BRICS RS-97-26 Aceto & Ingólfsdóttir: A Characterization of Finitary Bisimulation

BRICS

Basic Research in Computer Science

A Characterization of Finitary Bisimulation

Luca Aceto
Anna Ingólfsdóttir

BRICS Report Series RS-97-26
ISSN 0909-0878 October 1997

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/97/26/`

A Characterization of Finitary Bisimulation

Luca Aceto*
BRICS†, Department of Computer Science, Aalborg University
Fredrik Bajersvej 7E, 9220 Aalborg Ø, Denmark
Email: `luca@cs.auc.dk`

Anna Ingólfsdóttir‡
Dipartimento di Sistemi ed Informatica, Università di Firenze
Via Lombroso 6/17, 50134 Firenze, Italy
Email: `annai@dsi2.ing.unifi.it`

Keywords and Phrases: Concurrency, labelled transition system with divergence, bisimulation preorder, finitary relation.

1 Introduction

Following a paradigm put forward by Milner and Plotkin, a primary criterion to judge the appropriateness of denotational models for programming and specification languages is that they be in agreement with operational intuition about program behaviour. Of the "good fit" criteria for such models that have been discussed in the literature, the most desirable one is that of *full abstraction*. Intuitively, a fully abstract denotational model is guaranteed to relate exactly all those programs that are operationally indistinguishable with respect to some chosen notion of observation.

Because of its prominent role in process theory, bisimulation [12] has been a natural yardstick to assess the appropriateness of denotational models for several process description languages. In particular, when proving full abstraction results for denotational semantics based on the Scott-Strachey approach for CCS-like languages, several preorders based on bisimulation have been considered; see, e.g., [6, 3, 4]. In this paper, we shall study one such bisimulation-based preorder whose connections with domain-theoretic models are by now well understood, viz. the *prebisimulation* preorder $\lesssim$ investigated in, e.g., [6, 3]. Intuitively, $p \lesssim q$ holds of processes p and q if p and q can simulate each other's behaviour, but at times the behaviour of p may be less specified than that of q.

A common problem in relating denotational semantics for process description languages, based on Scott's theory of domains or on the theory of algebraic semantics, with behavioural semantics based on bisimulation is that the chosen behavioural theory is, in general, too concrete. The reason for this phenomenon

*Partially supported by the Human Capital and Mobility project Express.
†Basic Research in Computer Science, Centre of the Danish National Research Foundation.
‡Supported by a grant from the Danish Research Council.

is that two programs are related by a standard denotational interpretation if, in some precise sense, they afford the same *finite* observations. On the other hand, bisimulation can make distinctions between the behaviours of two processes based on *infinite* observations. (Cf. the seminal study [1] for a detailed analysis of this phenomenon.) To overcome this mismatch between the denotational and the behavioural theory, all the aforementioned full abstraction results are obtained with respect to the so-called *finitely observable*, or *finitary*, part of bisimulation. The *finitary bisimulation* is defined on any labelled transition system thus: $p \lesssim^F q$ iff $t \lesssim p$ implies $t \lesssim q$, for every finite synchronization tree t.

An alternative characterization of the finitary bisimulation for arbitrary transition systems has been given by Abramsky in [1]. This characterization is couched in logical terms, and is an impressive byproduct of Abramsky's "theory of domains in logical form" programme. More precisely, Abramsky shows that two processes in any transition system are equated by the finitary bisimulation iff they satisfy the same formulae in the finitary version of the domain logic for transition systems. The existence of this logical view of the finitary bisimulation gives us a handle to work with this relation. However, an alternative, *behavioural* view of the finitary bisimulation might be more useful when establishing results which are more readily shown on the behavioural, rather than on the logical, side. Examples of such results are complete axiomatizations for the finitary bisimulation and full abstraction results. A behavioural characterization of the finitary bisimulation would also provide an easier way to establish when two processes in a transition system are related by it or not, thus giving more insight on the kind of identifications made by this relation.

In this study, we offer a behavioural characterization of the finitary bisimulation for arbitrary transition systems (cf. Thm. 3.5). This result may be seen as the behavioural counterpart of Abramsky's logical characterization theorem [1, Thm. 5.5.8]. Moreover, for the important class of sort-finite transition systems we present a sharpened version of a behavioural characterization result first proven by Abramsky in [3, Propn. 6.13]. The interested reader may consult the unpublished report [5] for more results on the finitary bisimulation.

2 Preliminaries

We begin by reviewing a variation on the model of labelled transition systems [9] that takes divergence information into account. We refer the interested readers to, e.g., [11] for motivation and more information on this semantic model for reactive systems. A *labelled transition system with divergence* (lts) is a quadruple $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$, where:

- Proc is a set of *processes*, ranged over by p, q, r, s, possibly subscripted or superscripted;
- Act is a set of *actions*, ranged over by a, b, possibly subscripted;
- $\rightarrow \subseteq \mathsf{Proc} \times \mathsf{Act} \times \mathsf{Proc}$ is a *transition relation*. As usual, we shall use the more suggestive notation $p \stackrel{a}{\rightarrow} q$ in lieu of $(p, a, q) \in \rightarrow$;

- $\uparrow \subseteq \mathsf{Proc}$ is a *divergence predicate*, notation $p\uparrow$.

We write $p\downarrow$, read "p converges", iff it is not the case that $p\uparrow$. Intuitively, the fact that a process p converges means that its initial behaviour is completely specified. On the contrary, the divergence of a process signifies that the information on its initial behaviour is incomplete.

For $n \geq 0$ and $\sigma = a_1 \ldots a_n \in \mathsf{Act}^*$, we write $p \stackrel{\sigma}{\to} q$ iff there exist processes $p_0, \ldots, p_n$ such that $p = p_0 \stackrel{a_1}{\to} p_1 \stackrel{a_2}{\to} \cdots p_{n-1} \stackrel{a_n}{\to} p_n = q$. For a process $p \in \mathsf{Proc}$ and action $a \in \mathsf{Act}$ we define:

$$\begin{aligned}
\mathsf{initials}(p) &\triangleq \left\{a \in \mathsf{Act} \mid \exists q:\ p \stackrel{a}{\to} q\right\} \\
\mathsf{sort}(p) &\triangleq \left\{a \in \mathsf{Act} \mid \exists \sigma \in \mathsf{Act}^*, r, s \in \mathsf{Proc}:\ p \stackrel{\sigma}{\to} r \stackrel{a}{\to} s\right\} \\
\mathsf{derivatives}(p,a) &\triangleq \left\{q \mid p \stackrel{a}{\to} q\right\}\ .
\end{aligned}$$

Following [3], we say that an lts is *sort-finite* iff $\mathsf{sort}(p)$ is finite for every $p \in \mathsf{Proc}$.

A useful source of examples for labelled transition systems with divergence is the set of countably branching synchronization trees over a set of labels Act, denoted by $\mathsf{ST}_\infty(\mathsf{Act})$. This is the set of infinitary terms generated by the inductive definition:

$$\frac{\{a_i \in \mathsf{Act}, t_i \in \mathsf{ST}_\infty(\mathsf{Act})\}_{i \in I}}{\sum_{i \in I} a_i : t_i[+\Omega] \in \mathsf{ST}_\infty(\mathsf{Act})}$$

where I is a countable index set, and the notation $[+\Omega]$ means optional inclusion of Ω as a summand. We shall write $\mathbb{O}$ for $\sum_{i \in \varnothing} a_i : t_i$, and Ω for $\sum_{i \in \varnothing} a_i : t_i + \Omega$. Intuitively, $\mathbb{O}$ stands for the one-node synchronization tree, a representation of an inactive process, and Ω stands for the synchronization tree whose behaviour is completely unspecified.

The set of terms built using only finite summations, i.e. the finite synchronization trees, will be denoted by $\mathsf{ST}(\mathsf{Act})$. The set of synchronization trees $\mathsf{ST}_\infty(\mathsf{Act})$ can be turned into a labelled transition system with divergence by stipulating that, for $t \in \mathsf{ST}_\infty(\mathsf{Act})$:

- $t\uparrow$ iff Ω is a summand of t, and
- $t \stackrel{a_i}{\to} t_i$ iff $a_i : t_i$ is a summand of t.

The behavioural relation over processes that we shall study in this paper is that of *prebisimulation* [8, 11, 6, 13] (also known as *partial bisimulation* [3]).

Definition 2.1 *Let* $(\mathsf{Proc}, \mathsf{Act}, \to, \uparrow)$ *be an lts. Let* $\mathsf{Rel}(\mathsf{Proc})$ *denote the set of binary relations over* Proc. *Define the functional* $F : \mathsf{Rel}(\mathsf{Proc}) \to \mathsf{Rel}(\mathsf{Proc})$ *by:*

$$\begin{aligned}
F(\mathcal{R}) \quad = \quad \{(p,q) \mid \forall a \in \mathsf{Act}\ & \\
& \bullet\ p \stackrel{a}{\to} p' \Rightarrow \exists q':\ q \stackrel{a}{\to} q' \ \textit{and}\ p'\,\mathcal{R}\,q' \\
& \bullet\ p\downarrow \Rightarrow \left(q\downarrow\ \textit{and}\ \left(q \stackrel{a}{\to} q' \Rightarrow \exists p':\ p \stackrel{a}{\to} p' \ \textit{and}\ p'\,\mathcal{R}\,q'\right)\right)\}
\end{aligned}$$

A relation $\mathcal{R}$ *is a* prebisimulation *iff* $\mathcal{R} \subseteq F(\mathcal{R})$. *We write* $p \lesssim q$ *iff there exists a prebisimulation* $\mathcal{R}$ *such that* $p\,\mathcal{R}\,q$.

An alternative method for using the functional F to obtain a behavioural preorder is to apply it inductively as follows:

$$\begin{aligned} \lesssim_0 &\triangleq \mathsf{Proc} \times \mathsf{Proc} \\ \lesssim_{n+1} &\triangleq F(\lesssim_n) \end{aligned}$$

and finally $\lesssim_\omega \triangleq \bigcap_{n\geq 0} \lesssim_n$. Intuitively, the preorder $\lesssim_\omega$ is obtained by restricting the prebisimulation relation to observations of finite depth. As a standard example of the relevance of this restriction, consider the processes

$$\begin{aligned} p &\triangleq \left(\textstyle\sum_{i\geq 1} \underbrace{a : \cdots : a :}_{i\text{-times}} \mathbb{O} \right) + \Omega \\ q &\triangleq p + a^\omega \end{aligned}$$

where a^ω denotes an infinite sequence of a actions. Then $q \not\lesssim p$ because the transition $q \xrightarrow{a} a^\omega$ cannot be matched by any a-transition emanating from p. On the other hand, it is easy to see that $q \lesssim_\omega p$ does hold.

In this paper, we are interested in studying the "finitely observable", or *finitary*, part of the bisimulation in the sense of, e.g., [6]. The following definition is from [3].

Definition 2.2 *Let* $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ *be an lts. The* finitary bisimulation preorder $\lesssim^F$ *over* Proc *is defined as follows:* $p \lesssim^F q$ *iff, for every* $t \in \mathsf{ST}(\mathsf{Act})$, $t \lesssim p$ *implies* $t \lesssim q$.

The preorders $\lesssim$, $\lesssim_\omega$ and $\lesssim^F$ are related thus: $\lesssim \;\subseteq\; \lesssim_\omega \;\subseteq\; \lesssim^F$. Moreover the inclusions are strict for infinitely branching lts's, and collapse to equalities for finitely branching ones. The interested reader is referred to [3] for a wealth of examples distinguishing these preorders, and a very deep analysis of their general relationships and properties.

3 The Behavioural Characterization

Abramsky's logical characterization of the finitary bisimulation provides one general, observation-independent alternative view of $\lesssim^F$. It can be viewed as the counterpart of the modal characterization theorems for bisimulation-based equivalences and preorders which have been so popular and fruitful since the seminal [11, 7]. However, in order to gain more insight into the exact nature of the relationships between processes supported by $\lesssim^F$, and as a further tool for the study of this preorder (for example to establish results on full abstraction of denotational models and complete axiomatizations), it is useful to have purely behavioural, observation-independent characterizations of it. One such characterization was provided by Abramsky in, e.g., [3, Propn. 6.13]. There Abramsky shows that in any sort-finite lts that satisfies his axiom scheme of bounded nondeterminacy (BN) (cf. [1, Page 114]), the finitary bisimulation coincides with $\lesssim_\omega$. We shall now present a bisimulation-like characterization of the finitary bisimulation for arbitrary transition systems. As a byproduct of our analysis of the finitary bisimulation, we shall be able to improve upon Abramsky's behavioural characterization of $\lesssim^F$ for sort-finite lts's. (Cf. Propn. 3.6.)

Consider an arbitrary lts $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$. For every $A \subseteq \mathsf{Act}$, we define the sequence of relations $\left\{ \sqsubseteq_n^A \mid n \geq 0 \right\}$ thus:

$$
\begin{array}{lcl}
p \sqsubseteq_0^A q & \Leftrightarrow & \text{true} \\[1ex]
p \sqsubseteq_{n+1}^A q & \Leftrightarrow & (1)\ \ \forall a \in A, p' \in \mathsf{Proc}.\ p \xrightarrow{a} p' \Rightarrow \exists q' :\ q \xrightarrow{a} q' \text{ and } p' \sqsubseteq_n^A q' \\
 & & (2)\ \ \text{If } \mathsf{initials}(p) \subseteq A \text{ and } p\downarrow \text{ then} \\
 & & \ \ (2.1)\ \ \mathsf{initials}(q) \subseteq A \text{ and } q\downarrow \\
 & & \ \ (2.2)\ \ \forall a \in A, q' \in \mathsf{Proc}.\ q \xrightarrow{a} q' \Rightarrow \exists p' :\ p \xrightarrow{a} p' \text{ and } p' \sqsubseteq_n^A q' \ .
\end{array}
$$

The following proposition collects some basic properties of the relations $\sqsubseteq_n^A$ which will be useful in the remainder of this study.

Proposition 3.1 *For every $n \geq 0$ and $A \subseteq \mathsf{Act}$, the following statements hold:*

1. *The relation $\sqsubseteq_n^A$ is a preorder.*
2. *For $p, q \in \mathsf{Proc}$, $p \sqsubseteq_{n+1}^A q$ implies $p \sqsubseteq_n^A q$.*
3. *Assume that $A \subseteq B \subseteq \mathsf{Act}$. Then, for $p, q \in \mathsf{Proc}$, $p \sqsubseteq_n^B q$ implies $p \sqsubseteq_n^A q$.*

We now define:

$$
\begin{array}{rcl}
p \sqsubseteq_\omega^A q & \triangleq & \forall n \geq 0.\ p \sqsubseteq_n^A q \\
p \sqsubseteq_\omega^{fin} q & \triangleq & \forall A \subseteq_{fin} \mathsf{Act}.\ p \sqsubseteq_\omega^A q
\end{array}
$$

where the notation $A \subseteq_{fin} \mathsf{Act}$ means that A is a *finite* subset of Act. Note that, in light of Propn. 3.1(1), both the relations defined above are preorders. As $\mathsf{initials}(p)$ is contained in Act for every process $p \in \mathsf{Proc}$, the preorder $\sqsubseteq_\omega^{\mathsf{Act}}$ coincides with $\lesssim_\omega$.

The above definitions are inspired by [8, page 266], where a preorder over value-passing CCS [10] which uses finite sets of communication capabilities of processes in a similar manner is presented. Intuitively, $p \sqsubseteq_\omega^A q$ holds for two processes p and q iff there is no observation, in the sense of [2], of finite depth, and with actions drawn from the set A, that can distinguish between p and q. For example, $p \sqsubseteq_\omega^\varnothing q$ holds unless p is a convergent inactive process and q is either divergent or capable of performing some action. A similar intuition applies to the relation $\sqsubseteq_\omega^{fin}$, but there observations can only be drawn from *finite* sets of actions and are therefore required to have finite width as well as finite depth. That this is significant is shown by the following example:

Example: Assume that $\mathsf{Act} = \{a_i \mid i \geq 0\}$, and that $i \neq j$ implies $a_i \neq a_j$. Consider the synchronization trees p and q given by:

$$
p \triangleq \sum_{i \geq 0} a_i : \mathbb{O} \qquad (1)
$$

$$
q \triangleq p + \Omega \ . \qquad (2)
$$

Then $p \not\lesssim_1 q$ because $p\downarrow$ but $q\uparrow$. However, $p \sqsubseteq_\omega^{fin} q$. In fact, every transition from p can be matched identically by q, and, for $A \subseteq_{fin} \mathsf{Act}$, clause (2) in the definition of $\sqsubseteq_{n+1}^A$ is always vacuously satisfied because $\mathsf{initials}(p) = \mathsf{Act}$, which is countably infinite.

Indeed, it is also the case that $p \lesssim^F q$. In fact, let $t \in \mathsf{ST}(\mathsf{Act})$ be such that $t \lesssim p$. We shall now argue that $t \lesssim q$ must also hold. First of all, note that $t \lesssim p$ implies that $t\uparrow$. (This is easy to see because otherwise the finite synchronization tree t would have to have Act as its set of initial actions.) Next we remark that if $t \xrightarrow{a} t'$ for some action a, then $t' \lesssim \mathbb{O}$. From these two observations, it follows immediately that $t \lesssim q$. (*End of Example*)

Let $(\mathsf{Proc}, \mathsf{Act}, \to, \uparrow)$ be an arbitrary lts. Our aim is to show that $\lesssim^F$ coincides with $\sqsubseteq_\omega^{fin}$ over Proc. We begin by establishing two auxiliary results.

Lemma 3.2 *For every $t \in \mathsf{ST}(\mathsf{Act}), p \in \mathsf{Proc}$, $t \lesssim p$ iff $t \sqsubseteq_\omega^{fin} p$*

Proof: The 'only if' implication is an immediate consequence of the following fact, which may be easily shown by mathematical induction on n,

$$\forall t \in \mathsf{ST}(\mathsf{Act}), p \in \mathsf{Proc}, n \geq 0, A \subseteq \mathsf{Act}.\ t \lesssim p \ \Rightarrow\ t \sqsubseteq_n^A p\ .$$

Here we just remark that if $t \lesssim p$, $t\downarrow$ and $\mathsf{initials}(t) \subseteq A$, then the definition of $\lesssim$ yields immediately that $p\downarrow$ and $\mathsf{initials}(p) \subseteq A$.

In the proof of the 'if' implication, we shall make use of the notion of *height* of a synchronization tree. This is the ordinal defined thus:

$$\mathsf{ht}(\textstyle\sum_{i\in I} a_i : t_i[+\Omega]) \ \triangleq\ \sup\{\mathsf{ht}(t_i) \mid i \in I\} + 1\ .$$

To establish the 'if' implication, it is sufficient to prove the following statement:

$$\forall t \in \mathsf{ST}(\mathsf{Act}), p \in \mathsf{Proc}.\ t \sqsubseteq_{\mathsf{ht}(t)}^{\mathsf{sort}(t)} p \ \Rightarrow\ t \lesssim p\ .$$

The straightforward proof is by complete induction on $\mathsf{ht}(t)$. Here we limit ourselves to showing that

$$t \sqsubseteq_{\mathsf{ht}(t)}^{\mathsf{sort}(t)} p \text{ and } t \xrightarrow{a} t' \quad \text{implies} \quad p \xrightarrow{a} p' \text{ for some } p' \text{ such that } t' \lesssim p'\ .$$

To this end, assume that $t \sqsubseteq_{\mathsf{ht}(t)}^{\mathsf{sort}(t)} p$ and $t \xrightarrow{a} t'$. Under these assumptions, $\mathsf{ht}(t) > 0$ and it follows that $p \xrightarrow{a} p'$ for some p' such that $t' \sqsubseteq_{\mathsf{ht}(t)-1}^{\mathsf{sort}(t)} p$. It is easy to see that $\mathsf{ht}(t') \leq \mathsf{ht}(t) - 1$ and $\mathsf{sort}(t') \subseteq \mathsf{sort}(t)$. Thus, Propn. 3.1(2)-(3) yields $t' \sqsubseteq_{\mathsf{ht}(t')}^{\mathsf{sort}(t')} p'$. As $\mathsf{ht}(t') < \mathsf{ht}(t)$, the induction hypothesis gives $t' \lesssim p'$ as required. □

The finitary bisimulation has the property that, for all $p, q \in \mathsf{Proc}$,

$$p \lesssim^F q \quad \text{iff} \quad \text{for every } t \in \mathsf{ST}(\mathsf{Act}),\ \ t \lesssim^F p \text{ implies } t \lesssim^F q\ . \tag{3}$$

This is an immediate consequence of the fact that $t \lesssim p$ iff $t \lesssim^F p$, for every finite synchronization tree t and process p. A binary relation over processes that enjoys property (3) is usually called *finitary* or *finitely approximable* [6, 4].

Lemma 3.3 *The preorder $\sqsubseteq_{\omega}^{fin}$ is finitary.*

Before proving this lemma, we introduce some intermediate definitions and results. For every process $p \in \mathsf{Proc}$, *finite* action set A and non-negative integer n, we define a synchronization tree $p^{(A,n)}$ as follows:

$$\begin{array}{rcl} p^{(A,0)} & \triangleq & \Omega \\ p^{(A,n+1)} & \triangleq & \sum \left\{ a : q^{(A,n)} \mid a \in A, q \in \mathsf{derivatives}(p,a) \right\} \\ & & [+\Omega \mid p\uparrow \text{ or } \mathsf{initials}(p) \not\subseteq A] \ . \end{array}$$

Intuitively, the synchronization tree $p^{(A,n)}$ stands for the approximation of the behaviour of p of width A and height $n+1$. For example, if we apply the above definition to derive the approximations of the infinitely branching synchronization trees p and q given in (1) and (2), respectively, we obtain that, for every $A \subseteq_{fin} \mathsf{Act}$ and $n \geq 0$,

$$p^{(A,n+1)} = \sum_{\{i \mid a_i \in A\}} a_i : \mathbb{O}^{(A,n)} + \Omega = q^{(A,n+1)}$$

where $\mathbb{O}^{(A,n)}$ is Ω if $n = 0$, and $\mathbb{O}$ otherwise. Thus, albeit p is a convergent synchronization tree, all of its approximations are divergent, and coincide with the approximations of the behaviour of q.

By a simple induction, we may show that, for every *finite* set of actions A and non-negative integer n, the set of synchronization trees $\left\{ p^{(A,n)} \mid p \in \mathsf{Proc} \right\}$ is finite. Therefore the synchronization tree $p^{(A,n+1)}$ is finite even when the set $\mathsf{derivatives}(p,a)$ is infinite for some $a \in A$. The fact that the synchronization trees $p^{(A,n)}$ do behave as approximations of the behaviour of p of width A and depth n is the import of the following result, which may be easily shown by mathematical induction:

Lemma 3.4 *For every $A \subseteq_{fin} \mathsf{Act}$, $n \geq 0$,*

1. *$p \sqsubseteq_{n}^{A} p^{(A,n)}$, and*
2. *$p^{(A,n)} \lesssim p$.*

We are now in a position to prove statement Lemma 3.3 above, i.e., that the preorder $\sqsubseteq_{\omega}^{fin}$ is finitary.

Proof of Lemma 3.3: We prove that

$$p \sqsubseteq_{\omega}^{fin} q \quad \text{iff} \quad \forall t \in \mathsf{ST(Act)}.\ t \sqsubseteq_{\omega}^{fin} p \ \Rightarrow \ t \sqsubseteq_{\omega}^{fin} q \ .$$

The 'only if' implication follows immediately from the fact that $\sqsubseteq_{\omega}^{fin}$ is a preorder. To establish the 'if' implication, let us assume that p and q are two processes such that, for every $t \in \mathsf{ST(Act)}$,

$$t \sqsubseteq_{\omega}^{fin} p \ \Rightarrow \ t \sqsubseteq_{\omega}^{fin} q \ . \tag{4}$$

We show that $p \sqsubseteq_{n}^{A} q$ holds for every finite set of actions A and non-negative integer n. To this end, let $A \subseteq_{fin} \mathsf{Act}$ and $n \geq 0$. We know that $p \sqsubseteq_{n}^{A} p^{(A,n)} \lesssim p$ (Lemma 3.4). As

$p^{(A,n)}$ is a finite synchronization tree, it follows that $p \sqsubseteq_n^A p^{(A,n)} \sqsubseteq_\omega^{fin} p$ (Lemma 3.2). Using (4), we now obtain that $p \sqsubseteq_n^A p^{(A,n)} \sqsubseteq_\omega^{fin} q$ holds. By the definition of $\sqsubseteq_\omega^{fin}$ and the transitivity of $\sqsubseteq_n^A$ (Propn. 3.1(1)), we finally infer that $p \sqsubseteq_n^A q$ holds, which was to be shown. $\square$

Collecting the intermediate results presented so far, we can now establish the main result of this paper.

Theorem 3.5 *For $p, q \in \mathsf{Proc}$ in any transition system, $p \lesssim^F q$ iff $p \sqsubseteq_\omega^{fin} q$.*

Proof: The preorder $\sqsubseteq_\omega^{fin}$ is finitary (Lemma 3.3), and coincides with $\lesssim$—and thus with $\lesssim^F$—over $\mathsf{ST(Act)} \times \mathsf{Proc}$ (Lemma 3.2). It is immediate to see that two finitary relations that coincide over $\mathsf{ST(Act)} \times \mathsf{Proc}$ do coincide over the whole of Proc. $\square$

In [3, Propn. 6.13], Abramsky showed that for any sort-finite lts satisfying his axiom scheme of bounded nondeterminacy (BN) (cf. *op. cit.* on page 193), the finitary bisimulation coincides with the ω-iterate of the bisimulation preorder $\lesssim_\omega$. Following the proof of our previous characterization theorem, we can now present a sharpened version of this result, which does *not* require the lts's to satisfy the axiom (BN).

Proposition 3.6 *For $p, q \in \mathsf{Proc}$ in any sort-finite transition system, $p \lesssim^F q$ iff $p \lesssim_\omega q$.*

Proof: It is well-known that, in any lts, not necessarily sort-finite, $p \lesssim_\omega q$ implies $p \lesssim^F q$. We are therefore left to show that, for $p, q \in \mathsf{Proc}$ in any sort-finite transition system, $p \lesssim^F q$ implies $p \lesssim_\omega q$. This statement can be shown by mimicking the proof of Lemma 3.3 presented above. In fact, it is not too hard to show that, for every process p in a sort-finite lts, $n \geq 0$ and finite set of actions A including $\mathsf{sort}(p)$, $p \lesssim_n p^{(A,n)} \lesssim p$. In particular, we obtain that $p \lesssim_n p^{(\mathsf{sort}(p),n)} \lesssim p$. If $p \lesssim^F q$ and $\mathsf{sort}(p)$ is finite, it follows that $p \lesssim_n p^{(\mathsf{sort}(p),n)} \lesssim q$. Using the fact that $\lesssim$ is included in $\lesssim_\omega$, we may now infer that $p \lesssim_n q$ holds for every $n \geq 0$. We have therefore shown that, if $p \lesssim^F q$ and $\mathsf{sort}(p)$ is finite, then $p \lesssim_\omega q$. $\square$

Acknowledgements: We thank the anonymous referees for their constructive suggestions.

References

[1] S. Abramsky, *Domain Theory and the Logic of Observable Properties*, PhD thesis, University of London, 1987. Available on the World Wide Web at the URL `http://theory.doc.ic.ac.uk:80/people/Abramsky/` as file `phd.ps.gz`.

[2] ——, *Observation equivalence as a testing equivalence*, Theoretical Comput. Sci., 53 (1987), pp. 225–241.

[3] ——, *A domain equation for bisimulation*, Information and Computation, 92 (1991), pp. 161–218.

[4] L. Aceto and M. Hennessy, *Termination, deadlock and divergence*, J. Assoc. Comput. Mach., 39 (1992), pp. 147–187.

[5] L. Aceto and A. Ingólfsdóttir, *On the finitary bisimulation*, Research Report RS–95–59, BRICS (Basic Research in Computer Science, Centre of the Danish National Research Foundation), Department of Computer Science, Aalborg University, Nov. 1995. Available by anonymous ftp at the address `ftp.daimi.aau.dk` in the directory `pub/BRICS/RS/95/59`.

[6] M. Hennessy, *A term model for synchronous processes*, Information and Control, 51 (1981), pp. 58–75.

[7] M. Hennessy and R. Milner, *Algebraic laws for nondeterminism and concurrency*, J. Assoc. Comput. Mach., 32 (1985), pp. 137–161.

[8] M. Hennessy and G. Plotkin, *A term model for CCS*, in 9^{th} Symposium on Mathematical Foundations of Computer Science, P. Dembiński, ed., vol. 88 of Lecture Notes in Computer Science, Springer-Verlag, 1980, pp. 261–274.

[9] R. Keller, *Formal verification of parallel programs*, Comm. ACM, 19 (1976), pp. 371–384.

[10] R. Milner, *A Calculus of Communicating Systems*, vol. 92 of Lecture Notes in Computer Science, Springer-Verlag, 1980.

[11] ——, *A modal characterisation of observable machine behaviour*, in Proceedings CAAP 81, G. Astesiano and C. Bohm, eds., vol. 112 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 25–34.

[12] D. Park, *Concurrency and automata on infinite sequences*, in 5^{th} GI Conference, Karlsruhe, Germany, P. Deussen, ed., vol. 104 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 167–183.

[13] D. Walker, *Bisimulation and divergence*, Information and Computation, 85 (1990), pp. 202–241.

Recent BRICS Report Series Publications

RS-97-26 **Luca Aceto and Anna Ingólfsdóttir.** ***A Characterization of Finitary Bisimulation.*** **October 1997. 9 pp. To appear in *Information Processing Letters*.**

RS-97-25 **David A. Mix Barrington, Chi-Jen Lu, Peter Bro Miltersen, and Sven Skyum.** ***Searching Constant Width Mazes Captures the AC^0 Hierarchy.*** **September 1997. 20 pp.**

RS-97-24 **Søren B. Lassen.** ***Relational Reasoning about Contexts.*** **September 1997. 45 pp. To appear as a chapter in the book *Higher Order Operational Techniques in Semantics*, eds. Andrew D. Gordon and Andrew M. Pitts, Cambridge University Press.**

RS-97-23 **Ulrich Kohlenbach.** ***On the Arithmetical Content of Restricted Forms of Comprehension, Choice and General Uniform Boundedness.*** **August 1997. 35 pp.**

RS-97-22 **Carsten Butz.** ***Syntax and Semantics of the logic $\mathcal{L}^{\lambda}_{\omega\omega}$.*** **July 1997. 14 pp.**

RS-97-21 **Steve Awodey and Carsten Butz.** ***Topological Completeness for Higher-Order Logic.*** **July 1997. 19 pp.**

RS-97-20 **Carsten Butz and Peter T. Johnstone.** ***Classifying Toposes for First Order Theories.*** **July 1997. 34 pp.**

RS-97-19 **Andrew D. Gordon, Paul D. Hankin, and Søren B. Lassen.** ***Compilation and Equivalence of Imperative Objects.*** **July 1997. iv+64 pp. Appears also as Technical Report 429, University of Cambridge Computer Laboratory, June 1997. To appear in *Foundations of Software Technology and Theoretical Computer Science: 17th Conference*, FCT&TCS '97 Proceedings, LNCS, 1997.**

RS-97-18 **Robert Pollack.** ***How to Believe a Machine-Checked Proof.*** **July 1997. 18 pp. To appear as a chapter in the book *Twenty Five Years of Constructive Type Theory*, eds. Smith and Sambin, Oxford University Press.**

RS-97-17 **Peter Bro Miltersen.** ***Error Correcting Codes, Perfect Hashing Circuits, and Deterministic Dynamic Dictionaries.*** **June 1997. 10 pp.**