüttel† Colin Stirling*

1 Introduction

Over the past decade much attention has been devoted to the study of process calculi such as CCS, ACP and CSP [13]. Of particular interest has been the study of the behavioural semantics of these calculi as given by labelled transition graphs. One important question is when processes can be said to exhibit the same behaviour, and a plethora of *behavioural equivalences* exists today. Their main rationale has been to capture behavioural aspects that language or trace equivalences do not take into account.

The theory of finite-state systems and their equivalences can now be said to be well-established. There are many automatic verification tools for their analysis which incorporate equivalence checking. Sound and complete equational theories exist for the various known equivalences, an elegant example is [18].

One may be led to wonder what the results will look like for *infinite-state* systems. Although language equivalence is decidable for finite-state processes, it is undecidable when one moves beyond finite automata to context-free languages. For finite-state processes all known behavioural equivalences can be seen to be decidable. In the setting of process algebra, an example of infinite-state systems is that of the transition graphs of processes in the calculus BPA (Basic Process Algebra) [4]. These are recursively defined processes with nondeterministic choice and sequential composition.

A special case is that of *normed BPA* processes. A process is said to be normed if it can terminate in finitely many steps at any point during the execution. Even though normed BPA does not incorporate all regular processes, systems defined in this calculus can in general have infinitely many states.

In [1, 2] Baeten, Bergstra and Klop proved the remarkable result that bisimulation equivalence is decidable on the class of *normed* context-free processes.

*Laboratory for Foundations of Computer Science, James Clerk Maxwell Building, University of Edinburgh, Edinburgh EH9 3JZ, Scotland.

†Department of Mathematics and Computer Science, Aalborg University Centre, Fredrik Bajersvej 7E, 9220 Aalborg Ø, Denmark.

Their proof is rather lengthy and hard to grasp; it ultimately relies on showing a periodicity for any transition graph generated from normed context-free processes. Caucal presented in [8] a more elegant (and shorter) proof of the same result utilising rewrite techniques. Finally, in [16] Hüttel and Stirling presented yet another proof of the decidability result by appealing to the tableau method. The tableau based approach also supports a sound and complete sequent based equational theory for normed context-free processes (see [16, 15]).

One remaining question to be answered is whether bisimulation equivalence is decidable for the full class of context-free processes. We here answer this question in the affirmative, using a technique inspired by Caucal's proof of the decidability of language equivalence for simple algebraic grammars (see [6]).

In the first section we introduce an alternative characterisation of bisimulation equivalence, namely via a sequence of approximations, which will enable us to conclude semi-decidability of bisimulation inequivalence on the class of guarded context-free processes. Thus we only need to consider semi-decidability of bisimulation equivalence in order to establish our result. This is achieved in the following (and final) section through a finite representability result; here the emphasis is on *decomposition* of pairs of bisimilar processes into "smaller" pairs of bisimilar processes such that only finitely many interesting pairs of bisimilar processes cannot be decomposed further.

2 BPA processes

The class of recursive BPA (Basic Process Algebra) processes [1, 4] is defined by the following abstract syntax

$$E ::= a \mid X \mid E_1 + E_2 \mid E_1 \cdot E_2$$

Here a ranges over a set of atomic actions Act, and X over a family of variables. The operator $+$ is nondeterministic choice while $E_1 \cdot E_2$ is the sequential composition of E_1 and E_2 – we usually omit the '$\cdot$'. In the operational semantics that follows, we shall also need to refer to the *empty process* ϵ – this process cannot occur in a BPA process definition and is thus not mentioned in the syntax. A BPA process is defined by a finite system of recursive process equations

$$\Delta = \{X_i \stackrel{\text{def}}{=} E_i \mid 1 \leq i \leq k\}$$

where the X_i are distinct, and the E_i are BPA expressions with free variables in $Var_\Delta = \{X_1, \ldots, X_k\}$. In a process definition, one variable (generally X_1) is singled out as the *root*. Usually one considers relations within the transition graph for a single Δ. This can be done without loss of generality, since we can let Δ be the disjoint union of any pair Δ_1 and Δ_2 that we wish to compare (with suitable renamings of variables, if required).

We restrict our attention to *guarded* systems of recursive equations.

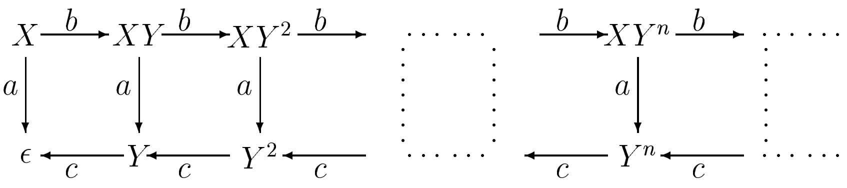

Figure 1: Transition graph for $X \stackrel{\text{def}}{=} a + bXY; \quad Y \stackrel{\text{def}}{=} c$ (Example 2.1)

Definition 2.1 *A BPA expression is* guarded *if every variable occurrence is within the scope of an atomic action. The system* $\Delta = \{X_i \stackrel{\text{def}}{=} E_i \mid 1 \leq i \leq k\}$ *is guarded if each* E_i *is guarded for* $1 \leq i \leq k$*.*

We use $X, Y, \ldots$ to range over variables in Var_Δ and Greek letters $\alpha, \beta, \ldots$ to range over elements in Var_Δ^*. In particular, ϵ denotes the empty variable sequence.

Definition 2.2 *Any system of process equations* Δ *defines a labelled transition graph. The transition relations are given as the least relations satisfying the following rules:*

$$a \stackrel{a}{\to} \epsilon, a \in Act \qquad \frac{E \stackrel{a}{\to} E'}{X \stackrel{a}{\to} E'} \; X \stackrel{\text{def}}{=} E \in \Delta$$

$$\frac{E \stackrel{a}{\to} E'}{E + F \stackrel{a}{\to} E'} \qquad \frac{F \stackrel{a}{\to} F'}{E + F \stackrel{a}{\to} F'}$$

$$\frac{E \stackrel{a}{\to} E'}{EF \stackrel{a}{\to} E'F} \; E' \neq \epsilon \qquad \frac{E \stackrel{a}{\to} \epsilon}{EF \stackrel{a}{\to} F}$$

Example 2.1 Consider the system $\Delta = \{X \stackrel{\text{def}}{=} a + bXY; \quad Y \stackrel{\text{def}}{=} c\}$. By the transition rules in Definition 2.2 X generates the transition graph in Figure 1. $\Box$

2.1 Bisimulation equivalence and Greibach Normal Form

Definition 2.3 *A relation* R *between processes is a* bisimulation *if whenever* pRq *then for each* $a \in Act$

1. $p \stackrel{a}{\to} p' \Rightarrow \exists q' : q \stackrel{a}{\to} q'$ *with* $p'Rq'$
2. $q \stackrel{a}{\to} q' \Rightarrow \exists p' : p \stackrel{a}{\to} p'$ *with* $p'Rq'$

Two processes p and q are *bisimulation equivalent*, written as $p \sim q$, if there is a bisimulation relation R such that pRq. The relation $\sim$ is an equivalence, and moreover it is a congruence relation with respect to the operators $+$ and $\cdot$, [4]. An alternative characterization of $\sim$ is via a sequence of approximations.

Definition 2.4 *The sequence of bisimulation approximations $\{\sim_n\}_{n=1}^{\omega}$ is defined inductively as follows.*

- $p \sim_0 q$ *for all processes* p *and* q*,*
- $p \sim_{n+1} q$ *iff for each* $a \in Act$
  - $p \xrightarrow{a} p'$ *implies* $q \xrightarrow{a} q'$ *and* $p' \sim_n q'$ *for some* q'
  - $q \xrightarrow{a} q'$ *implies* $p \xrightarrow{a} p'$ *and* $p' \sim_n q'$ *for some* p'

It is a standard result, see [19] for instance, that for any image-finite labelled transition graph (that is, where for each p and a the set $\{q \mid p \xrightarrow{a} q\}$ is finite):

$$\sim \; = \bigcap_{n=0}^{\omega} \sim_n$$

Clearly, the transition graph for any family Δ of guarded BPA processes is image-finite.

Any system Δ of guarded BPA equations has a unique solution up to bisimulation equivalence [3]. Moreover, in [1] it is shown that any such system can be effectively presented in a normal form

$$\{X_i \stackrel{\mathrm{def}}{=} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij} \mid 1 \leq i \leq m\}$$

such that bisimilarity is preserved. From the transition rules we see that if $X_i \xrightarrow{w} E$ then E is just a sequence α of variables. The normal form is called Greibach Normal Form, GNF, by analogy with context-free grammars (without the empty production) in GNF (see e.g. [14]). There is an obvious correspondence with grammars in GNF: process variables correspond to non-terminals, the root is the start symbol, actions correspond to terminals, and each equation $X_i \stackrel{\mathrm{def}}{=} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij}$ can be viewed as the family of productions $\{X_i \to a_{ij}\alpha_{ij} \mid 1 \leq j \leq n_i\}$.

3 Decidability of bisimulation equivalence

Assume a fixed system Δ of BPA equations in GNF whose variable set is Var. The bisimulation equivalence problem is whether or not $\alpha \sim \beta$ when α and β are sequences of variables drawn from Var. In the case that these are finite-state processes, a very naive decision procedure consists of enumerating all binary relations over the finite state space generated by α and β using the rules for transitions and determining if there is a relation among them which is a bisimulation containing the pair (α, β). But of course BPA processes are not generally finite-state, and therefore bisimulations can now be infinite.

On the other hand for any n, the n-bisimulation equivalence problem (whether or not $\alpha \sim_n \beta$) is decidable. This means that bisimulation inequivalence is semi-decidable via the simple procedure which seeks the least i such that $\alpha \not\sim_i \beta$. Therefore we just need to establish the semi-decidability of bisimulation equivalence. The proof of this (inspired by [6, 7, 8]) relies on showing that there is a *finite self-bisimulation* relation which generates the bisimulation equivalence.

3.1 Self-bisimulations

The notion of self-bisimulation was introduced by Didier Caucal in [8] (originally published as [7]). Here the notion of a least congruence is essential.

Definition 3.1 *For any binary relation R on Var^*, $\underset{R}{\rightarrow}$ is the least precongruence w.r.t. sequential composition that contains R, $\underset{R}{\longleftrightarrow}$ the symmetric closure of $\underset{R}{\rightarrow}$ and $\underset{R}{\longleftrightarrow}^*$ the reflexive and transitive closure of $\underset{R}{\longleftrightarrow}$ and thus the least congruence w.r.t. sequential composition containing R.*

A self-bisimulation is then simply a bisimulation up to congruence w.r.t. sequential composition.

Definition 3.2 *A relation $R \subseteq Var^* \times Var^*$ is called* a self-bisimulation *iff $\alpha R \beta$ implies that for each $a \in Act$*

1. *if $\alpha \xrightarrow{a} \alpha'$ then $\beta \xrightarrow{a} \beta'$ for some β' with $\alpha' \underset{R}{\longleftrightarrow}^* \beta'$*
2. *if $\beta \xrightarrow{a} \beta'$ then $\alpha \xrightarrow{a} \alpha'$ for some α' with $\alpha' \underset{R}{\longleftrightarrow}^* \beta'$*

The following lemma, due to Didier Caucal, shows that a self-bisimulation is a witness for bisimulation equivalence.

Lemma 3.1 [8] *If R is a self-bisimulation then $\underset{R}{\longleftrightarrow}^* \subseteq \sim$.*

Corollary 3.1 *$\alpha \sim \beta$ iff there is a self-bisimulation R such that $\alpha R \beta$.*

3.2 Decompositions

Our aim is to show that bisimulation equivalence on Var^* is generable from a finite self-bisimulation. To do this we must find techniques for decomposing bisimilar sequences of variables α and β into "smaller" subsequences $\alpha_1 \dots \alpha_n$ and $\beta_1 \dots \beta_n$ with $\alpha_i \sim \beta_i$ for each i in such a way that there are only "finitely" many pairs α and β that can not be decomposed. Extra definitions and some preliminary results are needed to achieve this.

A process $\alpha \in Var^+$ is *normed* if there is a $w \in Act^+$ such that $\alpha \xrightarrow{w} \epsilon$. When α is normed we let the *norm* of α, written as $|\alpha|$ following [1], be defined as:

$$|\alpha| = \min\{length(w) \mid \alpha \xrightarrow{w} \epsilon, w \in Act^+\}$$

By convention we also assume that $|\epsilon| = 0$. Clearly α is normed just in case each variable occurring in it has a norm. We divide the variable set Var into disjoint subsets $V_{fin} = \{X \in Var \mid X \text{ is normed}\}$ and $V_\infty = Var \setminus V_{fin}$. The system of equations, example 2.1, only contains normed variables with $|X| = |Y| = 1$, so $V_{fin} = \{X, Y\}$ and $V_\infty = \emptyset$. Example 3.1 contains an unnormed X so in this case $V_{fin} = \{Y\}$ and $V_\infty = \{X\}$.

Example 3.1 In the system of equations $\Delta = \{X \stackrel{\text{def}}{=} aX; \quad Y \stackrel{\text{def}}{=} c + aX\}$ the variable X is not normed since there is no w such that $X \xrightarrow{w} \epsilon$ whereas $|Y| = 1$. $\square$

A straightforward consequence of the definition of having a norm is the following:

$$\text{if } \; X \in V_\infty \; \text{ then } \; \alpha X \beta \sim \alpha X$$

Therefore we can assume that our fixed system of BPA equations in normal form $\Delta = \{X_i \stackrel{\text{def}}{=} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij} \mid 1 \leq i \leq m\}$ has the property that each $\alpha_{ij} \in (V_{fin}^* V_\infty) \cup V_{fin}^*$.

The next definition stipulates what we mean by decomposition:

Definition 3.3 *When $X\alpha \sim Y\beta$ we say that the pair $(X\alpha, Y\beta)$ is decomposable if $X, Y \in V_{fin}$ and there is a γ such that*

- *$\alpha \sim \gamma\beta$ and $X\gamma \sim Y$ if $|X| \leq |Y|$*
- *$\gamma\alpha \sim \beta$ and $X \sim Y\gamma$ if $|Y| \leq |X|$.*

In the case of normed processes (where each variable in Var is normed) the important property underpinning decidability of $\sim$ is that any bisimilar pair $(X\alpha, Y\beta)$ is decomposable (see [6]). Assuming that $|X| \leq |Y|$ and that β is not empty this means that there is a decomposition of $X\alpha$ into the two smaller (with respect to norm) subsequences $X\gamma$ and β with $X\gamma \sim Y$. Consequently bisimulation equivalence is then generable from a finite self-bisimulation consisting of pairs of the form (X, α).

However, in the presence of unnormed variables the situation is much more complex, as there can be bisimilar pairs $(X\alpha, Y\beta)$ which are not decomposable. We therefore need to show that in some sense there are only finitely many of them. A special class of pairs have the form $(\alpha, X\gamma\alpha)$. The following lemma provides some information about them.

Lemma 3.2 *If $\alpha \sim X\gamma\alpha$ and $\beta \sim X\gamma\beta$ then $\alpha \sim \beta$.*

PROOF: If $\alpha \sim X\gamma\alpha$ and $\beta \sim X\gamma\beta$ then both α and β are solutions to the same (guarded) equation. As any system of guarded equations has a unique solution up to bisimulation equivalence [1] it follows that $\alpha \sim \beta$. □

Let us call a $\phi \in Var^*$ a *unifier* for $\alpha, \beta \in Var^*$ if $\alpha \not\sim \beta$ but $\alpha\phi \sim \beta\phi$. Intuitively, a unifier repairs a bisimulation error by introducing a tail of infinite transitions.

Example 3.2 Consider the system $\Delta = \{X_1 = aX_2 + aX_3; X_2 = b; X_3 = c; X_4 = aX_2 + bX_3 + a; Y = bY; Z = cZ\}$. Clearly $X_1 \not\sim X_4$. However, Y and Z are unifiers of the pair (X_1, X_4) as $X_1Y \sim X_4Y$ and $X_1Z \sim X_4Z$. It is not difficult to check that any other unifier must be bisimilar to either Y or Z. □

We now present a crucial lemma which shows that there can only be finitely many different unifiers for any pair of non-bisimilar processes. For an arbitrary pair of non-bisimilar processes we do not know the upper bound on the number of such unifiers. However if we know that the pair is not in approximation relation $\sim_n$ then there is a bound which depends on the degree of Δ, $deg(\Delta)$, defined as the size of the largest set $\{\alpha |\ X \xrightarrow{a} \alpha,\ a \in Act\}$ when $X \in Var$: for instance, both systems of equations in Examples 2.1 and 3.1 have degree 2.

Lemma 3.3 *For any $\alpha, \beta \in Var^*$, if $\alpha \not\sim_n \beta$ then there are at most $(deg(\Delta))^{n-1}$ different unifiers up to $\sim$.*

PROOF: Induction on n using the previous lemma. For the base case if $\alpha \not\sim_1 \beta$ then without loss of generality $\alpha \xrightarrow{a}$ but $\beta \not\xrightarrow{a}$ for some a. But there can not be unifier ϕ giving $\alpha\phi \sim \beta\phi$, unless $\beta = \epsilon$. By Lemma 3.2 there is only one ϕ up to $\sim$ such that $\alpha\phi \sim \phi$. If $\alpha \not\sim_{n+1} \beta$ then without loss of generality $\alpha \xrightarrow{b} \alpha'$ and for all β' such that $\beta \xrightarrow{b} \beta'$ it is the case that $\alpha' \not\sim_n \beta'$. Now suppose $\alpha\phi \sim \beta\phi$. Any transition $\alpha\phi \xrightarrow{b} \alpha'\phi$ can be matched by a transition $\beta\phi \xrightarrow{b} \beta'\phi$ with $\alpha'\phi \sim \beta'\phi$. By the induction hypothesis there are only $(deg(\Delta))^{n-1}$ distinct ϕ_1 such that

$\alpha'\phi_1 \sim \beta'\phi_1$. Let $S = \{a_j\alpha'_j \mid \alpha \xrightarrow{a_j} \alpha'_j, \forall\beta' : \beta \xrightarrow{a_j} \beta' \Rightarrow \alpha_j \not\sim_n \beta'\}$. S can have at most $deg(\Delta)$ distinct elements. We can write $\alpha \sim \sum_{a_j\alpha_j \in S} a_j\alpha_j + \sum_{b_j\beta_j \notin S} b_j\beta_j$, and from this we see that since for each $a_j\alpha_j \in S$ there are at most $(deg(\Delta))^{n-1}$ distinct unifiers ϕ such that $\alpha_j\phi \sim \beta'\phi$, there are all in all at most $(deg(\Delta))^n$ distinct unifiers ϕ such that $\alpha\phi \sim \beta\phi$. □

We say that the pairs $(X\alpha, Y\beta)$ and $(X\alpha_1, Y\beta_1)$ are *distinct* when $\alpha \not\sim \alpha_1$ or $\beta \not\sim \beta_1$. The next surprising result shows that there are only finitely many interesting pairs $(X\alpha, Y\beta)$ that are not decomposable.

Lemma 3.4 *For any $X, Y \in Var$ any set R of the form*

$$\{(X\alpha, Y\beta) \mid X\alpha, Y\beta \in (V^*_{fin}V_\infty) \cup V^*_{fin},\ X\alpha \sim Y\beta,\ (X\alpha, Y\beta) \text{ is not decomposable}\}$$

such that all pairs are distinct is finite.

Proof: First, if both X and Y belong to V_∞ then R contains just one member. Otherwise assume only one of them is in V_∞, without loss of generality let this be X. As Y is normed let $|Y| = n$. Therefore $Y \xrightarrow{w} \epsilon$ for some w of length n. But there are only finitely many γ such that $X \xrightarrow{w} \gamma$. If R were infinite containing pairs $(X, Y\beta_i)$ for all i, then as every β_i should be bisimilar to some γ, we would have that for some j $\beta_j \sim \beta_k$ for infinitely many k. But this would contradict distinctness. There can thus only be finitely many β_i such that $X \sim Y\beta_i$. Now, assume that both $X, Y \in V_{fin}$ and without loss of generality let $|X| \leq |Y|$ with $|X| = n$. Consider a $w = a_1 \cdots a_n$ such that $X \xrightarrow{w} \epsilon$. Since $X\alpha_i \sim Y\beta_i$ for all $(X\alpha_i, Y\beta_i) \in R$ we must have $Y \xrightarrow{w} \gamma$ for some γ. But then consider the set $B = \{\gamma_j \mid \exists u : Y \xrightarrow{u} \gamma_j, length(u) = n\}$. This set is finite and has at most $(deg(\Delta))^n$ elements. But then for some $\gamma \in B$, since $\alpha_i \sim \gamma\beta_i$, it must be the case that for infinitely many $(X\alpha_i, Y\beta_i) \in R$ we have $X\gamma\beta_i \sim Y\beta_i$. But this would imply an unbounded number of unifiers for $(X\gamma, Y)$ and this is impossible by Lemma 3.3, as $X\gamma \not\sim Y$ follows from the assumption that the pairs are not decomposable. □

3.3 Finite representability of $\sim$

We are now almost in a position to prove our main theorem, which relies on an induction on size, defined for every $\alpha \in (V^*_{fin}V_\infty) \cup V^*_{fin}$ and denoted by $s(\alpha)$:

$$s(\alpha X) = \begin{cases} |\alpha X| & \text{if } X \in V_{fin} \\ |\alpha| & \text{otherwise} \end{cases}$$

We let $\sqsubseteq$ be the well-founded ordering on $(V^*_{fin}V_\infty) \cup V^*_{fin} \times (V^*_{fin}V_\infty) \cup V^*_{fin}$ given by $(\alpha_1, \alpha_2) \sqsubseteq (\beta_1, \beta_2)$ if $\max\{s(\alpha_1), s(\alpha_2)\} \leq \max\{s(\beta_1), s(\beta_2)\}$.

Theorem 3.1 *There is a finite relation R on $(V_{fin}^* V_\infty) \cup V_{fin}^*$ such that* $\sim \, = \, \underset{R}{\longleftrightarrow}^*$.

PROOF: We define R as the union of two finite relations R_1 and R_2. R_1 is a largest relation of the form

$$\{(X, \alpha) \mid X, \alpha \in V_{fin}^*, \; X \sim \alpha\}$$

and R_2 is a largest relation of the form

$$\{(X\alpha, Y\beta) \mid X\alpha, Y\beta \in (V_{fin}^* V_\infty) \cup V_{fin}^*, \; X\alpha \sim Y\beta, \; (X\alpha, Y\beta) \text{ is not decomposable}\}$$

such that each pair $(X\alpha, Y\beta)$, $(X\alpha', Y\beta')$ in R_2 is distinct. Moreover, we assume *minimal* elements w.r.t. $\sqsubseteq$, i.e. if $(X\alpha, Y\beta) \in R_2$ is *not* distinct from $(X\alpha', Y\beta')$ then $(\alpha, \beta) \sqsubseteq (\alpha', \beta')$. Notice that both R_1 and R_2 are finite; the finiteness of R_1 follows from the fact that there are only finitely many elements of V_{fin}^* with a given finite norm and the finiteness of R_2 follows from lemma 3.4. Thus R is finite.

We now want to show that $\sim \, = \, \underset{R}{\longleftrightarrow}^*$. As $R \subseteq \, \sim$ and $\sim$ is a congruence w.r.t. sequential composition we immediately have $\underset{R}{\longleftrightarrow}^* \subseteq \, \sim$. So we consider proving $\sim \, \subseteq \, \underset{R}{\longleftrightarrow}^*$ and proceed by induction on $\sqsubseteq$. Let $X\alpha \sim Y\beta$. There are two cases:

- Suppose that $(X\alpha, Y\beta)$ is not decomposable. Then by the maximality of R_2 we have $(X\alpha', Y\beta')$ in R_2 such that $(\alpha', \beta') \sqsubseteq (\alpha, \beta)$ with $\alpha \sim \alpha'$ and $\beta \sim \beta'$. If $X, Y \in V_{fin}$ then clearly $(\alpha', \beta') \sqsubseteq (\alpha, \beta) \sqsubset (X\alpha, Y\beta)$ from which it follows that $(\alpha, \alpha') \sqsubset (X\alpha, Y\beta)$ and $(\beta, \beta') \sqsubset (X\alpha, Y\beta)$. By the induction hypothesis we now conclude that $\alpha \underset{R}{\longleftrightarrow}^* \alpha'$ and $\beta \underset{R}{\longleftrightarrow}^* \beta'$ from which we get $X\alpha \underset{R}{\longleftrightarrow}^* Y\beta$ as desired. If $X \in V_\infty$ and $Y \in V_{fin}$ we get $\alpha = \alpha' = \epsilon$ and therefore $X \sim Y\beta$. As $Y \in V_{fin}$ we have $s(\beta') \leq s(\beta) < s(Y\beta)$ hence $(\beta, \beta') \sqsubset (X, Y\beta)$ which by the induction hypothesis implies $\beta \underset{R}{\longleftrightarrow}^* \beta'$. But then $X \underset{R}{\longleftrightarrow}^* Y\beta' \underset{R}{\longleftrightarrow}^* Y\beta$ as desired. Finally, if $X, Y \in V_\infty$ then $\alpha = \alpha' = \epsilon$ and also $\beta = \beta' = \epsilon$. Hence we have $(X, Y) \in R_2$ from which $X \underset{R}{\longleftrightarrow}^* Y$ follows.

- Suppose $(X\alpha, Y\beta)$ is decomposable. By the definition of decomposability it follows that $X, Y \in V_{fin}$. Assume without loss of generality that we have γ such that $\gamma\alpha \sim \beta$ and $X \sim Y\gamma$. As X is normed and $X \sim Y\gamma$ clearly $s(\gamma\alpha) < s(X\alpha)$. Similarly as Y is normed we also have $s(\beta) < s(Y\beta)$ and therefore $(\gamma\alpha, \beta) \sqsubset (X\alpha, Y\beta)$ from which $\gamma\alpha \underset{R}{\longleftrightarrow}^* \beta$ follows by the induction hypothesis. As $X \sim Y\gamma$ with $X \in V_{fin}$ we have $(X, Y\gamma) \in R_1$ from the maximality of R_1. But then $X\alpha \underset{R}{\longleftrightarrow}^* Y\gamma\alpha \underset{R}{\longleftrightarrow}^* Y\beta$ as desired.

This completes the proof. $\square$

Thus, Corollary 3.1 can be strengthened to: $\alpha \sim \beta$ iff there is a *finite* self-bisimulation R such that $\alpha R \beta$. We now show that this is sufficient for semi-decidability of $\sim$. For given a finite relation R on $(V_{fin}^* V_\infty) \cup V_{fin}^*$ it is semi-decidable whether it is a self-bisimulation. The procedure consists in defining a derivation or proof system: the axioms are the pairs in R, and the rules are congruence rules for sequential composition together with the usual equivalence rules. Consequently, for each n let $D_n(R)$ be the finite set of pairs (α, β) which are derivable within n steps of the proof system.

Definition 3.4 *A finite relation R on $(V_{fin}^* V_\infty) \cup V_{fin}^*$ is an* n-self-bisimulation *iff $\alpha R \beta$ implies that for all $a \in Act$*

1. *if $\alpha \xrightarrow{a} \alpha'$ then $\beta \xrightarrow{a} \beta'$ for some β' with $(\alpha', \beta') \in D_n(R)$*
2. *if $\beta \xrightarrow{a} \beta'$ then $\alpha \xrightarrow{a} \alpha'$ for some α' with $(\alpha', \beta') \in D_n(R)$*

For each n clearly it is decidable whether a finite relation R on $(V_{fin}^* V_\infty) \cup V_{fin}^*$ is an n-self-bisimulation. Moreover, if R is a finite self-bisimulation then for some n it is an n-self-bisimulation.

We now complete the proof that bisimulation equivalence is semi-decidable using a dovetailing technique (compare [6]). Let $R_0 \ldots R_i \ldots$ be an effective enumeration of all finite relations on $(V_{fin}^* V_\infty) \cup V_{fin}^*$ and let $g : \mathbf{N}^2 \to \mathbf{N}$ be an effective bijection. To check whether $\alpha \sim \beta$, for each $n \geq 0$ in turn consider the pair $(i, j) = g^{-1}(n)$: if $\alpha R_i \beta$ then test if R_i is a j-self-bisimulation. Consequently, if $\alpha \sim \beta$ this must be established at the n^{th} stage of this procedure for some n. The decidability result is now established.

Theorem 3.2 *Bisimulation equivalence is decidable for all guarded BPA processes.*

4 Conclusion

We have shown that bisimulation equivalence is decidable for BPA. As the proof involves two semi-decision procedures it is not obvious how to determine the complexity of solving this problem. Moreover it does not provide us with an intuitive technique for deciding bisimilarity as does the tableau method in [16, 15] which also has the advantage of providing us with a way of extracting a complete axiomatization for normed BPA processes. A similar result for full BPA would be a proper extension of Milner's axiom system for regular processes [18].

More generally this work addresses the area of infinite-state processes. Besides deciding equivalences there is also the question of model checking: a recent result [5] shows decidability for fragments of the modal mu-calculus in the case of normed BPA. There is also the question of pushdown automata processes (which generate a richer family of transition graphs than BPA processes). [20] contains a very elegant characterization of their graphs.

Of more interest to concurrency theory are process languages with parallel combinators. Although bisimulation equivalence is undecidable for ACP, CCS, and CSP it is decidable for the calculus BPP (Basic Parallel processes), which is the recursive fragment of CCS with parallel but without the restriction operator [9, 10]. An open question is whether bisimulation is decidable in the case of the PA calculus which is BPA with an added parallel operator. Moreover there may be even finer useful equivalences which permit general decidability results.

Acknowledgements

The authors would like to thank Didier Caucal and Robin Milner for useful discussions and insights.

References

[1] 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.

[2] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *JACM*, 40:653-682, 1993.

[3] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Control*, 60:109–137, 1984.

[4] 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.

[5] O. Burkart and B. Steffen. Model checking for context-free processes. *Proceedings of CONCUR'92, LNCS 630*, 1992.

[6] D. Caucal. Décidabilité de l'egalité des langages algébriques infinitaires simples. In *Proceedings of STACS 86, LNCS 210*, pages 37–48. Springer-Verlag, 1986.

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

[8] D. Caucal. Graphes canoniques de graphes algébriques. *Informatique théorique et Applications (RAIRO)*, 24(4):339–352, 1990.

[9] S. Christensen. *Decidability and decomposition in process algebra.* PhD thesis, University of Edinburgh, 1993.

[10] S. Christensen, Y. Hirshfeld and F. Moller Bisimulation equivalence is decidable for basic parallel processes. *Proceedings of CONCUR'93*, *LNCS 715*, 143-157, 1993.

[11] J. F. Groote. A short proof of the decidability of bisimulation for normed BPA-processes. Tech. Report Utrecht University 1992.

[12] J.F. Groote and H. Hüttel. Undecidable equivalences for basic process algebra. Technical Report ECS-LFCS-91-169, Department of Computer Science, University of Edinburgh, August 1991.

[13] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, 1988.

[14] J. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[15] H. Hüttel. *Decidability, Behavioural Equivalences and Infinite Transition Graphs.* PhD thesis, University of Edinburgh, December 1991.

[16] 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.

[17] Dung T. Huynh and Lu Tian. On deciding readiness and failure equivalences for processes. Technical Report UTDCS-31-90, University of Texas at Dallas, September 1990.

[18] R. Milner. A complete inference system for a class of regular behaviours. *Journal of Computer and System Sciences*, 28:439–466, 1984.

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

[20] D. Muller and P. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37:51–75, 1985.