A polynomial-time algorithm for deciding bisimulation equivalence of normed Basic Parallel Processes

Yoram Hirshfeld*
School of Mathematics and Computer Science
Tel Aviv University
Israel

Mark Jerrum†
Department of Computer Science
University of Edinburgh
United Kingdom

Faron Moller‡
Swedish Institute of Computer Science
Kista, Sweden

April 22, 1994

Abstract

A polynomial-time algorithm is presented for deciding bisimulation equivalence of so-called Basic Parallel Processes: multisets of elementary processes combined by a commutative parallel-composition operator.

Keywords. Design and analysis of algorithms, Theory of concurrent systems, Formal languages, Process algebra.

*The first author is currently visiting the Department of Computer Science, University of Edinburgh.

†The second author is a Nuffield Foundation Science Research Fellow, and is supported in part by grant GR/F 90363 of the UK Science and Engineering Research Council, and by Esprit Working Group No. 7097, "RAND."

‡The third author is supported by Esprit Basic Research Action No. 7166, "CONCUR2."

1 Discussion

Consider a collection (actually a multiset) of "elementary processes," each of a finite number of different kinds. At any instant, one of the elementary processes may spontaneously perform one of a finite number of "actions," appropriate to its kind, and in so doing transform itself into a certain multiset (possibly empty) of new elementary processes. These multisets of elementary processes are called "processes." The set of possible transitions available to a process is conveniently described by a context-free grammar in Greibach normal form: the variable on the left of a production represents the elementary process about to undergo the transition, the variables on the right represent the new elementary processes into which it is transformed, and the sole terminal symbol represents the action. (Precise definitions of these and subsequent notions are presented in Section 2.) The simple processes described above have been termed "Basic Parallel Processes" (BPP) in the literature.

Two such processes are said to be "bisimilar" or "bisimulation equivalent" if, roughly, they may evolve together in such a way that whenever the first process performs a certain action, the second process is able to respond by performing the same action, and vice versa. The notion of bisimulation equivalence was introduced by Park [13] as a notion of behavioural equivalence between concurrent systems, and has been intensively studied in the intervening years; see [1, 2, 3, 4, 5, 6, 7, 9, 10, 12] for just some of the work in this area which is relevent to our present study. Bisimulation equivalence plays an important role in algebraic theories of concurrency, such as that of Milner [11].

Bisimulation equivalence in the context of BPP differs from language equivalence of context-free grammars in two ways: (a) the former concerns multisets, and hence is commutative in some sense, whereas the latter concerns words and is non-commutative, and (b) the former is "on-line" or "reactive" and must hold at every step, whereas the latter only considers the end product (a word in the language).

If it is possible from any process to reach the empty process through some sequence of transitions, we say the system is *normed*. Christensen, Hirshfeld, and Moller [2] have demonstrated that bisimulation equivalence is decidable for normed BPP. Since the state space (all multisets of elementary processes) is infinite, the issue of decidability is a real one. Indeed, the somewhat related question of language equivalence of context-free grammars has long been known to be undecidable [8], while the more closely related question of deciding "trace equivalence" for normed BPP was recently shown to be undecidable by Hirshfeld [6].

The time complexity of the decision procedure of Christensen et al. is not known to be bounded by any primitive recursive function. In this article we present (see Theorem 2) the first polynomial-time algorithm for deciding bisimulation equivalence for normed BPP: indeed the first decision procedure for this problem with any quantified time bound. More recently Christensen, Hirshfeld, and Moller [3] were able to prove decidability of this problem without the assumption that processes are normed. However, the algorithm we present here uses the notion of norm in an essential way, and there is no obvious way to adapt it to the general case. It is entirely possible that bisimulation equivalence for unrestricted BPP is decidable in polynomial time, but it seems that new ideas would be required to demonstrate this.

The result presented here complements a recent result by the authors to the effect that bisimulation equivalence for "normed BPA" is decidable in polynomial time [7]. Basic Process Algebra (BPA) is, roughly, the non-commutative analogue of BPP. Interestingly, different techniques seem to be necessary in the two cases. As with BPP, this problem remains decidable when the norm condition is dropped (see Christensen, Hüttel, and Stirling [4]), but again no polynomial-time decision procedure is known.

2 Notation and basic facts

Let Δ be a context-free grammar in Greibach normal form with variables $V = \{X_1, \ldots, X_n\}$ and terminals A.[1] For reasons that will become apparent presently, we sometimes refer to variables as *(elementary) processes* and terminals as *actions*. Denote by $\mathrm{Mon}\, V$ the set of all monomials over the variable set V, i.e., all formal products of the form $\alpha = X_1^{a_1} \ldots X_n^{a_n}$, where $(a_1, \ldots, a_n) \in \mathbb{N}^n$. Informally, a monomial corresponds to a *process*, which is a multiset of elementary processes. In the present context, the ordering of variables in the right-hand side of a production is not significant; thus each production may be expressed in the form $X \stackrel{a}{\rightarrow} \beta$, where $X \in V$, $a \in A$, and $\beta \in \mathrm{Mon}\, V$. If a monomial α contains variable X then the production $X \stackrel{a}{\rightarrow} \beta$ may be applied to the monomial, and the result is the monomial $\alpha' = \alpha\beta/X$. We express this state of affairs by writing $\alpha \stackrel{a}{\rightarrow} \alpha'$: in the terminology of process algebra, the process α makes a transition to process α' by

[1] Recall that a grammar is in *Greibach normal form* if the right hand side of every production consists of a single terminal followed by a (possibly empty) sequence of variables.

performing action a.

A relation $\sim$ on the process set $\mathrm{Mon}\, V$ is a *bisimulation* if, for all $\alpha, \beta \in \mathrm{Mon}\, V$ satisfying $\alpha \sim \beta$, conditions (a) and (b) hold:

(a) if $\alpha \xrightarrow{a} \alpha'$ for some $a \in A$ then $\beta \xrightarrow{a} \beta'$ for some β' with $\alpha' \sim \beta'$, and

(b) if $\beta \xrightarrow{a} \beta'$ for some $a \in A$ then $\alpha \xrightarrow{a} \alpha'$ for some α' with $\alpha' \sim \beta'$.

Two processes α and β are *bisimilar* or *bisimulation equivalent* if there exists a bisimulation $\sim$ such that $\alpha \sim \beta$. The set of bisimulations is clearly closed under union, so there is a unique maximal bisimulation. Two processes are thus bisimilar if they are related in the maximal bisimulation. It is easily checked that the maximal bisimulation is a congruence relation.

We may assume that every variable occurs on the left of some production, otherwise the variable is redundant and can simply be removed from all productions. If, for every variable $X \in V$, the language of words generated from X via productions in Δ is non-empty, we say that Δ is *normed*. For $\alpha \in \mathrm{Mon}\, V$, define $\mathrm{norm}\, \alpha$, the *norm* of α, to be the length of a shortest sequence of transitions leading from α to the empty monomial 1.[2] Note that if the grammar Δ is normed then every variable X has finite norm. Without loss of generality we may assume that the variables are given in order of non-decreasing norm, so that $\mathrm{norm}\, X_1 \leq \mathrm{norm}\, X_2 \leq \cdots \leq \mathrm{norm}\, X_n$.

Our proof relies heavily on the fact that normed processes (i.e., processes with finite norm) are uniquely decomposable into prime processes. The following result was obtained (in a slightly more general setting) by Christensen, Hirshfeld, and Moller [2]. The proof is quite straightforward, and we repeat it here in the interest of making the article self contained. Say that an elementary process X_i is *prime* if there does not exist a process $\alpha \in \mathrm{Mon}\{X_1, \ldots, X_{i-1}\}$ such that $X_i \sim \alpha$.

Theorem 1 *Let* $\mathrm{Mon}\, V$ *be the set of processes defined by a normed grammar* Δ *in Greibach normal form. Let* $P_1, \ldots, P_r$ *be the primes of* V*, ordered consistently with* V *itself. For any processes* $\alpha = P_1^{a_1} \ldots P_r^{a_r}$ *and* $\beta = P_1^{b_1} \ldots P_r^{b_r}$*, if* $\alpha \sim \beta$ *then* $(a_1, \ldots, a_r) = (b_1, \ldots, b_r)$.

Proof Suppose that the claim is false, and that α and β form a counterexample of smallest norm;[3] thus $\alpha \sim \beta$ and yet $(a_1, \ldots, a_r) \neq (b_1, \ldots, b_r)$. Let j be the

[2] In the language CCS of Milner [11], this "inactive process" is denoted $\mathbf{0}$; the different symbols merely reflect the fact that we have chosen a multiplicative notation instead of an additive one.

[3] Note that $\mathrm{norm}\, \alpha = \mathrm{norm}\, \beta$, since $\alpha \sim \beta$.

largest index such that $a_j \neq b_j$. Without loss of generality, suppose $a_j > b_j$. We distinguish three cases. In each case, we show that process α may perform a norm-reducing transition $\alpha \xrightarrow{a} \alpha'$ that cannot be matched by any transition $\beta \xrightarrow{a} \beta'$ with $\alpha' \sim \beta'$ (or vice versa with the roles of α and β reversed): a contradiction. Observe that, by minimality of the counterexample, if α' and β' are to be bisimilar then their prime decompositions must be identical.

Case I. If $a_k > 0$ for some $k < j$, then let α perform some norm-reducing transition via process P_k. Process β cannot match this transition, since it cannot increase the exponent b_j without decreasing some exponent to the right of b_j. This contradicts minimality of the counterexample, and the assumption that α and β are bisimilar.

Case II. If $a_k > 0$ for some $k > j$, then let α perform a norm-reducing transition via process P_k that maximises (after reduction into primes) the increase in the exponent a_j. Again the process β is unable to match this transition.

Case III. The monomial $\alpha = P_j^{a_j}$ is a prime power. Note that $b_k = 0$ for all $k > j$ by choice of j, and that $a_j \geq 2$ by the definition of "prime." If $b_j > 0$, let β perform a norm-reducing transition via P_j; this transition cannot be matched by α, since it would require the exponent a_j to decrease by at least two. Finally, if $b_j = 0$, then let α perform a norm-reducing transition via P_j; this transition cannot be matched by β, since β is unable to increase the exponent b_j.

The cases are inclusive, so the theorem is proved. $\square$

3 The decision procedure

Define the *size* of monomial $\alpha \in \operatorname{Mon} V$ to be the sum of the lengths of the binary encodings of the various exponents appearing in the monomial; the size of a production $X \xrightarrow{a} \beta$ to be the length of the triple (X, a, β), encoded in binary; and the size of a context-free grammar Δ to be the sum of the sizes of all the productions contained within it. Our aim is to prove the following.

Theorem 2 *Suppose the set* $\operatorname{Mon} V$ *of processes is defined by a normed, context-free grammar* Δ *in Greibach normal form. There is a polynomial-time (in the size of* α*,* β*, and* Δ*) algorithm to decide* $\alpha \sim \beta$ *for arbitrary* $\alpha, \beta \in \operatorname{Mon} V$*.*

To prepare for the description of the algorithm and the proof of the theorem, we require some definitions and a few preparatory lemmas. To ensure a smooth development, the proofs of the lemmas are deferred to the end of the section.

Suppose R is any relation on $\mathrm{Mon}\, V$. We say that a pair $(\alpha, \beta) \in (\mathrm{Mon}\, V)^2$ *satisfies (norm-reducing) expansion in* R provided conditions (a) and (b) hold:

(a) if $\alpha \xrightarrow{a} \alpha'$ for some $a \in A$ (and $\mathrm{norm}\, \alpha' < \mathrm{norm}\, \alpha$) then $\beta \xrightarrow{a} \beta'$ for some β' with $\alpha' \mathrel{R} \beta'$, and

(b) if $\beta \xrightarrow{a} \beta'$ for some $a \in A$ (and $\mathrm{norm}\, \beta' < \mathrm{norm}\, \beta$) then $\alpha \xrightarrow{a} \alpha'$ for some α' with $\alpha' \mathrel{R} \beta'$.

Observe that a relation $\sim$ is a bisimulation if every pair (α, β) with $\alpha \sim \beta$ satisfies expansion in $\sim$. Observe also that if R is an equivalence relation (respectively, congruence) then the relation "satisfies (norm-reducing) expansion in R" is an equivalence relation (respectively, congruence).

Define a *unique decomposition base*, D, to be a pair (Π, Γ), where $\Pi = \Pi(D) = \{P_1, \ldots, P_r\} \subseteq V$ is a set of *primes*, and $\Gamma = \Gamma(D)$ is a set of pairs $(X, P_1^{x_1} \ldots P_r^{x_r})$, one for each non-prime elementary process $X \in V - \Pi$. The set Γ may be viewed as specifying, for each non-prime process X, a decomposition of X into primes.[4] A unique decomposition base defines an equivalence relation $\equiv_D$ on $\mathrm{Mon}\, V$: the relation $\alpha \equiv_D \beta$ holds between $\alpha, \beta \in \mathrm{Mon}\, V$ if the prime decompositions of α and β are equal (as monomials).

Lemma 3 *Let D be a unique decomposition base. Then:*

(i) *the equivalence relation $\equiv_D$ is a congruence with cancellation,*[5] *which coincides with $\overset{D}{\equiv}$, the smallest congruence containing $\Gamma(D)$;*

(ii) *there is a polynomial-time (in n, and the sizes of α and β) algorithm to decide $\alpha \equiv_D \beta$ for arbitrary $\alpha, \beta \in \mathrm{Mon}\, V$;*

(iii) *the relation $\equiv_D$ is a bisimulation provided every pair in $\Gamma(D)$ satisfies expansion in $\equiv_D$; this condition may be checked by a polynomial-time algorithm;*

(iv) *the maximal bisimulation $\sim$ coincides with the congruence $\equiv_D$, where D represents the unique decomposition in $\sim$.*

[4] These "primes" are not in general the primes with respect to the maximal bisimulation, which were the subject of Theorem 1.

[5] Thus, in addition to satisfying the conditions of a congruence, $\equiv_D$ has the property that $\alpha \equiv_D \beta$ whenever $\alpha\gamma \equiv_D \beta\gamma$.

The next lemma allows us to shrink a congruence, defined by a unique decomposition base, whenever it is strictly larger than the maximal bisimulation.

Lemma 4 *Let D be a unique decomposition base such that the congruence $\equiv_D$ is norm-preserving and strictly contains the maximal bisimulation $\sim$. Then it is possible, in polynomial time, to find (a representation of) a relation $\equiv$ on* $\mathrm{Mon}\, V$ *such that:*

(i) *the relation $\alpha \equiv \beta$ is decidable in polynomial time (in n, and the sizes of α and β);*

(ii) *the relation $\equiv$ is a congruence;*

(iii) *there is a variable $X \in V$ that is decomposable in $\equiv_D$ but not in $\equiv$;*

(iv) *the inclusions $\sim \subseteq \equiv \subset \equiv_D$ hold.*

The final lemma allows us to "smooth out" an unmanageable congruence into a congruence defined by a unique decomposition base.

Lemma 5 *Let $\equiv$ be a norm-preserving, polynomial-time computable congruence satisfying $\equiv \supseteq \sim$, where $\sim$ denotes maximal bisimulation. Then there is a decomposition base D, computable in polynomial time, such that $\sim \subseteq \equiv_D \subseteq \equiv$.*

With Lemmas 3, 4, and 5 in place, the procedure for deciding bisimulation equivalence writes itself; in outline it goes as follows.

(1) Let the congruence $\equiv$ be defined by $\alpha \equiv \beta$ iff $\mathrm{norm}\, \alpha = \mathrm{norm}\, \beta$.

(2) Compute a decomposition base D such that $\sim \subseteq \equiv_D \subseteq \equiv$, using Lemma 5.

(3) If $\equiv_D$ is a bisimulation — a condition that can be checked in polynomial time using Lemma 3 — then halt and return the relation $\equiv_D$.

(4) Compute a congruence $\equiv$ satisfying $\sim \subseteq \equiv \subset \equiv_D$, using Lemma 4. Go to step (2).

The proof of the main result is virtually immediate.

Proof of Theorem 2 On each iteration of the loop formed by lines (2)–(4), the number of primes increases by at least one. Thus the number of iterations is bounded by n, and each iteration requires only polynomial time by Lemmas 3, 4, and 5. $\square$

We complete the section by providing the missing proofs of the various lemmas.

Proof of Lemma 3

(i) It is easy to check that $\equiv_D$ is a congruence with cancellation containing $\Gamma(D)$; thus $\equiv_D$ certainly includes $\stackrel{D}{\equiv}$, the smallest congruence containing $\Gamma(D)$. On the other hand, if $\alpha \equiv_D \beta$, then β can be obtained from α via a finite sequence of substitutions chosen from $\Gamma(D)$, and the reverse inclusion holds.

(ii) The algorithm may be modelled directly on the definition of $\equiv_D$.

(iii) Suppose $\alpha \equiv_D \beta$ and let $\alpha \equiv_D P_1^{a_1} \dots P_r^{a_r} \equiv_D \beta$ be the common prime decomposition of α and β. By assumption, the pairs $(\alpha, P_1^{a_1} \dots P_r^{a_r})$ and $(\beta, P_1^{a_1} \dots P_r^{a_r})$ both satisfy expansion in $\equiv_D$, and so then does the pair (α, β), by transitivity of the relation "satisfies expansion in $\equiv_D$."

(iv) This part follows from Theorem 1.

This concludes the proof of the lemma. $\square$

Proof of Lemma 4 Define the relation $\equiv$ as follows: for all $\alpha, \beta \in \mathrm{Mon}\, V$, the relationship $\alpha \equiv \beta$ holds iff $\alpha \equiv_D \beta$ and the pair (α, β) satisfies expansion in $\equiv_D$. We must demonstrate that $\equiv$ satisfies conditions (i)–(iv) in the statement of the lemma.

(i) The relationship $\alpha \equiv \beta$ is clearly decidable in polynomial time: an algorithm follows directly from the definition of $\equiv$.

(ii) The relation $\equiv$ is the intersection of two congruences, and hence itself a congruence.

(iii) If the congruence $\equiv_D$ is not a bisimulation then, by Lemma 3, there is a first (non-prime) variable X such that the pair $(X, P_1^{x_1} \dots P_r^{x_r}) \in \Gamma(D)$ does not satisfy expansion in $\equiv_D$. We show that X is indecomposable with respect to the relation $\equiv$.

Suppose to the contrary that X is decomposable, that is to say, $X \equiv \alpha \in \mathrm{Mon}\, V$, where α contains only variables that are smaller than X. By definition of $\equiv$, the pair (X, α) satisfies expansion in $\equiv_D$, and $X \equiv_D \alpha \equiv_D P_1^{x_1} \dots P_r^{x_r}$. By minimality of X, for every non-prime variable Y

occurring in α, the pair $(Y, P_1^{y_1} \dots P_r^{y_r}) \in \Gamma(D)$ satisfies expansion in $\equiv_D$. Thus the pair $(\alpha, P_1^{x_1} \dots P_r^{x_r})$, and by transitivity the pair $(X, P_1^{x_1} \dots P_r^{x_r})$, satisfies expansion in $\equiv_D$, contradicting the choice of X.

(iv) It is clear that the relation $\equiv$ is contained in $\equiv_D$. On the other hand, if $\alpha \sim \beta$ then the pair (α, β) satisfies expansion in $\sim$ and hence in $\equiv_D$; it follows that $\alpha \equiv \beta$.

This concludes the proof of the lemma. $\square$

Proof of Lemma 5 As before, assume that variables are in order of non-decreasing norm. Given a congruence $\equiv \supseteq \sim$ on $\mathrm{Mon}\, V$ we define, by induction, a unique decomposition base D_i for $\mathrm{Mon}\{X_1, \dots, X_i\}$, with primes $\Pi(D) = \{P_1, \dots, P_r\}$, such that:

(a) the inclusion $\equiv_{D_i} \subseteq \equiv$ holds on the set $\mathrm{Mon}\{X_1, \dots, X_i\}$;

(b) if $X_j \equiv_{D_i} P_1^{x_1} \dots P_r^{x_r}$ is the decomposition of X_j, for some $j \leq i$, then the pair $(X_j, P_1^{x_1} \dots P_r^{x_r})$ satisfies norm-reducing expansion in $\equiv_{D_i}$;

(c) if $X_j \sim Q_1^{x_1} \dots Q_t^{x_t}$ is the prime decomposition of X_j with respect to $\sim$, for some $j \leq i$, then $X_j \equiv_{D_i} Q_1^{x_1} \dots Q_t^{x_t}$;[6]

(d) if $P_1^{x_1} \dots P_r^{x_r} \equiv P_1^{y_1} \dots P_r^{y_r}$ and the pair $(P_1^{x_1} \dots P_r^{x_r}, P_1^{y_1} \dots P_r^{y_r})$ satisfies norm-reducing expansion in $\equiv_{D_i}$, then $(x_1, \dots, x_r) = (y_1, \dots, y_r)$.

Observe that the lemma follows from (a) and (c), specialised to $i = n$.

Assume that there exists a unique decomposition base D_i for $\mathrm{Mon}\{X_1, \dots, X_i\}$ that satisfies these properties. We wish to demonstrate that D_i may be extended to a unique decomposition base D_{i+1} for $\mathrm{Mon}\{X_1, \dots, X_{i+1}\}$ also satisfying conditions (a)–(d) above; this involves finding, in polynomial time, a consistent decomposition for the variable X_{i+1}.

In outline the extension of the unique decomposition base is achieved as follows. By condition (d) we know that there is at most one product $P_1^{x_1} \dots P_r^{x_r}$ of primes of D_i such that $X_{i+1} \equiv P_1^{x_1} \dots P_r^{x_r}$ and such that the pair $(X_{i+1}, P_1^{x_1} \dots P_r^{x_r})$ satisfies norm-reducing expansions in $\equiv_{D_i}$. If there is such a product, it is declared

[6] Again, note that $Q_1, \dots, Q_t$, although primes with respect to $\sim$, are not in general primes with respect to $\equiv_{D_i}$.

as the decomposition of X_{i+1}; otherwise X_{i+1} is declared to be prime. It is clear that conditions (a) and (b) continue to hold.

To show (c), assume that $X_{i+1} \sim Q_1^{x_1} \dots Q_t^{x_t}$ is the prime decomposition of X_{i+1} with respect to the maximal bisimulation $\sim$. Note that if X_{i+1} is prime with respect to $\sim$ then there is nothing to prove, so we may assume the decomposition is non-trivial. Let $Q_1 \equiv_{D_i} \alpha_1, \dots, Q_t \equiv_{D_i} \alpha_t$ be the prime decompositions of $Q_1, \dots, Q_t$ with respect to $\equiv_{D_i}$. Then $X_{i+1} \equiv \alpha_1^{x_1} \dots \alpha_t^{x_t}$, where, it will be observed, the right-hand side is a product of primes with respect to $\equiv_{D_i}$. The pairs (Q_j, α_j) satisfy norm-reducing expansion in $\equiv_{D_i}$, by condition (b), and the pair $X_{i+1} \sim Q_1^{x_1} \dots Q_t^{x_t}$ satisfies norm-reducing expansions in $\equiv_{D_i}$, by virtue of X_{i+1} and $Q_1^{x_1} \dots Q_t^{x_t}$ being bisimilar; it follows by transitivity that the pair $(X_{i+1}, \alpha_1^{x_1} \dots \alpha_t^{x_t})$ also satisfies norm-reducing expansion in $\equiv_{D_i}$. Thus, by the uniqueness condition (d), $X_{i+1} \equiv_{D_{i+1}} \alpha_1^{x_1} \dots \alpha_t^{x_t}$ must be the chosen prime decomposition of X_{i+1} with respect to $\equiv_{D_{i+1}}$.

To show (d), assume to the contrary that $\Pi(D_{i+1}) = \{P_1, \dots, P_r\}$ is the set of primes with respect to $\equiv_{D_{i+1}}$, and that the pair (α, β), where $\alpha = P_1^{a_1} \dots P_r^{a_r}$ and $\beta = P_1^{b_1} \dots P_r^{b_r}$, is a counterexample to condition (d): that is, $\alpha \equiv \beta$, $(a_1, \dots, a_r) \neq (b_1, \dots, b_r)$, and the pair (α, β) satisfies norm-reducing expansions in $\equiv_{D_{i+1}}$. We demonstrate that this assumption leads to a contradiction.

Let j be the largest index such that $a_j \neq b_j$, and assume, without loss of generality, that $a_j > b_j$. We distinguish three cases:

Case I. If $a_k > 0$ for some $k < j$, then let α perform some norm-reducing transition via process P_k. Process β cannot match this transition, since it cannot increase the exponent b_j without decreasing some exponent to the right of b_j.

Case II. If $a_k > 0$ for some $k > j$, then let α perform a norm-reducing transition via process P_k that maximises the increase in the exponent a_j. Again the process β is unable to match this transition.

Case III. The monomial $\alpha = P_j^{a_j}$ is a power of a prime with respect to $\equiv_{D_{i+1}}$. Note that $b_k = 0$ for all $k > j$ by choice of j, and $a_j \geq 2$, otherwise, P_j would not be prime with respect to D_{i+1}. If $b_j > 0$, let β perform a norm-reducing transition via P_j; this transition cannot be matched by α, since it would require the exponent a_j to decrease by at least two. Finally, if $b_j = 0$, then let α perform a norm-reducing transition via P_j; this transition cannot be matched by β, since β is unable to increase the exponent b_j.

This completes the inductive step.

It only remains to show that the extension of D_i to D_{i+1} may be computed in polynomial time. We need to investigate the possibility that X_{i+1} may be expressed as $X_{i+1} \equiv P_1^{x_1} \dots P_r^{x_r}$ where the pair $(X_{i+1}, P_1^{x_1} \dots P_r^{x_r})$ satisfies norm-reducing expansion in $\equiv_{D_i}$. Recall that effecting the transition $X \xrightarrow{a} \beta$ may be viewed as multiplication by β/X; thus the transition $\alpha \xrightarrow{a} \beta$ may occur precisely if $\alpha' = \alpha\beta/X$ is a monomial (i.e., the exponent of X is non-negative), in which case α' is the resulting process.

Now choose any norm-reducing transition $X_{i+1} \xrightarrow{a} \alpha \in \mathrm{Mon}\{X_1, \dots, X_i\}$, and let $\alpha \equiv_{D_i} P_1^{a_1} \dots P_r^{a_r}$ be the prime decomposition of α. If this transition is to be matched by $\beta = P_1^{x_1} \dots P_r^{x_r}$ then α/β must be one of a finite set of possibilities, one for each production in Δ. Thus there are only as many possibilities for the process β as there are productions in Δ; for each possibility it is easy to check whether (i) $X_{i+1} \equiv P_1^{x_1} \dots P_r^{x_r}$, and (ii) the pair $(X_{i+1}, P_1^{x_1} \dots P_r^{x_r})$ satisfies norm-reducing expansion in $\equiv_{D_i}$. Thus the extension of D_i to D_{i+1} may indeed be computed in polynomial time. $\square$

References

[1] J. C. M. Baeten, J. A. Bergstra, and J. W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. In Proceedings of PARLE 87, J. W. de Bakker, A. J. Nijman, P. C. Treleaven (eds), *Lecture Notes in Computer Science* **259**, pp 93–114, Springer-Verlag, 1987.

[2] S. Christensen, Y. Hirshfeld, and F. Moller, Decomposability, decidability and axiomatisability for bisimulation equivalence on basic parallel processes. In *Proceedings of LICS 93*, IEEE Computer Society Press, 1993.

[3] S. Christensen, Y. Hirshfeld, and F. Moller, Bisimulation equivalence is decidable for basic parallel processes. In Proceedings of CONCUR 93, E. Best (ed), *Lecture Notes in Computer Science* **715**, pp 143–157, Springer-Verlag, 1993.

[4] S. Christensen, H. Hüttel, and C. Stirling, Bisimulation equivalence is decidable for all context-free processes. In Proceedings of CONCUR 92, W. R. Cleaveland (ed.), *Lecture Notes in Computer Science* **630**, pp 138–147. Springer-Verlag, 1992.

[5] J. F. Groote, A short proof of the decidability of bisimulation for normed BPA processes, *Information Processing Letters* **42**, pp 167–171, 1991.

[6] Y. Hirshfeld, Petri nets and the equivalence problem. To appear in *Proceedings of CSL 93* (Karl Meinke, ed.), Springer Lecture Notes in Computer Science.

[7] Y. Hirshfeld, M. Jerrum, and F. Moller, *A polynomial algorithm for deciding bisimilarity of normed context-free processes*, Report ECS-LFCS-94-286, Department of Computer Science, University of Edinburgh, March 1994.

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

[9] H. Hüttel and C. Stirling, Actions speak louder than words: proving bisimilarity for context-free processes. In *Proceedings of LICS 91*, IEEE Computer Society Press, pp 376–386, 1991.

[10] D. T. Huynh and L. Tian, Deciding bisimilarity of normed context-free processes is in Σ_2^{P}, *Journal of Theoretical Computer Science* **123**, pp 183–197, 1994.

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

[12] R. Milner and F. Moller, Unique decomposition of processes, *Journal of Theoretical Computer Science* **107**, pp 357–363, 1993.

[13] D. M. R. Park, Concurrency and Automata on Infinite Sequences, *Lecture Notes in Computer Science* **104**, pp 168–183, Springer Verlag, 1981.