Book chapter :
Harwood, W., Moller, F. & Setzer, A. (2006). *Weak Bisimulation Approximants.* Computer Science Logic, (pp. 365

http://dx.doi.org/10.1007/11874683_24

Weak Bisimulation Approximants

Will Harwood, Faron Moller, and Anton Setzer*

Department of Computer Science, Swansea University
Singleton Park, Sketty, Swansea SA2 8PP, UK
{cswill,f.g.moller,a.g.setzer}@swansea.ac.uk

Abstract. ***Bisimilarity*** $\sim$ and ***weak bisimilarity*** $\approx$ are canonical notions of equivalence between processes, which are defined co-inductively, but may be approached – and even reached – by their (transfinite) inductively-defined ***approximants*** $\sim_\alpha$ and $\approx_\alpha$. For arbitrary processes this approximation may need to climb arbitrarily high through the infinite ordinals before stabilising. In this paper we consider a simple yet well-studied process algebra, the Basic Parallel Processes (BPP), and investigate for this class of processes the minimal ordinal α such that $\approx\, = \,\approx_\alpha$.

The main tool in our investigation is a novel proof of ***Dickson's Lemma***. Unlike classical proofs, the proof we provide gives rise to a tight ordinal bound, of ω^n, on the order type of non-increasing sequences of n-tuples of natural numbers. With this we are able to reduce a long-standing bound on the approximation hierarchy for weak bisimilarity $\approx$ over BPP, and show that $\approx\, = \,\approx_{\omega^\omega}$.

1 Introduction

There has been great interest of late in the development of techniques for deciding equivalences between infinite-state processes, particularly for the question of bisimilarity between processes generated by some type of term algebra. Several surveys of this developing area have been published, beginning with [16], and there is now a chapter in the Handbook of Process Algebra dedicated to the topic [2], as well as a website devoted to maintaining an up-to-date comprehensive overview of the state-of-the-art [20].

While questions concerning strong bisimilarity have been successfully addressed, techniques for tackling the question of weak bisimilarity, that is, when silent unobservable transitions are allowed, are still lacking, and many open problems remain. The main difficulty arising when considering weak bisimilarity is that processes immediately become infinite-branching: at any point in a computation, a single action can result in any number of transitions leading to any one of an infinite number of next states. Common finiteness properties fail due to this; in particular, bisimilarity can no longer be characterised by its finite approximations in the way that it can for finite-branching processes. For arbitrary infinite-branching processes, we may need to climb arbitrarily high through

* Supported by EPSRC grant GR/S30450/01.

the transfinite approximations to bisimilarity before reaching the bisimulation relation itself.

In this paper we consider the problem of weak bisimilarity for so-called Basic Parallel Processes (BPP), a simple yet well-studied model of concurrent processes. These correspond to commutative context-free processes, or equivalently to communication-free Petri nets. The question as to the decidability of weak bisimilarity between BPP processes remains unsolved (though decidability results for very restricted classes of BPP have been established by Hirshfeld in [10] and Stirling in [21]). It has recently been shown that the problem is at least PSPACE-hard [19], even in the restricted case of so-called normed BPP, but this sheds no light one way or the other as to decidability. Jančar suggests in [14] that the techniques he uses there to establish PSPACE-completeness of strong bisimilarity for BPP might be exploited to give a decision procedure for weak bisimilarity, but three years later this conjecture remains unsubstantiated.

It has long been conjectured that for BPP, weak bisimilarity is characterised by its $(\omega\times 2)$-level approximation. Such a result could provide a way to a decision procedure. However, no nontrivial approximation bound has before now been established; the strength of the $(\omega\times 2)$-conjecture remains rooted only in the fact that no counterexample has been found. In this paper we provide the first non-trivial countable bound on the approximation: for a BPP defined over k variables, weak bisimilarity is reached by the ω^{2k} level; weak bisimilarity is thus reached by the ω^ω level for any BPP.

Our argument is based on a new constructive proof of Dickson's Lemma which provides an ordinal bound on the sequences described by the Lemma. This proof is presented in Section 2 of the paper. After this, the definitions necessary for the remainder of the paper are outlined in Section 3 along with a variety of results, and our results on BPP are presented in Section 4. We finish with some concluding observations in Section 5.

2 Ordinal Bounds for Dickson's Lemma

In the sequel we shall use the following notation. We let x, y (with subscripts) range over natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$; $\vec{\boldsymbol{x}}$, $\vec{\boldsymbol{y}}$ (with subscripts) range over finite sequences (n-tuples) of natural numbers; and $\vec{\boldsymbol{X}}$, $\vec{\boldsymbol{Y}}$ range over arbitrary (finite or infinite) sequences of such n-tuples. We shall use angle brackets to denote sequences, such as $\vec{\boldsymbol{x}} = \langle x_1, \ldots, x_n\rangle$ and $\vec{\boldsymbol{X}} = \langle \vec{\boldsymbol{x}}_1, \vec{\boldsymbol{x}}_2, \ldots\rangle$, and juxtaposition to represent concatenation; e.g., if $\vec{\boldsymbol{X}}$ is a finite sequence of n-tuples then $\vec{\boldsymbol{X}}\langle\vec{\boldsymbol{x}}\rangle$ is the longer sequence which has the extra n-tuple $\vec{\boldsymbol{x}}$ added to the end. Finally, we shall use the notation $(\cdot)_k$ to select the kth component from a sequence; for example, if $\vec{\boldsymbol{x}}_i = \langle x_1, \ldots, x_n\rangle$ then $(\vec{\boldsymbol{x}}_i)_k = x_k$. (The parentheses are used to avoid confusion with the subscripting allowed in the variable naming the sequence.)

One n-tuple $\vec{\boldsymbol{y}} = \langle y_1, \ldots, y_n\rangle$ of natural numbers ***dominates*** another such n-tuple $\vec{\boldsymbol{x}} = \langle x_1, \ldots, x_n\rangle$ if $\vec{\boldsymbol{x}} \le \vec{\boldsymbol{y}}$, where $\le$ is considered pointwise, that is, $x_i \le y_i$ for each $i \in \{1, \ldots, n\}$. A sequence of n-tuples is a ***non-dominating sequence*** over $\mathbb{N}^n$ if no element of the sequence dominates any of its predecessors in the sequence. A ***tree*** – by which we mean a rooted directed graph with no undirected cycles – with nodes labelled by n-tuples from $\mathbb{N}^n$ is a ***non-dominating tree*** over $\mathbb{N}^n$ if the sequence of labels along any path through the tree is a non-dominating sequence.

Dickson's Lemma [6] asserts that there can be no infinite non-dominating sequences.

Lemma 1 (Dickson's Lemma). *All non-dominating sequences are finite. That is, given an infinite sequence of vectors $\vec{\boldsymbol{x}}_1, \vec{\boldsymbol{x}}_2, \vec{\boldsymbol{x}}_3, \ldots \in \mathbb{N}^n$, we can always find indices i, j with $i < j$ such that $\vec{\boldsymbol{x}}_i \leq \vec{\boldsymbol{x}}_j$.*

The standard proof of this lemma uses a straightforward induction on n: for the base case, any sequence of decreasing natural numbers must be finite; and for the induction step, from an infinite sequence of n-tuples you extract an infinite subsequence in which the last components are nondecreasing (either constant or increasing), and then apply induction on the sequence of $(n-1)$-tuples which arise by ignoring these last components.

The problem with this proof is that it is nonconstructive; in particular, it gives no clue as to the ordinal bound on the lengths of non-dominating sequences. The difficulty with determining an ordinal bound comes from the fact that the domination order is not a total order on n-tuples (as opposed, for example, to lexicographical order). We provide here an alternative constructive proof from which we can extract an ordinal bound on the lengths of such sequences.

Theorem 1 (Constructive Dickson's Lemma). *The order type of the set of non-dominating sequences of n-tuples of natural numbers with partial ordering*

$$\vec{\boldsymbol{X}} \prec \vec{\boldsymbol{Y}} \quad \overset{def}{\Leftrightarrow} \quad \vec{\boldsymbol{X}} \textit{ strictly extends } \vec{\boldsymbol{Y}}$$

is ω^n.

Proof. That the order type is at *least* ω^n is clear: the order type of the set of lexicographically descending sequences with respect to extension is ω^n, and this set is contained in the set of non-dominating sequences.

It remains to show that the order type is at *most* ω^n. This result will follow immediately from the construction of a function

$$f_n : (\mathbb{N}^n)^+ \to \mathbb{N}^n$$

on non-empty finite sequences of n-tuples which satisfies the following property:

> If $\vec{\boldsymbol{X}}\langle\vec{\boldsymbol{x}}\rangle$ is a non-dominating sequence of n-tuples, and $\vec{\boldsymbol{X}}$ is itself non-empty, then $f_n(\vec{\boldsymbol{X}}\langle\vec{\boldsymbol{x}}\rangle) <_{\text{lex}} f_n(\vec{\boldsymbol{X}})$.

We shall inductively define these functions f_n. The base case is straightforward: we can define f_1 by

$$f_1(\langle x_1, \ldots, x_k\rangle) \overset{\text{def}}{=} x_k.$$

A non-dominating sequence of natural numbers is simply a decreasing sequence, which has ordinal bound ω.

For illustrative purposes we carry out the construction of the function f_2 for sequences of pairs, and later generalise our construction to sequences of n-tuples.

Given a non-empty finite sequence of pairs $\vec{\boldsymbol{X}} = \langle\langle x_1, y_1\rangle, \ldots, \langle x_k, y_k\rangle\rangle$, define

- $\textsc{min}_x(\vec{\boldsymbol{X}}) \stackrel{\text{def}}{=} \min\{\, x_i : 1 \le i \le k \,\}$,
- $\textsc{min}_y(\vec{\boldsymbol{X}}) \stackrel{\text{def}}{=} \min\{\, y_i : 1 \le i \le k \,\}$, and
- $S_2(\vec{\boldsymbol{X}}) \stackrel{\text{def}}{=} \Big\{ \langle x, y\rangle : \textsc{min}_x(\vec{\boldsymbol{X}}) \le x,\ \textsc{min}_y(\vec{\boldsymbol{X}}) \le y,\ \textit{and}$
  $\langle x_i, y_i\rangle \not\le \langle x, y\rangle \textit{ for all } i : 1 \le i \le k \Big\}.$

$S_2(\vec{\boldsymbol{X}})$ consists of the pairs with which the sequence $\vec{\boldsymbol{X}}$ can be extended without altering the $\textsc{min}_x$ and $\textsc{min}_y$ values and yet while maintaining non-domination. Note that $S_2(\vec{\boldsymbol{X}})$ must be finite: if we let i and j be such that $x_i = \textsc{min}_x(\vec{\boldsymbol{X}})$ and $y_j = \textsc{min}_y(\vec{\boldsymbol{X}})$, then in order for $\langle x, y\rangle \not\ge \langle x_i, y_i\rangle$ and $\langle x, y\rangle \not\ge \langle x_j, y_j\rangle$ we must have $x < x_j$ (since $y \ge y_j$) and $y < y_i$ (since $x \ge x_i$).

Suppose that $\vec{\boldsymbol{Y}} = \vec{\boldsymbol{X}}\langle\langle x, y\rangle\rangle$ is a non-dominating sequence, and that $\vec{\boldsymbol{X}}$ is itself non-empty. Then clearly $\textsc{min}_x(\vec{\boldsymbol{Y}}) \le \textsc{min}_x(\vec{\boldsymbol{X}})$ and $\textsc{min}_y(\vec{\boldsymbol{Y}}) \le \textsc{min}_y(\vec{\boldsymbol{X}})$; and if equality holds in both cases then $S_2(\vec{\boldsymbol{Y}}) \subsetneq S_2(\vec{\boldsymbol{X}})$, since $S_2(\vec{\boldsymbol{Y}}) \subseteq S_2(\vec{\boldsymbol{X}})$ yet $\langle x, y\rangle \in S_2(\vec{\boldsymbol{X}}) \setminus S_2(\vec{\boldsymbol{Y}})$. Thus $|S_2(\vec{\boldsymbol{Y}})| < |S_2(\vec{\boldsymbol{X}})|$.

We can then define the function f_2 on non-empty sequences $\vec{\boldsymbol{X}}$ of pairs as follows:

$$f_2(\vec{\boldsymbol{X}}) \stackrel{\text{def}}{=} \langle\, \textsc{min}_x(\vec{\boldsymbol{X}}) + \textsc{min}_y(\vec{\boldsymbol{X}}),\ |S_2(\vec{\boldsymbol{X}})|\,\rangle$$

If $\vec{\boldsymbol{X}}\langle\langle x, y\rangle\rangle$ is a non-dominating sequence and $\vec{\boldsymbol{X}}$ is itself non-empty, then by the above argument we must have that $f_2(\vec{\boldsymbol{X}}\langle\langle x, y\rangle\rangle) <_{\text{lex}} f_2(\vec{\boldsymbol{X}})$.

For the inductive construction of f_n we assume we have constructed the function f_{n-1} as required. For $1 \le i \le n$ we define the function

$$\pi_{\text{-}i}(\langle x_1, \ldots, x_n\rangle) \stackrel{\text{def}}{=} \langle x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\rangle$$

which simply deletes the ith component from the n-tuple $\langle x_1, \ldots, x_n\rangle$. Next, given a non-empty finite sequence $\vec{\boldsymbol{X}} = \langle \vec{\boldsymbol{x}}_1, \ldots, \vec{\boldsymbol{x}}_k\rangle$ of n-tuples, we define the set

$$\textsc{nd}_{\text{-}i}(\vec{\boldsymbol{X}}) \stackrel{\text{def}}{=} \Big\{ \langle \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_1}), \ldots, \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_p})\rangle \ :\ p > 0,\ 0 < i_1 < \cdots < i_p \le k, \\ \textit{and}\ \ \langle \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_1}), \ldots, \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_p})\rangle \text{ is non-dominating} \Big\}$$

which consists of the non-dominating subsequences of $(n-1)$-tuples of $\vec{\boldsymbol{X}}$ in which the ith components of the n-tuples have been deleted. Finally we make the following definitions:

- $\text{MIN}_{\text{-}i}(\vec{\boldsymbol{X}}) \stackrel{\text{def}}{=} \min_{<_{\text{lex}}}\{\, f_{n-1}(\vec{\boldsymbol{Y}}) \;:\; \vec{\boldsymbol{Y}} \in \text{ND}_{\text{-}i}(\vec{\boldsymbol{X}})\,\}$
- $S_n(\vec{\boldsymbol{X}}) \stackrel{\text{def}}{=} \Big\{ \vec{\boldsymbol{x}} \;:\; \text{MIN}_{\text{-}i}(\vec{\boldsymbol{X}}) = \text{MIN}_{\text{-}i}(\vec{\boldsymbol{X}}\langle\vec{\boldsymbol{x}}\rangle) \textit{ for all } i : 1 \le i \le n, \textit{ and } \vec{\boldsymbol{x}}_i \not\le \vec{\boldsymbol{x}} \textit{ for all } i : 1 \le i \le k \Big\}$

$S_n(\vec{\boldsymbol{X}})$ consists of the n-tuples with which the sequence $\vec{\boldsymbol{X}}$ can be extended without altering the $\text{MIN}_{\text{-}i}$ values and yet while maintaining non-domination. Note that $S_n(\vec{\boldsymbol{X}})$ must be finite. To see this, let $1 \le i \le n$ and $i_1, \ldots, i_p$ be such that

$$\text{MIN}_{\text{-}i}(\vec{\boldsymbol{X}}) = f_{n-1}(\langle \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_1}), \ldots, \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_p})\rangle),$$

and suppose that $\vec{\boldsymbol{x}} \in S_n(\vec{\boldsymbol{X}})$. If the sequence $\langle \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_1}), \ldots, \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_p}), \pi_{\text{-}i}(\vec{\boldsymbol{x}})\rangle$ is non-dominating, then by induction we would get that

$$\begin{aligned} \text{MIN}_{\text{-}i}(\vec{\boldsymbol{X}}\langle\vec{\boldsymbol{x}}\rangle) &\le_{\text{lex}} f_{n-1}(\langle \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_1}), \ldots, \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_p}), \pi_{\text{-}i}(\vec{\boldsymbol{x}})\rangle) \\ &<_{\text{lex}} f_{n-1}(\langle \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_1}), \ldots, \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_p})\rangle) \\ &= \text{MIN}_{\text{-}i}(\vec{\boldsymbol{X}}) \end{aligned}$$

contradicting $\vec{\boldsymbol{x}} \in S_n(\vec{\boldsymbol{X}})$. Therefore we must have that $\pi_{\text{-}i}(\vec{\boldsymbol{x}}) \ge \pi_{\text{-}i}(\vec{\boldsymbol{x}}_{i_j})$ for some j. But since $\vec{\boldsymbol{x}} \not\ge \vec{\boldsymbol{x}}_{i_j}$ we must then have that $(\vec{\boldsymbol{x}})_i < (\vec{\boldsymbol{x}}_{i_j})_i$.

Suppose that $\vec{\boldsymbol{Y}} = \vec{\boldsymbol{X}}\langle\vec{\boldsymbol{x}}\rangle$ is a non-dominating sequence, and that $\vec{\boldsymbol{X}}$ is itself non-empty. Then $\text{MIN}_{\text{-}i}(\vec{\boldsymbol{Y}}) \le \text{MIN}_{\text{-}i}(\vec{\boldsymbol{X}})$ for all i (since $\text{ND}_{\text{-}i}(\vec{\boldsymbol{X}}) \subseteq \text{ND}_{\text{-}i}(\vec{\boldsymbol{Y}})$); and if equality holds in all cases then $S_n(\vec{\boldsymbol{Y}}) \subsetneq S_n(\vec{\boldsymbol{X}})$ since $S_n(\vec{\boldsymbol{Y}}) \subseteq S_n(\vec{\boldsymbol{X}})$ yet $\vec{\boldsymbol{x}} \in S_n(\vec{\boldsymbol{X}}) \setminus S_n(\vec{\boldsymbol{Y}})$. Thus $|S_n(\vec{\boldsymbol{Y}})| < |S_n(\vec{\boldsymbol{X}})|$.

We can then define the function f_n on non-empty sequences $\vec{\boldsymbol{X}}$ of n-tuples as follows:

$$f_n(\vec{\boldsymbol{X}}) \;=\; \Big(\textstyle\sum_{i=1}^{n} \text{MIN}_{\text{-}i}(\vec{\boldsymbol{X}})\Big)\langle |S_n(\vec{\boldsymbol{X}})| \rangle$$

where the sum is taken component-wise. (This sum is, if we identify $\langle k_1, \ldots, k_n\rangle$ with $\omega^{n-1} \cdot k_1 + \omega^{n-2} \cdot k_2 + \cdots + \omega^0 \cdot k_n$, the natural sum of ordinals.) If $\vec{\boldsymbol{X}}\langle\vec{\boldsymbol{x}}\rangle$ is a non-dominating sequence and $\vec{\boldsymbol{X}}$ is itself non-empty, then by the above argument we must have that $f_n(\vec{\boldsymbol{X}}\langle\vec{\boldsymbol{x}}\rangle) <_{\text{lex}} f_n(\vec{\boldsymbol{X}})$. □

2.1 Ordinal Bounds on Trees

Our constructive version of Dickson's Lemma easily extends to trees, where we take the following definition of the height of a well-founded tree (that is, a tree with no infinite paths).

Definition 1. *The* ***height*** *of a well-founded tree rooted at t is defined by*

$$h(t) \stackrel{\text{def}}{=} \sup\{\, h(s) + 1 \;:\; t \longrightarrow s \,\}.$$

(By convention, $\sup \emptyset = 0$*.)*

Theorem 2. *If t is (the root of) a non-dominating tree over $\mathbb{N}^n$, then $h(t) \leq \omega^n$.*

Proof. For each node x of the tree, define $\ell(x) \in \mathbb{N}^n$ by $\ell(x) = f_n(\pi_x)$, where f_n is as defined in the proof of Lemma 1, and π_x is the non-dominating sequence of labels on the path from (the root) t to x. It will suffice then to prove that $h(x){\leq}\ell(x)$ (viewing $\ell(x)$ as an ordinal, that is, interpreting the n-tuple $\langle k_1, \ldots, k_n\rangle \in \mathbb{N}^n$ as $\omega^{n-1}k_1 + \omega^{n-2}k_2 + \cdots + \omega^0 k_n$) for all nodes x of the tree. This is accomplished by a straightforward induction on $h(x)$:

$$\begin{aligned} h(x) &= \sup\{\, h(y){+}1 : x \to y \,\} \\ &\leq \sup\{\, \ell(y){+}1 : x \to y \,\} \quad \text{(by induction)} \\ &\leq \ell(x). \end{aligned}$$

□

3 Processes and Bisimilarity

A ***process*** is represented by (a state in) a ***labelled transition system*** defined as follows.

Definition 2. *A **labelled transition system (LTS)** is a triple $\mathcal{S} = (S, Act, \to)$ where S is a set of **states**, Act is a finite set of **actions**, and $\to \subseteq S \times Act \times S$ is a **transition relation**.*

We write $s \xrightarrow{a} t$ instead of $(s, a, t) \in \to$, thus defining an infix binary relation $\xrightarrow{a} = \{(s,t) \,:\, (s,a,t) \in \to\}$ for each action $a \in Act$.

It is common to admit silent transitions to model the internal unobservable evolution of a system. In standard automata theory these are typically referred to as "epsilon" (or occasionally "lambda") transitions, but in concurrency theory they are commonly represented by a special action $\tau \in Act$. With this, we can then define observable transitions as follows:

$$\begin{aligned} &s \overset{\tau}{\Rightarrow} t \;\text{ iff }\; s \,(\xrightarrow{\tau})^*\, t \;\text{ and} \\ &s \overset{a}{\Rightarrow} t \;\text{ iff }\; s \,(\xrightarrow{\tau})^* \cdot \xrightarrow{a} \cdot (\xrightarrow{\tau})^*\, t \quad \text{for } a \neq \tau. \end{aligned}$$

In general, $\overset{a}{\Rightarrow} \supseteq \xrightarrow{a}$; and over an LTS with no silent transitions, $\overset{a}{\Rightarrow} = \xrightarrow{a}$, and in this case all the relations we define wrt $\Rightarrow$ will be identical to the analogous relations defined wrt $\to$.

The notion of "behavioural sameness" between two processes (which we view as two states in the same LTS) can be formally captured in many different ways (see, e.g., [8] for an overview). Among those behavioural equivalences, ***bisimilarity*** enjoys special attention. Its formal definition is as follows.

Definition 3. *Let $\mathcal{S} = (S, Act, \to)$ be an LTS. A binary relation $\mathcal{R} \subseteq S \times S$ is a **bisimulation relation** iff whenever $(s,t) \in R$, we have that*

- *for each transition $s \xrightarrow{a} s'$ there is a transition $t \xrightarrow{a} t'$ such that $(s', t') \in \mathcal{R}$; and*

– *for each transition* $t \xrightarrow{a} t'$ *there is a transition* $s \xrightarrow{a} s'$ *such that* $(s', t') \in \mathcal{R}$.

Processes s *and* t *are* ***bisimulation equivalent (bisimilar)****, written* $s \sim t$*, iff they are related by some bisimulation. Thus* $\sim$ *is the union, and ergo the largest, of all bisimulation relations.*

If we replace the transition relation $\rightarrow$ *in this definition with the weak transition relation* $\Rightarrow$*, we arrive at the definition of a* ***weak bisimulation relation*** *defining* ***weak bisimulation equivalence (weak bisimilarity)****, which we denote by* $\approx$*. In general,* $\approx\ \supseteq\ \sim$*; and over an LTS with no silent transitions,* $\approx\ =\ \sim$.

The above definition of (weak) bisimilarity is a co-inductive one, but can be approximated using the following inductively-defined stratification.

Definition 4. *The* ***bisimulation approximants*** $\sim_\alpha$*, for all ordinals* $\alpha \in \mathcal{O}$*, are defined as follows:*

– $s \sim_0 t$ *for all process states* s *and* t.
– $s \sim_{\alpha+1} t$ *iff*
  - *for each transition* $s \xrightarrow{a} s'$ *there is a transition* $t \xrightarrow{a} t'$ *such that* $s' \sim_\alpha t'$*; and*
  - *for each transition* $t \xrightarrow{a} t'$ *there is a transition* $s \xrightarrow{a} s'$ *such that* $s' \sim_\alpha t'$.
– *For all limit ordinals* λ, $s \sim_\lambda t$ *iff* $s \sim_\alpha t$ *for all* $\alpha < \lambda$.

The ***weak bisimulation approximants*** $\approx_\alpha$ *are defined by replacing the transition relation* $\rightarrow$ *with the weak transition relation* $\Rightarrow$ *in the above definition.*

The following results are then standard.

Theorem 3.

1. *Each* $\sim_\alpha$ *and* $\approx_\alpha$ *is an equivalence relation over the states of any LTS.*
2. *Given* $\alpha < \beta$, $\sim_\alpha\ \supseteq\ \sim_\beta$ *and* $\approx_\alpha\ \supseteq\ \approx_\beta$ *over the states of any LTS; and in general these define strictly decreasing hierarchies: given any ordinal* α *we can provide an LTS with states* s *and* t *satisfying* $s \sim_\alpha t$ *but* $s \not\sim_{\alpha+1} t$ *(and* $s \approx_\alpha t$ *but* $s \not\approx_{\alpha+1} t$*).*
3. $s \sim t$ *iff* $s \sim_\alpha t$ *for all ordinals* $\alpha \in \mathcal{O}$*, and* $s \approx t$ *iff* $s \approx_\alpha t$ *for all ordinals* $\alpha \in \mathcal{O}$*. That is,* $\sim\ =\ \bigcap_{\alpha\in\mathcal{O}} \sim_\alpha$ *and* $\approx\ =\ \bigcap_{\alpha\in\mathcal{O}} \approx_\alpha$.

Remark 1. *For Part 2 of Theorem 3 we can define an LTS over a singleton alphabet* $\{a\}$ $(a{\neq}\tau)$ *whose state set is* γ *for some ordinal* γ *(that is, each ordinal smaller than* γ *is a state), and such that* $\alpha \xrightarrow{a} \beta$ *iff* $\beta < \alpha$*. Then it is easy to show that for* $\alpha < \beta$, $\alpha \sim_\alpha \beta$ *but* $\alpha \not\sim_{\alpha+1} \beta$*. (First we show, by induction on* α*, that if* $\alpha \leq \mu, \nu$ *then* $\mu \sim_\alpha \nu$*; then we show, by induction on* α*, that if* $\alpha < \beta$ *then* $\alpha \not\sim_{\alpha+1} \beta$*.) As this LTS does not have* τ *actions, and hence* $\approx{=}\sim$*, this also gives that* $\alpha \approx_\alpha \beta$ *but* $\alpha \not\approx_{\alpha+1} \beta$.

If $s \not\sim t$, we must have a least ordinal $\alpha \in \mathcal{O}$ such that $s \not\sim_{\alpha+1} t$, and for this ordinal α we must have $s \sim_\alpha t$. (If $s \not\sim_\lambda t$ for a limit ordinal λ then we must have $s \not\sim_\alpha t$, and hence $s \not\sim_{\alpha+1} t$, for some $\alpha < \lambda$.) We shall identify this value α by writing $s \sim^!_\alpha t$. In the same way we write $s \approx^!_\alpha t$ to identify the least ordinal $\alpha \in \mathcal{O}$ such that $s \not\approx_{\alpha+1} t$.

3.1 Bisimulation Games and Optimal Move Trees

There is a further approach to defining (weak) bisimilarity, one based on games and strategies, whose usefulness is outlined in the tutorial [15]. We describe it here for bisimilarity; its description for weak bisimilarity requires only replacing the transition relation $\rightarrow$ with the weak transition relation $\Rightarrow$, after which all results stated will hold for the weak bisimilarity relations.

A game $\mathcal{G}(s,t)$ corresponding to two states s and t of an LTS is played between two players, **A** and **B**; the first player **A** (the *adversary*) wants to show that the states s and t are different, while the second player **B** (the *bisimulator)* wants to show that they are the same. To this end the game is played by the two players exchanging moves as follows:

- **A** chooses any transition $s \xrightarrow{a} s'$ or $t \xrightarrow{a} t'$ from one of the states s and t;
- **B** responds by choosing a matching transition $t \xrightarrow{a} t'$ or $s \xrightarrow{a} s'$ from the other state;
- the game then continues from the new position $\mathcal{G}(s',t')$.

The second player **B** wins this game if **B** can match every move that the first player **A** makes (that is, if **A** ever cannot make a move or the game continues indefinitely); if, however, **B** at some point cannot match a move made by **A** then player **A** wins. The following is then a straightforward result.

Theorem 4. *$s \sim t$ iff the second player* **B** *has a winning strategy for $\mathcal{G}(s,t)$.*

If $s \not\sim t$, then $s \sim^!_\alpha t$ for some $\alpha \in \mathcal{O}$, and this α in a sense determines how long the game must last, assuming both players are playing optimally, before **B** loses the game $\mathcal{G}(s,t)$:

- Since $s \not\sim_{\alpha+1} t$, **A** can make a move such that, regardless of **B**'s response, the exchange of moves will result in a game $\mathcal{G}(s',t')$ in which $s' \not\sim_\alpha t'$; such a move is an ***optimal move*** for **A**.
- For every $\beta < \alpha$, regardless of the move made by **A**, **B** can respond in such a way that the exchange of moves will result in a game $\mathcal{G}(s',t')$ in which $s' \sim_\beta t'$.

With this in mind, we can make the following definition.

Definition 5. *An* ***optimal move tree*** *is a tree whose nodes are labelled by pairs of non-bisimilar states of an LTS in which an edge $(s,t) \longrightarrow (s',t')$ exists precisely when (s,t) is a node of the tree and the following holds:*

In the game $\mathcal{G}(s,t)$, a single exchange of moves in which **A** *makes an optimal move may result in the game $\mathcal{G}(s',t')$*

The optimal move tree rooted at (s,t) is denoted by ***omt***(s,t).

If $(s,t) \longrightarrow (s',t')$ is an edge in an optimal move tree, then $s \sim^!_\alpha t$ and $s' \sim^!_\beta t'$ for some α and β with $\alpha > \beta$. Hence, every optimal move tree is well-founded. Furthermore, the following result is easily realised.

Lemma 2. $h(omt(s,t)) = \alpha$ *iff* $s \sim^!_\alpha t$.

3.2 Bounded Branching Processes

Over the class of finite-branching labelled transition systems, it is a standard result that $\sim\ =\ \sim_\omega$. We give here a generalisation of this result for infinite-branching processes.

Definition 6. *An infinite cardinal κ is* ***regular*** *iff it is not the supremum of fewer than κ smaller ordinals.*

Thus for example ω is regular as it is not the supremum of any finite collection of natural numbers.

Definition 7. *A process is* ***$<$-κ-branching*** *iff all of its states have fewer than κ transitions leading out of them. A tree t is* ***$<$-κ-branching*** *iff all of its nodes have fewer than κ children.*

Theorem 5. *If κ is a regular cardinal, and t is a well-founded $<$-κ-branching tree, then $h(t) < \kappa$.*

Proof. By (transfinite) induction on $h(t)$. If $t \longrightarrow s$ then $h(s) < h(t)$; and by induction $h(s) < \kappa$ and hence $h(s)+1 < \kappa$. Since $h(t) = \sup\{\, h(s)+1 \;:\; t \longrightarrow s \,\}$, by the regularity of κ we must have that $h(t) < \kappa$. $\square$

The most basic form of this result is König's Lemma: any finite-branching well-founded tree can only have finitely-many nodes (and hence finite height).

The next result follows directly from the fact that $|A \times A| = |A|$ for any infinite set A.

Lemma 3. *If s and t are non-equivalent states of a $<$-κ-branching process, then $omt(s,t)$ is $<$-κ-branching.*

¿From the above, we arrive at a theorem on approximant collapse, which generalises the standard result that $\sim\ =\ \sim_\omega$ on finite-branching processes as well as a result in [22] concerning countably-branching processes.

Theorem 6. *For regular cardinals κ, $\sim\ =\ \sim_\kappa$ over the class of $<$-κ-branching processes.*

Proof. $\sim\ \subseteq\ \sim_\kappa$ is a given. If on the other hand $s \not\sim t$, then $h(omt(s,t)) = \alpha$ where $s \sim^!_\alpha t$. Thus, by Lemma 3 and Theorem 5, $\alpha < \kappa$, and hence $s \not\sim_\kappa t$. $\square$

4 Basic Parallel Processes

A Basic Process Algebra (BPA) process is defined by a context-free grammar in Greibach normal form. Formally this is given by a triple $G = (V, A, \Gamma)$, where V is a finite set of *variables* (*nonterminal symbols*), A is a finite set of *labels* (*terminal symbols*), and $\Gamma \subseteq V \times A \times V^*$ is a finite set of *rewrite rules* (*productions*); it is assumed that every variable has at least one associated rewrite rule. Such a grammar gives rise to the LTS $\mathcal{S}_G = (V^*, A, \rightarrow)$ in which the states are sequences of variables, the actions are the labels, and the transition relation is given by the rewrite rules extended by the *prefix rewriting rule*: if $(X, a, u) \in \Gamma$ then $Xv \xrightarrow{a} uv$ for all $v \in V^*$. In this way, concatenation of variables naturally represents sequential composition.

A Basic Parallel Processes (BPP) process is defined in exactly the same fashion from such a grammar. However, in this case elements of V^* are read modulo commutativity of concatenation, so that concatenation is interpreted as parallel composition rather than sequential composition. The states of the BPP process associated with a grammar are thus given not by sequences of variables but rather by multisets of variables.

As an example, Figure 1 depicts BPA and BPP processes defined by the same

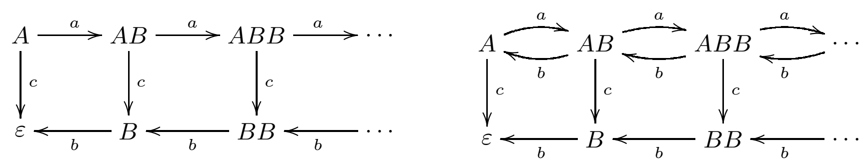

Fig. 1. BPA and BPP processes defined by the grammar $A \xrightarrow{a} AB$, $A \xrightarrow{c} \varepsilon$, $B \xrightarrow{b} \varepsilon$

grammar given by the three rules $A \xrightarrow{a} AB$, $A \xrightarrow{c} \varepsilon$ and $B \xrightarrow{b} \varepsilon$.

Decidability results for (strong) bisimilarity checking have been long established for both BPA [5] and BPP [3, 4]. For a wide class of interest (normed processes) these problems have even been shown to have polynomial-time solutions [11–13]. More recently, the decision problems for full BPA and BPP have been shown to be PSPACE-hard [17, 18].

Decidability results for weak bisimilarity are much harder to establish, mainly due to the problems of infinite branching. While over BPA and BPP we have $\sim \; = \bigcap_{n\in\omega} \sim_n$, the infinite-branching nature of the weak transition relations makes this result false. As an example, Figure 2 gives a BPP process with states P and Q in which $P \approx_n Q$ for all $n \in \omega$ yet $P \not\approx Q$. In this case we have $P \approx^!_\omega Q$, but from these we can produce BPP process states X_n and Y_n such that $X_n \approx^!_{\omega+n} Y_n$ by adding the following production rules to the defining grammar:

$$X_1 \xrightarrow{a} P \qquad X_{i+1} \xrightarrow{a} X_i \qquad Y_1 \xrightarrow{a} Q \qquad Y_{i+1} \xrightarrow{a} Y_i$$

$A \xrightarrow{a} A \qquad P \xrightarrow{\tau} A \qquad Q \xrightarrow{a} \varepsilon$

$\qquad\qquad P \xrightarrow{\tau} Q \qquad Q \xrightarrow{\tau} QQ$

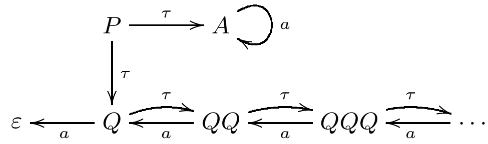

Fig. 2. A BPP process with states P and Q satisfying $P \approx^!_\omega Q$

However, no example BPP states X and Y are known which satisfy $X \approx^!_{\omega\times 2} Y$. This leads to the following long-standing conjecture.

Conjecture (Hirshfeld, Jančar). *Over BPP processes,* $\approx \, = \, \approx_{\omega\times 2}$.

Remark 2. *The situation is different for BPA, as noted originally in [22]: for any* $\alpha<\omega^\omega$, *we can construct BPA processes* P *and* Q *for which* $P \approx^!_\alpha Q$. *To see this, we consider the grammar* $G = (V, A, \Gamma)$ *in which* $V = \{X_0, X_1, \ldots, X_{n-1}\}$, $A = \{a, \tau\}$, *and* Γ *consists of the following rules:*

$$X_0 \xrightarrow{a} \varepsilon \qquad X_i \xrightarrow{\tau} \varepsilon \qquad X_{i+1} \xrightarrow{\tau} X_{i+1}X_i$$

For each $\alpha<\omega^n$, *with Cantor normal form*

$$\alpha \;=\; \omega^{n-1}a_{n-1} \;+\; \cdots \;+\; \omega^2 a_2 \;+\; \omega a_1 \;+\; a_0,$$

let $P_\alpha = X_0^{a_0}X_1^{a_1}X_2^{a_2}\cdots X_{n-1}^{a_{n-1}}$. *We can show that, for* $\alpha<\beta<\omega^n$, $P_\alpha \approx^!_\alpha P_\beta$. *This will follow from the following sequence of observations which demonstrate a close analogy between the processes* P_α *and the ordinal processes* α *from Remark 1:*

- *If* $P \approx Q$ *then* $RP \approx RQ$ *and* $PR \approx QR$. *The first conclusion is true for every BPA process, while the second conclusion is true for every BPA process in which* $P \overset{\tau}{\Rightarrow} \varepsilon$ *whenever* $P \approx \varepsilon$, *which is certainly the case for the BPA process under consideration since in this case* $P \approx \varepsilon$ *implies that* $P = \varepsilon$. *(Proof:* $\{(RP, RQ) : P \approx Q\}$ *and* $\{(PR, QR) : P \approx Q\}$ *are easily verified to be weak bisimulation relations.)*
- *For* $i>j$: $X_iX_j \approx X_i$. *(Proof: by induction on* $i-j$.*)*
- *Every state* $P \in V^*$ *is weakly bisimilar to some state* P_α. *(Proof: follows directly from the above observations.)*
- $X_k \overset{\tau}{\Rightarrow} P_\alpha$ *for every* $\alpha\leq\omega^k$. *(Proof: easily verified.)*
- $P_\alpha \overset{\tau}{\Rightarrow} P_\beta$ *for every* $\beta\leq\alpha$. *(Proof: generalisation of the above observation.)*
- $P_\alpha \overset{a}{\Rightarrow} P_\beta$ *for every* $\beta<\alpha$. *(Proof: follows from the previous observation and the fact that* $P_{\beta+1} \xrightarrow{a} P_\beta$.*)*
- *If* $P_\alpha \overset{\tau}{\Rightarrow} P$ *then* $P \approx P_\beta$ *for some* $\beta\leq\alpha$. *(Proof: easily verified.)*
- *If* $P_\alpha \overset{a}{\Rightarrow} P$ *then* $P \approx P_\beta$ *for some* $\beta<\alpha$. *(Proof: again easily verified.)*

We thus arrive at the following important observations about the states P_α:

- $P_\alpha \stackrel{\tau}{\Rightarrow} P_\beta$ *for all* $\beta \le \alpha$*, and if* $P_\alpha \stackrel{\tau}{\Rightarrow} P$ *then* $P \approx P_\beta$ *for some* $\beta \le \alpha$*; and*
- $P_\alpha \stackrel{a}{\Rightarrow} P_\beta$ *for all* $\beta < \alpha$*, and if* $P_\alpha \stackrel{a}{\Rightarrow} P$ *then* $P \approx P_\beta$ *for some* $\beta < \alpha$*.*

This suffices to deduce with little effort, analogously to Remark 1, that if $\alpha < \beta$ *then* $P_\alpha \approx^!_\alpha P_\beta$*. (The conjecture for BPA, though, is that the bound given by this construction is tight:* $\approx = \approx_{\omega^\omega}$*.)*

BPP processes with silent moves are countably-branching, and thus by Theorem 6 $\approx = \approx_{\aleph_1}$. In [22] there is an argument attributed to J. Bradfield which shows that the approximation hierarchy collapses by the level $\approx_{\omega_1^{CK}}$, the first non-recursive ordinal. (The argument is made there for BPA but clearly holds as well for BPP.) But this is to measure in lightyears what should require centimetres; we proceed here to a more modest bound, based on our ordinal analysis of Dickson's Lemma.

We assume an underlying grammar (V, A, Γ) defining our BPP process, and recall that a state in the associated process is simply a sequence $u \in V^*$ viewed as a multiset. With this, we make the important observation about weak bisimulation approximants over BPP: besides being equivalences, they are in fact congruences.

Lemma 4. *For all* $u, v, w \in V^*$*, if* $u \approx_\alpha v$ *then* $uw \approx_\alpha vw$*.*

Proof. By a simple induction on α. □

We next observe a result due to Hirshfeld [10].

Lemma 5. *If* $u \approx^!_\alpha v$ *and* $uu' \approx^!_\beta vv'$ *with* $\beta < \alpha$ *then* $uu' \approx^!_\beta uv'$ *and* $vu' \approx^!_\beta vv'$*.*

Proof. $uu' \approx_\beta uv'$ since $uu' \approx_\beta vv' \approx_\alpha uv'$. On the other hand, if $uu' \approx_{\beta+1} uv'$ then $uu' \approx_{\beta+1} uv' \approx_\alpha vv'$. Thus $uu' \approx^!_\beta uv'$. ($vu' \approx^!_\beta vv'$ can be shown similarly). □

BPP processes, being multisets over the finite variable set V, can be represented as $|V|$-tuples over $\mathbb{N}$. Given non-equivalent BPP states u_0 and v_0, $omt(u_0, v_0)$ can then be viewed as a $\mathbb{N}^{2\cdot|V|}$-labelled tree. In general this tree will not be non-dominating, but the above lemma will enable us to produce a non-dominating $\mathbb{N}^{2\cdot|V|}$-labelled tree from $omt(u_0, v_0)$

Lemma 6. *For BPP processes, if* $u_0 \approx^!_\alpha v_0$ *then there exists a* $\mathbb{N}^{2\cdot|V|}$*-labelled non-dominating tree of height* α*.*

Proof. We apply the following substitution procedure to each successive level of the weak-transition optimal move tree $omt(u_0, v_0)$ (where the *level* of a node refers to the distance from the root (u_0, v_0) to the node) by induction on the levels:

For each node x at this level, if x dominates some ancestor node y, that is, if there exists an ancestor node $y = (u, v)$ where $x = (uu', vv')$, then replace the subtree rooted at x with either $x' = omt(uu', uv')$ (if $u <_{\text{lex}} v$) or with $x' = omt(vu', vv')$ (if $v <_{\text{lex}} u$). (If this x' itself then dominates an ancestor node, repeat this action.)

That $<_{\text{lex}}$ is a well-founded relation on $\mathbb{N}^{2\cdot|V|}$ means this repetition must halt; and Lemma 5 implies that this is a height-preserving operation. □

Theorem 7. *Over BPP processes,* $\approx \, = \, \approx_{\omega^\omega}$

Proof. If $u \approx v$ then $u \approx_{\omega^\omega} v$ is a given. If, on the other hand, $u \not\approx v$, then $u \approx^!_\alpha v$ for some α, and by the combination of Lemma 6 and Theorem 2 we must have that $\alpha \leq \omega^{2\cdot|V|}$. Thus, $u \not\approx_{\omega^\omega} v$. □

5 Conclusions

In this paper we provide a bound on the level at which the bisimulation approximation relations collapse over BPP. The bound we give of ω^ω is still a far cry from the widely-accepted conjectured bound of $\omega\times 2$, but it nonetheless represents the first nontrivial countable bound that has been discovered in the decade since this conjecture was first uttered (originally by Hirshfeld and Jančar).

We arrive at our bound through a careful analysis of Dickson's Lemma, and in particular via a novel constructive proof which provides this ordinal bound on non-dominating sequences of n-tuples. (Dickson's Lemma itself merely declares that such sequences are necessarily finite without actually identifying any ordinal bound.) This approach does not immediately seem to be applicable to strengthening the bound, given that this bound on Dickson's Lemma is tight. However, it seems equally likely that by taking into consideration the restricted form of non-dominating sequences produced by BPP transitions we can identify the missing ingredient for the proof of the tighter bound.

There have been other similar constructive proofs of Dickson's Lemma in the area of term rewriting. In particular, Sustik [23] provides a similar proof using an ordinal mapping on sequences in order to mechanically prove Dickson's Lemma using the ACL2 theorem prover. However, the ordinal mapping defined by Sustik gives an inferior bound to the one we provide; in particular, it requires ω^ω already for sequences of pairs.

Blass and Gurevich have very recently (preprint March 2006) written a manuscript [1] in which they define the *stature* of a well partial ordering P to be the order type of nondominating sequences of P, and (amongst other things) derive the same tight bound of ω^n as we have done. Their application of interest lies in program termination, and their proofs, being of more general results, are more complicated than the proof we provide. We therefore feel that our proof, which appeared in an earlier mauscript [9], as well as our application to bisimulation checking is of independent interest.

If the $\omega\times 2$ bound for the weak bisimulation approximation relations over BPP is resolved positively, this can potentially be exploited to resolve the decidability of weak bisimilarity over BPP. Esparza in [7] has shown that weak equivalence is semi-decidable, by demonstrating a semilinear witness of equivalence, so semi-decidability of non-equivalence is all that is required. If $\approx_\omega$ can be shown to be decidable (which is likely a much simpler result to attain than for $\approx$) then it would naturally be expected that the successor relations $\approx_{\omega+1}, \approx_{\omega+1}, \ldots$ would also be decidable, which would give rise to a semi-decision procedure for $\not\approx_{\omega\times 2}$: test each relation $\approx_{\omega+i}$ in turn until one such test fails.

References

1. A. Blass and Y. Gurevich. Program termination and well partial orderings. Microsoft Technical Report MSR-TR-2006-27, 31 pages, March 2006. (Available at ftp://ftp.research.microsoft.com/pub/tr/TR-2006-27.pdf.)
2. O. Burkart, D. Caucal, F. Moller and B. Steffen. Verification over Infinite States. Chapter 9 in the *Handbook of Process Algebra*, pages 545-623, Elsevier Publishers, 2001.
3. S. Christensen, Y. Hirshfeld, and F. Moller. Bisimulation equivalence is decidable for basic parallel processes. In *Proceedings of the 4th International Conference on Concurrency Theory (CONCUR'93)*, LNCS 715, pages 143-157, Springer, 1993.
4. S. Christensen, Y. Hirshfeld, and F. Moller. Decomposability, decidability and axiomatisability for bisimulation equivalence on basic parallel processes. In *Proceedings of the 8th Annual IEEE Symposium on Logic in Computer Science (LICS'93)*, pages 386-396, IEEE Computer Society Press, 1993.
5. S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. In *Proceedings of the 3rd International Conference on Concurrency Theory (CONCUR'92)*, LNCS 630, pages 138-147, Springer, 1992.
6. L.E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with distinct factors. *American Journal of Mathematics* 3:413-422, 1913.
7. J. Esparza. Petri nets, commutative context-free grammars, and basic parallel proceses. *Fundamenta Informaticae* 30:23-41, 1997.
8. R. Glabbeek. The linear time – branching time spectrum I: The semantics of concrete sequential processes. Chapter 1 in the *Handbook of Process Algebra*, pages 3-99, Elsevier Publishers, 2001.
9. W. Harwood and F. Moller. Weak bisimulation approximants. In *Selected Papers from the CALCO Young Researchers Workshop (CALCO-jnr 2005)*, Swansea University Research Report CSR 18-2005, pages 27-40, December 2005. (Available at http://www-compsci.swan.ac.uk/reports/2005.html.)
10. Y. Hirshfeld. Bisimulation trees and the decidability of weak bisimulation. *Electronic Notes in Theoretical Computer Science* 5:2-13, 1997.
11. Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS'94)*, pages 623-631, IEEE Computer Society Press, 1994.
12. Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theoretical Computer Science* 15:143-159, 1996.

13. Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimulation equivalence of normed basic parallel processes. *Mathematical Structures in Computer Science* 6:251-259, 1996.
14. P. Jančar. Strong bisimilarity on Basic Parallel Processes is PSPACE-complete. In *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 218-227, IEEE Computer Society Press, 2003.
15. P. Jančar and F. Moller. Techniques for decidability and undecidability of bisimilarity. In *Proceedings of the 10th International Conference on Concurrency Theory (CONCUR'99)*, LNCS 1664, pages 30-45, Springer, 1999.
16. F. Moller. Infinite Results. In *Proceedings of the 7th International Conference on Concurrency Theory (CONCUR'96)*, LNCS 1119, pages 195-216, Springer, 1996.
17. J. Srba. Strong bisimilarity and regularity of Basic Process Algebra is PSPACE-hard. In *Proceedings of the 29th International Conference on Automata, Languages and Programming (ICALP'02)*, LNCS 2380, pages 716-727, Springer, 2002.
18. J. Srba. Strong bisimilarity and regularity of Basic Parallel Processes is PSPACE-hard. In *Proceedings of the 19th International Symposium on Theoretical Aspects of Computer Science (STACS'02)*, LNCS 2285, pages 535-546, Springer, 2002.
19. J. Srba. Complexity of weak bisimilarity and regularity for BPA and BPP. *Mathematical Structures in Computer Science* 13:567-587, 2003.
20. J. Srba. Roadmap of Infinite Results. http://www.brics.dk/~srba/roadmap.
21. C. Stirling. Decidability of weak bisimilarity for a subset of Basic Parallel Processes. In *Proceedings of Foundations of Software Science and Computation Structures (FOSSACS'01)*, LNCS 2030, pages 379-393, 2001.
22. J. Stříbrná. Approximating weak bisimulation on Basic Process Algebras. In *Proceedings of Mathematical Foundations of Computer Science (MFCS'99)*, LNCS 1672, pages 366-375, 1999.
23. M. Sustik. Proof of Dickson's Lemma using the ACL2 theorem prover via an explicit ordinal mapping. Unpublished presentation at the *Fourth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2-2003)*. (Available from http://www.cs.utexas.edu/users/moore/acl2/workshop-2003.)