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

Yoram Hirshfeld, Mark Jerrum and Faron Moller

Mathematical Structures in Computer Science / Volume 6 / Issue 03 / June 1996, pp 251 - 259
DOI: 10.1017/S0960129500000992, Published online: 04 March 2009

Math. Struct. in Comp. Science (1996), *vol.* 6, *pp.* 251 –259

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

YORAM HIRSHFELD[†], MARK JERRUM[‡] and FARON MOLLER[§]

[†]*School of Mathematics and Computer Science, Tel Aviv University, Israel*

[‡]*Department of Computer Science, University of Edinburgh, United Kingdom*

[§]*Swedish Institute of Computer Science, Kista, Sweden*

Received 16 May 1994; revised 11 May 1995

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.

1. Discussion

We consider a class of processes formed by parallel composition from a finite set of 'elementary processes'. Each of these elementary process may spontaneously perform one of a finite number of 'actions', and in so doing make a transition to a certain multiset (possibly empty) of new elementary processes. The set of possible transitions available to an elementary process is conveniently described by a context-free grammar in Greibach normal form: the variable on the left-hand side of a production represents the elementary process about to undergo the transition, the variables on the right-hand side represent the new elementary processes into which it is transformed, and the sole terminal symbol represents the action.

A multiset P of elementary processes is called a 'process'. At any instant one of the elementary processes contained in the multiset may undergo a transition, as specified by the grammar, and, as a result, the process P itself undergoes a transition to new process (multiset) P'. (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, and can be viewed as a fragment of CCS, the *Calculus*

[†] The work described here was done while the first author was visiting the Department of Computer Science, University of Edinburgh.

[‡] At the time the work was done, the second author was a Nuffield Foundation Science Research Fellow, and was 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 was supported by Esprit Basic Research Action No. 7166, 'CONCUR2'.

of Communicating Systems (Milner 1989), or of the process algebra ACP, the *Algebra of Communicating Processes* (Bergstra and Klop 1985).

Two such processes are said to be 'bisimilar' or 'bisimulation equivalent' if, roughly speaking, 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 in Park (1981) as a notion of behavioural equivalence between concurrent systems, and has been intensively studied in the intervening years: see Baeten *et al.* (1993), Christensen *et al.* (1993a; 1993b), Christensen *et al.* (1992), Groote (1991), Hirshfeld (1994), Hirshfeld *et al.* (1994), Hüttel and Stirling (1991), Huynh and Tian (1994), and Milner and Moller (1993) for just some of the work in this area relevant to our present study. Bisimulation equivalence plays an important role in algebraic theories of concurrency, such as that based on CCS.

There is a certain formal similarity between the question of bisimulation equivalence in the context of BPP, and language equivalence of context-free grammars, and it is helpful at this juncture to contrast the two. There are two differences: (a) the former concerns multisets, and hence is commutative (when an elementary process undergoes a transition, there is no ordering on the multiset of elementary processes that is formed), whereas the latter concerns words and is non-commutative (when a production is applied to a non-terminal symbol, the ordering of resulting sequence of terminals and non-terminals is significant); and (b) the former is 'on-line' or 'reactive' and involves a condition that must hold at every step, whereas the latter only considers the end product (a word in the language generated by the grammar).

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 (Christensen *et al.* 1993a) 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 (Hopcroft and Ullman 1979), while the more closely related question of deciding 'trace equivalence' for normed BPP was recently shown to be undecidable (Hirshfeld 1994).

The time complexity of the decision procedure of Christensen, Hirshfeld, and Moller 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 (Christensen *et al.* 1993b) 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 (Hirshfeld *et al.* 1994). Basic Process Algebra (BPA) is, roughly speaking, 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 (Christensen *et al.* 1992), but again no polynomial-time decision procedure is known.

Our primary aim in this article is to demonstrate membership of the equivalence problem in the class P rather than to strive for the best possible time bound. We therefore make no attempt to optimise the various operations, or to provide a tight exact complexity analysis. Nevertheless, none of the steps of the algorithm are at all outlandish, and it should be possible to construct an efficient implementation based on the ideas presented here.

2. Notation and basic facts

Recall that a context-free 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. Let Δ be a context-free grammar in Greibach normal form with variables $V = \{X_1, \ldots, X_n\}$ and terminals A. 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, that is, 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, which will generally be denoted by a lower case Greek letter, 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 $X \xrightarrow{a} \beta$, where $X \in V$, $a \in A$, and $\beta \in \mathrm{Mon}\, V$. If a monomial α contains variable X, the production $X \xrightarrow{a} \beta$ may be applied to the monomial, and the result is the monomial $\alpha' = \alpha\beta/X$. We express this state of affairs by writing $\alpha \xrightarrow{a} \alpha'$: in the terminology of process algebra, the process α makes a transition to process α' by performing action a.

A relation R on the process set $\mathrm{Mon}\, V$ is a *bisimulation* if, for all $\alpha, \beta \in \mathrm{Mon}\, V$ satisfying $\alpha\, R\, \beta$, 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 $\alpha'\, R\, \beta'$; and
(b) if $\beta \xrightarrow{a} \beta'$ for some $a \in A$, then $\alpha \xrightarrow{a} \alpha'$ for some α' with $\alpha'\, R\, \beta'$.

Two processes α and β are *bisimilar* or *bisimulation equivalent* if there exists a bisimulation R such that $\alpha\, R\, \beta$. The set of bisimulations is clearly closed under union, so there is a unique maximal bisimulation, denoted $\sim$. 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.

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[†]; if no such sequence exists, define $\mathrm{norm}\, \alpha = \infty$. Note that if the grammar Δ is normed, 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 $\mathrm{norm}\, X_1 \leq \mathrm{norm}\, X_2 \leq \cdots \leq \mathrm{norm}\, X_n$.

[†] In the language CCS of Milner (1989), this 'inactive process' is denoted **0**; the different symbols merely reflect the fact that we have chosen a multiplicative notation instead of an additive one.

Our proof relies heavily on the fact that normed processes (that is, processes with finite norm) are uniquely decomposable into prime processes. The following result was given, in a slightly more general setting, in Christensen *et al.* (1993a). The proof is quite straightforward, and we repeat it here in the interests 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 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[‡]; thus $\alpha \sim \beta$, and yet $(a_1, \ldots, a_r) \neq (b_1, \ldots, b_r)$. Let j be the 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), which is a contradiction. Observe that, by minimality of the counterexample, if α' and β' are to be bisimilar, their prime decompositions must be identical.

Case I. If $a_k > 0$ for some $k < j$, 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 $k > j$, 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 $b_j = 0$, 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. □

3. The decision procedure

Define the *size* of monomial $\alpha \in \mathrm{Mon}\, V$ 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.

[†] Note that the notion of prime is defined relative to a particular ordering of elementary processes.

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

Theorem 2. Suppose the set $\mathrm{Mon}\, V$ 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 $\alpha, \beta \in \mathrm{Mon}\, V$.

A key element in the development of the algorithm whose existence is asserted in Theorem 2 is the notion of 'decomposition base', which will be defined presently. Even before we reach the precise definitions, however, it is possible to give an overview of the algorithm. For this purpose it is sufficient to know that any decomposition base D defines a congruence relation $\equiv_D$ on processes. The idea is to start with a congruence $\equiv$ that is potentially much coarser than the maximal bisimulation $\sim$; here, we choose to start with the congruence $\equiv$ in which two processes are related if they have equal norm. We then construct a decomposition base such that the congruence $\equiv_D$ defined by D contains $\sim$, and is contained in $\equiv$ (symbolically, $\sim \subseteq \equiv_D \subseteq \equiv$). If the congruences $\equiv_D$ and $\sim$ are equal, we are done, otherwise we construct a new congruence $\equiv$ that contains $\sim$ and is *strictly* contained in $\equiv_D$. This process is iterated, and in a (small) finite number of steps, the congruence $\equiv$ converges from above to $\sim$.

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 expansion* (*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 $\alpha' \, R \, \beta'$; 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 $\alpha' \, R \, \beta'$.

Observe that a relation R is a bisimulation if every pair (α, β) with $\alpha \, R \, \beta$ satisfies expansion in R. 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 *decomposition base* D, for V, to be a pair (Π, Γ), where $\Pi = \Pi(D) = \{P_1, \ldots, P_r\} \subseteq V$ is a set of *indecomposable elementary processes* or *atoms*, and $\Gamma = \Gamma(D)$ is a set of pairs $(X, P_1^{x_1} \ldots P_r^{x_r})$, one for each decomposable elementary process $X \in V - \Pi$. The set Γ may be viewed as specifying, for each decomposable process X, a decomposition of X into atoms. A 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 decompositions of α and β into atoms are equal (as monomials). We say that a relation $\equiv$ on $\mathrm{Mon}\, V$ is a *congruence with cancellation* if, in addition to satisfying the conditions of a congruence, $\equiv$ has the property that $\alpha \equiv \beta$ whenever $\alpha\gamma \equiv \beta\gamma$.

Lemma 3. Let D be a decomposition base. Then

(i) the equivalence relation $\equiv_D$ is a congruence with cancellation, 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 into primes (with respect to $\sim$).

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

Lemma 4. Let D be a 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 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 $\equiv$ (that is, there is no monomial $\alpha \in \text{Mon}\, V$, containing only variables smaller than X, such that $X \equiv \alpha$);
(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 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 $\text{norm}\,\alpha = \text{norm}\,\beta$.
(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 atoms (indecomposable elementary processes) 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. □

We now provide 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 $\overset{D}{\equiv}$, 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 $\alpha \equiv_D P_1^{a_1} \ldots P_r^{a_r} \equiv_D \beta$ be the common decomposition of α and β into atoms. By assumption, the pairs $(\alpha, P_1^{a_1} \ldots P_r^{a_r})$ and $(\beta, P_1^{a_1} \ldots P_r^{a_r})$ both satisfy expansion in $\equiv_D$, and thus so does the pair (α, β), by transitivity of the relation 'satisfies expansion in $\equiv_D$'. (Recall that the relation 'satisfies expansion in $\equiv_D$' is a congruence provided $\equiv_D$ is.)

(iv) This part follows from Theorem 1.

This concludes the proof of the lemma. □

Proof of Lemma 4. Define the relation $\equiv$ as follows: for all $\alpha, \beta \in \operatorname{Mon} V$, 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 is hence itself a congruence.

(iii) If the congruence $\equiv_D$ is not a bisimulation, then, by Lemma 3, there is a first (decomposable) variable X such that the pair $(X, P_1^{x_1} \ldots P_r^{x_r}) \in \Gamma(D)$ 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, $X \equiv \alpha \in \operatorname{Mon} V$, where α contains only variables that are smaller than X. By definition of $\equiv$, the pair (X, α) satisfies expansion in $\equiv_D$, and $X \equiv_D \alpha \equiv_D P_1^{x_1} \ldots P_r^{x_r}$. By minimality of X, for every decomposable variable Y occurring in α, the pair $(Y, P_1^{y_1} \ldots P_r^{y_r}) \in \Gamma(D)$ satisfies expansion in $\equiv_D$. Thus the pair $(\alpha, P_1^{x_1} \ldots P_r^{x_r})$, and by transitivity the pair $(X, P_1^{x_1} \ldots P_r^{x_r})$, 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 $\alpha \sim \beta$, the pair (α, β) satisfies expansion in $\sim$, and hence in $\equiv_D$, and it follows that $\alpha \equiv \beta$.

This concludes the proof of the lemma. □

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

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

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

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

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

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

For the base case of the induction, we take D_1 to be the decomposition base containing the single atom X_1; this is easily seen to satisfy the conditions (a)–(d). For the inductive step, assume that there exists a decomposition base D_i for $\operatorname{Mon}\{X_1, \ldots, X_i\}$ that satisfies these conditions. We wish to demonstrate that D_i may be extended to a decomposition

base D_{i+1} for $\mathrm{Mon}\{X_1,\ldots,X_{i+1}\}$ 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 decomposition base is achieved as follows. By condition (d) we know that there is at most one product $P_1^{x_1}\ldots P_r^{x_r}$ of atoms of D_i such that $X_{i+1} \equiv P_1^{x_1}\ldots P_r^{x_r}$ and such that the pair $(X_{i+1}, P_1^{x_1}\ldots P_r^{x_r})$ satisfies norm-reducing expansion in $\equiv_{D_i}$. If there is such a product, it is declared as the decomposition of X_{i+1}, otherwise X_{i+1} is declared to be an atom. It is clear that conditions (a) and (b) continue to hold.

To show (c), assume that $X_{i+1} \sim Q_1^{x_1}\ldots Q_t^{x_t}$ 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 $\sim$, there is nothing to prove, so we may assume the decomposition is non-trivial. Let $Q_1 \equiv_{D_i} \alpha_1$, ..., $Q_t \equiv_{D_i} \alpha_t$ be the decompositions of $Q_1,\ldots,Q_t$ into atoms with respect to $\equiv_{D_i}$. Then $X_{i+1} \equiv \alpha_1^{x_1}\ldots\alpha_t^{x_t}$, where, it will be observed, the right-hand side is a product of atoms 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 $X_{i+1} \sim Q_1^{x_1}\ldots Q_t^{x_t}$ satisfies norm-reducing expansion in $\equiv_{D_i}$, by virtue of X_{i+1} and $Q_1^{x_1}\ldots Q_t^{x_t}$ being bisimilar; it follows by transitivity that the pair $(X_{i+1}, \alpha_1^{x_1}\ldots\alpha_t^{x_t})$ also satisfies norm-reducing expansion in $\equiv_{D_i}$. Thus, by the uniqueness condition (d), $X_{i+1} \equiv_{D_{i+1}} \alpha_1^{x_1}\ldots\alpha_t^{x_t}$ 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 $\Pi(D_{i+1}) = \{P_1,\ldots,P_r\}$ is the set of atoms of $\equiv_{D_{i+1}}$, and that the pair (α,β), where $\alpha = P_1^{a_1}\ldots P_r^{a_r}$ and $\beta = P_1^{b_1}\ldots P_r^{b_r}$, is a counterexample to condition (d) (that is, $\alpha \equiv \beta$, $(a_1,\ldots,a_r) \neq (b_1,\ldots,b_r)$, 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 an atom of $\equiv_{D_{i+1}}$. Note that $b_k = 0$ for all $k > j$ by choice of j, and $a_j \geq 2$, otherwise, P_j would be decomposable 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 $b_j = 0$, 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 $X_{i+1} \equiv P_1^{x_1}\ldots P_r^{x_r}$, where the pair $(X_{i+1}, P_1^{x_1}\ldots P_r^{x_r})$ 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 (that is, the exponent of X is non-negative), in which case α' is the resulting process.
Now choose any norm-reducing transition $X_{i+1} \xrightarrow{a} \alpha \in \mathrm{Mon}\{X_1, \ldots, X_i\}$, and let $\alpha \equiv_{D_i} P_1^{a_1} \ldots P_r^{a_r}$ be the decomposition of α into atoms of D_i. If this transition is to be matched by $\beta = P_1^{x_1} \ldots P_r^{x_r}$, 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) $X_{i+1} \equiv P_1^{x_1} \ldots P_r^{x_r}$, and (ii) the pair $(X_{i+1}, P_1^{x_1} \ldots P_r^{x_r})$ 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. □

References

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

Bergstra, J. A. and Klop, J. W. (1985) Algebra of Communicating Processes with Abstraction. *Theoretical Computer Science* **37** (1) 77–121.

Christensen, S., Hirshfeld Y. and Moller, F. (1993a) Decomposability, decidability and axiomatisability for bisimulation equivalence on basic parallel processes. In: *Proceedings of the Eighth Symposium on Logic in Computer Science*, IEEE Computer Society Press 386–396.

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

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

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

Hirshfeld, Y. (1994) Petri nets and the equivalence problem. In Börger, E., Gurevich, Y. and Meinke, K. (eds.) Proceedings of CSL '93. *Springer-Verlag Lecture Notes in Computer Science* **832** 165–174.

Hirshfeld, Y., Jerrum, M. and Moller, F. (1994) A polynomial algorithm for deciding bisimilarity of normed context-free processes, Report ECS-LFCS-94-286, Department of Computer Science. University of Edinburgh. (To appear in *Theoretical Computer Science.*)

Hopcroft, J. E. and Ullman, J. D. (1979) *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley.

Hüttel, H. and Stirling, C. (1991) Actions speak louder than words: proving bisimilarity for context-free processes. In: *Proceedings of the Sixth Symposium on Logic in Computer Science*, IEEE Computer Society Press 376–386.

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

Milner, R. (1989) *Communication and Concurrency*, Prentice Hall.

Milner, R. and Moller, F. (1993) Unique decomposition of processes. *Journal of Theoretical Computer Science* **107** 357–363.

Park, D. M. R. (1981) Concurrency and Automata on Infinite Sequences. *Springer-Verlag Lecture Notes in Computer Science* **104** 168–183.