BRICS

Basic Research in Computer Science

BRICS RS-95-59 Aceto & Ingólfsdóttir: On the Finitary Bisimulation

On the Finitary Bisimulation

Luca Aceto
Anna Ingólfsdóttir

BRICS Report Series **RS-95-59**

ISSN 0909-0878 **November 1995**

See back inner page for a list of recent publications in the BRICS Report Series. 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@daimi.aau.dk

BRICS publications are in general accessible through WWW and anonymous FTP:

`http://www.brics.aau.dk/BRICS/`
`ftp ftp.brics.aau.dk (cd pub/BRICS)`

On the Finitary Bisimulation

Luca Aceto* Anna Ingólfsdóttir†

BRICS‡
Department of Mathematics and Computer Science
Aalborg University
Fredrik Bajersvej 7E
9220 Aalborg Ø, Denmark

Abstract

The finitely observable, or *finitary*, part of bisimulation is a key tool in establishing full abstraction results for denotational semantics for process algebras with respect to bisimulation-based preorders. A bisimulation-like characterization of this relation for arbitrary transition systems is given, relying on Abramsky's characterization in terms of the finitary domain logic. More informative behavioural, observation-independent characterizations of the finitary bisimulation are also offered for several interesting classes of transition systems. These include transition systems with countable action sets, those that have bounded convergent sort and the sort-finite ones. The result for sort-finite transition systems sharpens a previous behavioural characterization of the finitary bisimulation for this class of structures given by Abramsky.

AMS Subject Classification (1991): 68Q10 (Modes of computation), 68Q55 (Semantics), 03B70 (Logic of Programming), 68Q90 (Transition nets).
Keywords and Phrases: Concurrency, labelled transition systems with divergence, bisimulation preorder, finitary relations, domain logic for transition systems.

1 Introduction

Following a paradigm put forward by Milner and Plotkin in their seminal papers [20, 21, 26], 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 (cf., e.g., the reference [19] for a discussion), the most desirable one is that of *full abstraction*. Intuitively, a denotational model for a programming or specification language is fully abstract with respect to a notion of operationally based (or behavioural) equivalence or preorder iff it is in complete agreement with it. In other

*On leave from School of Cognitive and Computing Sciences, University of Sussex, Brighton BN1 9QH, UK. Partially supported by HCM project EXPRESS. Email: `luca@iesd.auc.dk`.
†Email: `annai@iesd.auc.dk`.
‡Basic Research in Computer Science, Centre of the Danish National Research Foundation.

words, 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 rôle in process theory, bisimulation [25, 24] has been a natural operational 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 [29] for CCS-like languages, several preorders based on bisimulation have been considered. (The interested reader is invited to consult, e.g., [13, 10, 4, 6, 17] for examples of full abstraction results with respect to bisimulation-based preorders for variations on CCS.) 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 investigated in, e.g., [10, 4]. This preorder will henceforth be denoted by $\lesssim$. Intuitively, $p \lesssim q$ holds of processes p and q iff 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 [27] or on the theory of algebraic semantics [9], with behavioural semantics based on bisimulation is that the chosen behavioural theory is, in general, too concrete. The reason for this phenomenon 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 [4] for a detailed analysis of this phenomenon.) As an example, consider the infinite synchronization trees

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

where Ω stands for the synchronization tree whose behaviour is completely unspecified, and $\mathbb{O}$ stands for the one-node synchronization tree. Then, in a precise sense, no finite amount of observation can distinguish the behaviour of p from that of q. On the other hand, $q \lesssim p$ does *not* hold because p cannot simulate the infinite a-computation possessed by q.

To overcome this mismatch between the denotational and the behavioural theory, all the aforementioned full abstraction results are couched in terms of the so-called *finitely observable*, or *finitary*, part of bisimulation. (This relation will be henceforth referred to as the finitary bisimulation.) The *finitary bisimulation preorder*, denoted by $\lesssim^F$, is defined on any labelled transition system thus:

$$p \lesssim^F q \text{ iff for every finite synchronization tree } t, \quad t \lesssim p \ \text{ implies } \ t \lesssim q \ .$$

The above definition of the finitary bisimulation, albeit very natural and intuitive, is rather indirect; it simply says that two processes are related iff they afford the same finite observations, which are taken to be finite, possibly partially specified, synchronization trees. Such an indirect definition often makes it quite hard to establish results about this relation, and a lot of research effort has been devoted to finding alternative, observation-independent versions of the finitary bisimulation for different process description languages. (Cf., e.g., [10, 4, 6, 16] for examples of these results.)

A general, observation-independent characterization of the finitary bisimulation for arbitrary transition systems has been given by Abramsky in [4]. This characterization is couched in logical terms, and is an impressive byproduct of Abramsky's "theory of domains in logical form" programme [5]. More precisely, Abramsky shows that the domain logic for transition systems synthesized in [2] characterizes the finitary bisimulation for all transition systems, i.e., that two processes in any transition system are related by the finitary bisimulation iff they satisfy the same formulae in the finitary version of the domain logic for transition systems. In many ways, Abramsky's logical characterization of the finitary bisimulation can be seen as the counterpart of the modal characterizations for bisimulation-like relations presented in, e.g., [22, 12, 30].

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. The existence of observation-independent, behavioural characterizations 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 particular, as transition systems abstract from the operational semantics of many process description languages, we believe that it would be worthwhile to establish general bisimulation-like characterizations of the finitary bisimulation for interesting classes of such structures. The availability of this type of results would imply, for instance, that, when establishing full abstraction results for a particular process description language, it would be sufficient to identify the kind of transition system giving an operational semantics to the chosen language, and check what form the finitary bisimulation for that type of transition system takes. One could then proceed to compare the appropriate explicit characterization of this behavioural preorder with the denotational ordering on processes.

1.1 Results

This study presents a collection of observation independent, bisimulation-like characterizations of the finitary bisimulation for several classes of labelled transition systems, including those that commonly arise in giving operational semantics to process description languages. First of all, we present a behavioural characterization of the finitary bisimulation for arbitrary transition systems (cf. Thm. 3.2). This result may be seen as the behavioural counterpart of Abramsky's logical characterization theorem [4, Thm. 5.5.8]. We offer two independent proofs of this characterization theorem. The first relies on Abramsky's logical characterization result in terms of the domain logic for transition systems; the second is purely operational, and is based on a generalization of a beautiful argument due to Hennessy [10]. This first behavioural characterization applies, e.g., to transition systems with an uncountable action set like those that arise in timed process calculi which postulate the positive real numbers as their time domain. (Cf., e.g., Wang's TCCS [33].)

We then concentrate our attention on transition systems over a countable action set. For several classes of such transition systems (viz. the class of all such transition systems, the class of those which have bounded convergent sort, the sort-finite transition systems and those that are image finite [12] and weakly finite branching), we provide customized, and more informative, versions of the general behavioural characterization offered by Thm. 3.2. In particular, for the important class of sort-finite transition systems we are able to present a sharpened version of a behavioural characterization result first proven by Abramsky in [4, Propn. 6.13].

We hope that this taxonomy of characterizations of the finitary bisimulation will be useful for researchers interested in full abstraction results for process description languages.

1.2 Outline of the Paper

We conclude this introduction with a brief road-map to the contents of this study. We begin by presenting the basic notions from process theory and Abramsky's domain logic for transition systems in Sect. 2. Section 3 is the côre of the paper, and is entirely devoted to a taxonomy of behavioural characterizations of the finitary bisimulation for various classes of transition systems. The characterization of $\lesssim^F$ for arbitrary transition systems is the subject of Sect. 3.1. Section 3.2 is devoted to behavioural characterizations of the finitary bisimulation for transition systems over a countable action set. Apart from the class of all such transition systems, we also deal with transition systems that have bounded convergent sort, sort-finite transition systems and the image finite ones that are also weakly finite branching.

2 Preliminaries

In this section we present the basic notions from process theory and Abramsky's domain logic for transition systems that will be needed in the remainder of this study.

2.1 Labelled Transition Systems and Prebisimulation

We begin by reviewing a variation on the model of labelled transition systems [18, 28] that takes divergence information into account, and abstracts from the operational semantics of many concurrent calculi. We refer the interested readers to, e.g., [13, 10, 22, 32] for motivation and more information on (variations on) this semantic model for reactive systems.

Definition 2.1 (Labelled Transition Systems with Divergence) *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 definitely converges", iff it is not the case that $p \uparrow$. Intuitively, the fact that a process p definitely converges means that the initial behaviour of p 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 $p \stackrel{\sigma}{\rightarrow} q$ iff there exist processes $p_0, \ldots, p_n$ such that

$$p = p_0 \stackrel{a_1}{\rightarrow} p_1 \stackrel{a_2}{\rightarrow} \cdots p_{n-1} \stackrel{a_n}{\rightarrow} p_n = q \ .$$

For a process $p \in \mathsf{Proc}$ and action $a \in \mathsf{Act}$ we define:

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

Following [4], we say that an lts is

image finite	iff	$\mathsf{derivatives}(p, a)$ is finite for every $p \in \mathsf{Proc}$ and $a \in \mathsf{Act}$
sort-finite	iff	$\mathsf{sort}(p)$ is finite for every $p \in \mathsf{Proc}$
finite branching	iff	$\left\{(a, q) \mid p \stackrel{a}{\rightarrow} q\right\}$ is finite for every $p \in \mathsf{Proc}$
weakly finite branching	iff	for every $p \in \mathsf{Proc}$, if $p \downarrow$ then $\left\{(a, q) \mid p \stackrel{a}{\rightarrow} q\right\}$ is finite
weakly initials finite	iff	for every $p \in \mathsf{Proc}$, if $p \downarrow$ then $\mathsf{initials}(p)$ is finite.

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
- $t \xrightarrow{a_i} t_i$ 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* [13, 22, 10, 32] (also known as *partial bisimulation* [4]).

Definition 2.2 (Prebisimulation) *Let* $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ *be an lts. Let* $\mathsf{Rel}(\mathsf{Proc})$ *denote the set of binary relations over* Proc. *Define the functional* $F : \mathsf{Rel}(\mathsf{Proc}) \rightarrow \mathsf{Rel}(\mathsf{Proc})$ *by:*

$$\begin{aligned} F(\mathcal{R}) \;=\; & \{(p,q) \mid \forall a \in \mathsf{Act} \\ & \bullet\ p \xrightarrow{a} p' \Rightarrow \exists q' :\ q \xrightarrow{a} q' \textit{ and } p' \mathrel{\mathcal{R}} q' \\ & \bullet\ p \downarrow \Rightarrow q \downarrow \ \textit{ and } [q \xrightarrow{a} q' \Rightarrow \exists p' :\ p \xrightarrow{a} p' \textit{ and } p' \mathrel{\mathcal{R}} q']\} \end{aligned}$$

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* $p \mathrel{\mathcal{R}} q$.

The relation $\lesssim$ is a preorder over Proc based on a variation on bisimulation equivalence [25, 24]. Intuitively, $p \lesssim q$ if q's behaviour is at least as specified as that of p, and p and q can simulate each other when restricted to the part of their behaviour that is fully specified. A divergent process p that, like the synchronization tree Ω, has no outgoing transition is a minimal element with respect to $\lesssim$, and intuitively corresponds to a process whose behaviour is totally unspecified — essentially an operational version of the bottom element $\perp$ in Scott's theory of domains (cf., e.g., the references [29, 27, 31] for information on domain theory).

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

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

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 $q \stackrel{a}{\to} a^\omega$ cannot be matched by any a-transition emanating from p. On the other hand, it is easy to see that, for every $i \geq 0$,

$$a^\omega \lesssim_i \underbrace{a : \cdots : a :}_{i\text{-times}} \mathbb{0} \ .$$

Therefore $q \lesssim_\omega p$ does instead hold.

Remark: Although the relations $\lesssim$ and $\lesssim_\omega$ have been defined over a given lts, we often want to use them to compare processes from different lts's; for example, we shall often compare processes in an lts with finite synchronization trees. This can be done in standard fashion by forming the disjoint union of the two transition systems, and then using $\lesssim$ or $\lesssim_\omega$ on the resulting lts. In the sequel, this will be done without further comment.

In this paper, we shall be interested in studying the "finitely observable", or *finitary*, part of the bisimulation in the sense of, e.g., [9, 10]. The following definition is from [4]. (The interested reader is invited to consult the aforementioned references for discussion and motivations.)

Definition 2.3 *Let* $(\mathsf{Proc}, \mathsf{Act}, \to, \uparrow)$ *be an lts. The* finitary bisimulation preorder $\lesssim^F$ *over* Proc *is defined thus:*

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

The preorders $\lesssim$, $\lesssim_\omega$ and $\lesssim^F$ are related thus:

$$\lesssim\ \subseteq\ \lesssim_\omega\ \subseteq\ \lesssim^F \ .$$

Moreover the inclusions are, in general, strict. The interested reader is referred to [4] for a wealth of examples distinguishing these preorders, and a very deep analysis of their general relationships and properties. Here we just discuss an instructive example from [4, Page 191] showing that the preorder $\lesssim_\omega$ (and *a fortiori* $\lesssim$) makes distinctions based on infinite observations.

Example: Consider the synchronization trees p and q given by:

$$p \triangleq a : (\textstyle\sum_{n \in \omega} b_n : \mathbb{0} + \Omega) + \Omega \qquad (1)$$

$$q \triangleq \sum_{n \in \omega} a : \left(\textstyle\sum_{m \in \omega - \{n\}} b_m : \mathbb{0} + \Omega\right) + \Omega \ . \qquad (2)$$

It is easy to see that $p \not\lesssim_2 q$. We shall now argue that $p \lesssim^F q$ does instead hold. To this end, let t be a finite synchronization tree such that $t \lesssim p$. Because of the definition of the bisimulation preorder, this implies that the following conditions are met:

1. $t \uparrow$;

2. $\mathsf{initials}(t) = \{a\}$;

3. for every t' such that $t \xrightarrow{a} t'$, $t' \lesssim \sum_{n\in\omega} b_n : \mathbb{O} + \Omega$.

It is not too difficult to see that for every finite synchronization tree u,

$$u \lesssim \sum_{n\in\omega} b_n : \mathbb{O} + \Omega \text{ implies } u \lesssim \sum_{m\in\omega-\{n\}} b_m : \mathbb{O} + \Omega, \quad \text{for some } n \in \omega.$$

This follows because any finite synchronization tree u such that $u \lesssim \sum_{n\in\omega} b_n : \mathbb{O} + \Omega$ is divergent, and its set of initial actions must be a *finite* subset of $\{b_n \mid n \in \omega\}$.

Collecting the above observations it is immediate to see that $t \lesssim q$ also holds. $\square$

To our mind, the above example demonstrates in an explicit and instructive way the mismatch between the finitary part of the bisimulation preorder and the preorder $\lesssim_\omega$. Intuitively, the preorder $\lesssim_\omega$ is based on observations of finite depth. However, in the presence of (weakly) infinite branching processes like p and q above, this preorder can make distinctions based on observations of *infinite width*. As outlined in the above example, no observation of finite depth and width can differentiate the processes defined in (1) and (2). (The interested reader may wish to consult the reference [1] for illuminating discussions on the issue of finite, and infinite, depth and width in experiments in the setting of applicative, non-deterministic programs.)

Abramsky's Domain Logic for Transition Systems

We now review the basic definitions and results on Abramsky's domain logic for transition systems that will be useful in this study. We shall follow the presentation of the logic given in [2]; the interested reader is referred to that reference for a detailed explanation of the logic, its origins, semantics and proof theory.

Abramsky's (finitary) domain logic for transition systems $\mathcal{L}_\omega$ (over a set of actions Act) is a two-sorted language with sorts π (process) and κ (capability). We write $\mathcal{L}_{\omega\pi}$ (respectively, $\mathcal{L}_{\omega\kappa}$) for the class of formulae of sort π (respectively, κ), which are defined inductively as follows:

$$\frac{\{\varphi_i \in \mathcal{L}_{\omega\sigma}\}_{i\in I}}{\bigwedge_{i\in I}\varphi_i, \bigvee_{i\in I}\varphi_i \in \mathcal{L}_{\omega\sigma}} \qquad \frac{a \in \mathsf{Act},\ \varphi \in \mathcal{L}_{\omega\pi}}{a(\varphi) \in \mathcal{L}_{\omega\kappa}} \qquad \frac{\varphi \in \mathcal{L}_{\omega\kappa}}{\Box\varphi, \Diamond\varphi \in \mathcal{L}_{\omega\pi}}$$

where I is a finite index set, and $\sigma \in \{\pi, \kappa\}$. As usual, we write $\mathsf{true} \triangleq \bigwedge_{i\in\varnothing}\varphi_i$ and $\mathsf{false} \triangleq \bigvee_{i\in\varnothing}\varphi_i$.

The *modal depth* of a formula φ, notation $\mathsf{md}(\varphi)$, is the non-negative integer giving the maximum nesting of occurrences of the modal operators $\Box$ and $\Diamond$ in it. Its sort, notation $\mathsf{sort}(\varphi)$, is the finite set of actions mentioned in φ.

For each $A \subseteq \mathsf{Act}$ and non-negative integer n:

$$\mathcal{L}_\omega^{(A,n)} \triangleq \{\varphi \in \mathcal{L}_\omega \mid \mathsf{md}(\varphi) \leq n \text{ and } \mathsf{sort}(\varphi) \subseteq A\} \ .$$

Given an lts $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$, we define:

$$\begin{aligned} \mathrm{Cap} &\triangleq \{\bot\} \cup (\mathsf{Act} \times \mathsf{Proc}) \quad \text{(the set of \textit{capabilities})} \\ C(p) &\triangleq \{\bot \mid p\uparrow\} \cup \left\{\langle a, q\rangle \mid p \xrightarrow{a} q\right\} \quad \text{(the set of capabilities of process } p) \ . \end{aligned}$$

The satisfaction relations

$$\begin{array}{rcl} \models_\pi & \subseteq & \mathsf{Proc} \times \mathcal{L}_{\omega\pi} \quad \text{and} \\ \models_\kappa & \subseteq & \mathrm{Cap} \times \mathcal{L}_{\omega\kappa} \end{array}$$

are now defined thus ($\sigma \in \{\pi, \kappa\}$, $w \in \mathsf{Proc} \cup \mathrm{Cap}$):

$$\begin{array}{rcl} w \models_\sigma \bigwedge_{i \in I} \varphi_i & \triangleq & \forall i \in I.\ w \models_\sigma \varphi_i \\ w \models_\sigma \bigvee_{i \in I} \varphi_i & \triangleq & \exists i \in I:\ w \models_\sigma \varphi_i \\ p \models_\pi \Box\varphi & \triangleq & \forall c \in C(p).\ c \models_\kappa \varphi \\ p \models_\pi \Diamond\varphi & \triangleq & \exists c \in C(p) \cup \{\perp\}.\ c \models_\kappa \varphi \\ c \models_\kappa a(\varphi) & \triangleq & c = \langle a, q \rangle \text{ and } q \models_\pi \varphi\ . \end{array}$$

For $\mathcal{F} \subseteq \mathcal{L}_\omega$ and process p, we write $\mathcal{F}(p)$ for the set of (process) formulae in the set $\mathcal{F}$ satisfied by p. In what follows, we shall always omit sort information from (sets of) formulae, and satisfaction relations.

As shown by Abramsky in his thesis [2], the logic $\mathcal{L}_\omega$ is a powerful tool in the study of the finitary bisimulation preorder $\lesssim^F$. In particular, Abramsky shows the following key characterization theorem (see [2, Thm. 5.5.8]):

Theorem 2.4 (Characterization Theorem for $\mathcal{L}_\omega$) *For $p, q \in \mathsf{Proc}$ in any transition system,*

$$p \lesssim^F q \iff \mathcal{L}_\omega(p) \subseteq \mathcal{L}_\omega(q)\ .$$

The above seminal result will be a major tool in this study. In what follows, we shall also have some use for the following observation, which is due to Abramsky (cf., e.g., the proof of Propn. 5.5.12 in [2]).

Fact 2.5 *Let $A \subseteq \mathsf{Act}$ be a* finite *set of actions. Then, for every non-negative integer n, $\mathcal{L}_\omega^{(A,n)}$ is finite up to logical equivalence.*

3 Behavioural Characterizations of the Finitary Bisimulation

The definition of the finitely observable part of the bisimulation preorder given in Def. 2.3 is rather indirect. Rather like the original definition of De Nicola and Hennessy's testing equivalence [11], it just says that, in order for $p \lesssim^F q$ to hold, every observation that is possible of p should also be possible of q, where we identify the set of observations with that of finite synchronization trees. Definitions of behavioural preorders like the one in Def. 2.3 are, albeit very natural and intuitively appealing, quite difficult to work with. For this reason, it is useful to have alternative, observation-independent characterizations for them. Again using the analogy with the testing equivalences, the alternative characterizations of these relations provided in [11] have proven to be indispensable tools in the development of their theory and practice.

Abramsky's logical characterization of the finitary bisimulation (cf. Thm. 2.4) 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 [22, 12]. 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., [4, Propn. 6.13]. There Abramsky shows that in any sort-finite lts that satisfies his axiom scheme of bounded nondeterminacy (BN) (cf. [2, Page 114]), the finitary bisimulation coincides with $\lesssim_\omega$. In this study, building on Abramsky's work, we shall present several bisimulation-like characterizations of the finitary bisimulation for various classes of 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. 3.11.)

3.1 A General Behavioural Characterization

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:

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

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 $p, q \in \mathsf{Proc}$ in any transition system, $p \sqsubseteq_{n+1}^A q$ implies $p \sqsubseteq_n^A q$.*
3. *Assume that $A \subseteq B \subseteq \mathsf{Act}$. Then, for $p, q \in \mathsf{Proc}$ in any transition system, $p \sqsubseteq_n^B q$ implies $p \sqsubseteq_n^A q$*

Proof: The proofs of all the statements are routine by mathematical induction on n. Here we only remark that, in the proof of the inductive step for statement 3, the following simple observation is used:

$$\text{if } p \sqsubseteq_{n+1}^B q,\ p \downarrow,\ q \downarrow \text{ and } \mathsf{initials}(p) \subseteq A \subseteq B, \text{ then } \mathsf{initials}(q) \subseteq A.$$

□

We now define:

$$
\begin{aligned}
p \underset{\sim}{\sqsubset}_{\omega}^{A} q &\triangleq \forall n \geq 0.\ p \underset{\sim}{\sqsubset}_{n}^{A} q \\
p \underset{\sim}{\sqsubset}_{\omega}^{fin} q &\triangleq \forall A \subseteq_{fin} \mathsf{Act}.\ p \underset{\sim}{\sqsubset}_{\omega}^{A} q
\end{aligned}
$$

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 $\underset{\sim}{\sqsubset}_{\omega}^{\mathsf{Act}}$ coincides with the preorder $\lesssim_\omega$, defined on page 6.

Intuitively, $p \underset{\sim}{\sqsubset}_{\omega}^{A} q$ holds for two processes p and q iff there is no observation, in the sense of [3], of finite depth, and with actions drawn from the set A, that can distinguish between p and q. For example, $p \underset{\sim}{\sqsubset}_{\omega}^{\emptyset} q$ 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 $\underset{\sim}{\sqsubset}_{\omega}^{fin}$, 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 example on page 7. A possibly even more striking example of the rôle played by finite width in observations, and of the weakness of $\lesssim^F$ over infinite branching processes, is the following:

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:

$$p \triangleq \sum_{i \geq 0} a_i : \mathbb{0} \tag{3}$$

$$q \triangleq p + \Omega\ . \tag{4}$$

Then $p \not\lesssim_1 q$ because $p\downarrow$ but $q\uparrow$. However, $p \underset{\sim}{\sqsubset}_{\omega}^{fin} q$. 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 $\underset{\sim}{\sqsubset}_{n+1}^{A}$ 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 $t' \lesssim \mathbb{0}$. From these two observations, it follows immediately that $t \lesssim q$.

The moral of the above example is that, for lts's that are not weakly initials finite, observations based on finite synchronization trees cannot, in general, be used to test the convergence of a process that, like p above, can perform infinitely many distinct initial actions. As shown by the technical results to follow, this is the reason for the presence of the non-standard test on the set of initial actions of a process in clause (2) of the definition of the preorder $\underset{\sim}{\sqsubset}_{n+1}^{A}$. □

Remark: The reader familiar with Apt and Plotkin's arguments for the failure of continuous semantics for random assignment presented in [7, pp. 741–747] might have noticed that the example we have just discussed is closely related to the one used *ibid.* to demonstrate the mismatch between operational and continuous denotational semantics for countable nondeterminism.

Apt and Plotkin show that in any 'reasonable' continuous semantics the programs

$$P \;\triangleq\; \mathsf{x{:=}?;while\ x>0\ do\ x{:=}x{-}1\ od}$$

and

$$P\ \mathsf{or}\ (\mathsf{while\ true\ do\ skip\ od})$$

where x:=? denotes the random assignment of a non-negative integer to the variable x, and or stands for nondeterministic choice, are necessarily identified, even though the former is guaranteed to terminate in a state where x has value 0, whereas the latter does not. The point is that, due to the countable nondeterminism in the computation tree for program P, no finite observation can detect the possible infinite loop in the second program.

Our aim is to prove:

Theorem 3.2 *For $p, q \in \mathsf{Proc}$ in any transition system,*

$$p \lesssim^F q \;\Leftrightarrow\; p \sqsubseteq_\omega^{fin} q \ .$$

Our proof of this general behavioural characterization theorem for the finitary bisimulation relies on the logical formulation of this relation given by Thm. 2.4. The key step in the proof is presented in the following result, that gives a logical characterization of the preorders $\sqsubseteq_n^A$ when the set of actions A is finite.

Proposition 3.3 *Let $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ be an lts. For every $A \subseteq_{fin} \mathsf{Act}$, $n \geq 0$,*

$$p \sqsubseteq_n^A q \;\Leftrightarrow\; \mathcal{L}_\omega^{(A,n)}(p) \subseteq \mathcal{L}_\omega^{(A,n)}(q) \ .$$

Proof: We prove the two implications separately.

- 'If Implication'. We prove the contrapositive statement; namely, we show that, for all $p, q \in \mathsf{Proc}$, $A \subseteq_{fin} \mathsf{Act}$, $n \geq 0$,

  $$p \not\sqsubseteq_n^A q \;\Rightarrow\; \exists \varphi \in \mathcal{L}_\omega^{(A,n)}(p) \setminus \mathcal{L}_\omega^{(A,n)}(q) \ .$$

  The proof is by mathematical induction on n, and the basis of the inductive argument is vacuously true.

  For the inductive step, assume that $p \not\sqsubseteq_{n+1}^A q$. By the definition of $\sqsubseteq_{n+1}^A$, one of the following cases must arise:

  1. there exist $a \in A$ and $p' \in \mathsf{Proc}$ such that $p \xrightarrow{a} p'$, and, for every $q' \in \mathsf{derivatives}(q, a)$, $p' \not\sqsubseteq_n^A q'$; or
  2. $\mathsf{initials}(p) \subseteq A$, $p \downarrow$ and
     (a) $q \uparrow$ or
     (b) $\mathsf{initials}(q) \not\subseteq A$; or
  3. $\mathsf{initials}(p) \subseteq A$, $\mathsf{initials}(q) \subseteq A$, $p \downarrow$, $q \downarrow$ and there exist $a \in A$ and $q' \in \mathsf{Proc}$ such that $q \xrightarrow{a} q'$, and, for every $p' \in \mathsf{derivatives}(p, a)$, $p' \not\sqsubseteq_n^A q'$.

  In each case, we shall show how to construct a formula $\varphi \in \mathcal{L}_\omega^{(A,n+1)}(p) \setminus \mathcal{L}_\omega^{(A,n+1)}(q)$.

1. Assume that there exist $a \in A$ and $p' \in \mathsf{Proc}$ such that $p \xrightarrow{a} p'$, and, for every $q' \in \mathsf{derivatives}(q,a)$, $p' \not\lesssim_n^A q'$. The inductive hypothesis now gives that, for every $q' \in \mathsf{derivatives}(q,a)$, there exists a formula $\varphi_{q'} \in \mathcal{L}_\omega^{(A,n)}(p') \setminus \mathcal{L}_\omega^{(A,n)}(q')$. Consider the set of formulae $\{\varphi_{q'} \mid q' \in \mathsf{derivatives}(q,a)\} \subseteq \mathcal{L}_\omega^{(A,n)}$. As A is a finite set of actions, Fact 2.5 gives that, up to logical equivalence, there are only finitely many distinct formulae in $\{\varphi_{q'} \mid q' \in \mathsf{derivatives}(q,a)\}$, say $\varphi_1, \ldots, \varphi_k$. We now define

$$\varphi \quad \triangleq \quad \Diamond a(\bigwedge_{i=1}^{k} \varphi_i) \ .$$

By construction, φ is a formula in $\mathcal{L}_\omega^{(A,n+1)}$. It is now a simple matter to show that p satisfies φ, while q does not.

2. Assume that $\mathsf{initials}(p) \subseteq A$, $p \downarrow$ and either $q \uparrow$ or $\mathsf{initials}(q) \not\subseteq A$. As A is a finite set, so is $\mathsf{initials}(p)$. We now define:

$$\varphi \quad \triangleq \quad \Box \bigvee \{a(\mathsf{true}) \mid a \in \mathsf{initials}(p)\} \ .$$

By construction, φ is a formula in $\mathcal{L}_\omega^{(A,n+1)}$. It is a simple matter to show that, as $p \downarrow$, p satisfies φ. On the other contrary, q does *not* satisfy φ. In fact, if $q \uparrow$, then $\perp \in C(q)$ and $\perp \not\models \bigvee \{a(\mathsf{true}) \mid a \in \mathsf{initials}(p)\}$. Otherwise, there exists an action $b \in \mathsf{initials}(q) \backslash A$. Thus, for some q', $\langle b, q' \rangle$ is a capability of q. As $\mathsf{initials}(p)$ is a subset of A, the capability $\langle b, q' \rangle$ does not satisfy the capability formula $\bigvee \{a(\mathsf{true}) \mid a \in \mathsf{initials}(p)\}$.

3. Assume that $\mathsf{initials}(p) \subseteq A$, $\mathsf{initials}(q) \subseteq A$, $p \downarrow$, $q \downarrow$ and there exist $a \in A$ and $q' \in \mathsf{Proc}$ such that $q \xrightarrow{a} q'$, and, for every $p' \in \mathsf{derivatives}(p,a)$, $p' \not\lesssim_n^A q'$.
By the inductive hypothesis, for every $p' \in \mathsf{derivatives}(p,a)$, there exists a formula $\varphi_{p'} \in \mathcal{L}_\omega^{(A,n)}(p') \setminus \mathcal{L}_\omega^{(A,n)}(q')$. Consider the collection of formulae

$$\{\varphi_{p'} \mid p' \in \mathsf{derivatives}(p,a)\} \subseteq \mathcal{L}_\omega^{(A,n)} \ .$$

As A is a finite set of actions, Fact 2.5 gives that, up to logical equivalence, there are only finitely many distinct formulae in $\{\varphi_{p'} \mid p' \in \mathsf{derivatives}(p,a)\}$, say $\varphi_1, \ldots, \varphi_k$. We now define

$$\varphi \quad \triangleq \quad \Box\Big(a(\textstyle\bigvee_{i=1}^{k} \varphi_i) \vee \bigvee \{b(\mathsf{true}) \mid b \in \mathsf{initials}(p) - \{a\}\}\Big) \ .$$

Note that, as $\mathsf{initials}(p) \subseteq A \subseteq_{fin} \mathsf{Act}$, φ is a formula in $\mathcal{L}_\omega^{(A,n+1)}$. It is now a simple matter to show that p satisfies φ, while q does not.

This completes the inductive argument and the proof of the 'if' implication.

- 'Only If Implication'. We prove that, for all $n \geq 0$, $p, q \in \mathsf{Proc}$,

$$p \lesssim_n^A q \quad \Rightarrow \quad \forall \varphi \in \mathcal{L}_\omega^{(A,n)}. \ (p \models \varphi \text{ implies } q \models \varphi) \ .$$

This we proceed to show by mathematical induction on n. The base case is trivially seen to hold because every formula in $\mathcal{L}_\omega^{(A,0)}$ is logically equivalent to either true or false.

For the inductive step, assume that $p \lesssim_{n+1}^A q$ and that $p \models \varphi \in \mathcal{L}_\omega^{(A,n+1)}$. We show that $q \models \varphi$ by a further structural induction on φ. Indeed, in light of [2, Lem. 5.5.2], it is sufficient to prove the claim for a specific class of formulae, viz. the normal forms defined in [2, Def. 5.5.1]. These are the formulae given by the following inductive definition:

1. If I is a finite index set and, for every $i \in I$, φ_i is a normal form, then $\bigwedge_{i\in I} \varphi_i$ and $\bigvee_{i\in I} \varphi_i$ are normal forms;
2. if $a \in \mathsf{Act}$ and φ is a normal form, then $\Diamond a(\varphi)$ is a normal form; and
3. If I is a finite index set, $\{a_i \mid i \in I\} \subseteq \mathsf{Act}$ and, for every $i \in I$, φ_i is a normal form, then $\Box \bigvee_{i\in I} a_i(\varphi_i)$ is a normal form. (Abramsky's definition of normal forms requires that the actions a_i be pairwise distinct. In the following, we shall not need this restriction, and we have decided to omit it for the sake of simplicity.)

We only consider the two interesting cases.

– Assume that $p \models \Diamond a(\varphi) \in \mathcal{L}_\omega^{(A,n+1)}$. By the definition of the satisfaction relation and the fact that $\bot \not\models a(\varphi)$, this is because there exists $c \in C(p)$ such that $c \models a(\varphi)$. This capability must be of the form $\langle a, p'\rangle$ for some $p' \models \varphi$, i.e., $p \xrightarrow{a} p' \models \varphi$ for some p'. As $a \in A$ and $p \sqsubseteq^A_{n+1} q$, it follows that $q \xrightarrow{a} q'$ and $p' \sqsubseteq^A_n q'$ for some q'. As $\varphi \in \mathcal{L}_\omega^{(A,n)}$, we may now apply the inductive hypothesis to infer that $q' \models \varphi$, from which $q \models \Diamond a(\varphi)$ follows immediately.

– Assume that $p \models \Box \bigvee_{i\in I} a_i(\varphi_i) \in \mathcal{L}_\omega^{(A,n+1)}$. Then, for every $c \in C(p)$,

$$c \models \bigvee_{i\in I} a_i(\varphi_i) \ .$$

Note, first of all, that this implies that $p \downarrow$, because otherwise $\bot \in C(p)$ and $\bot \not\models \bigvee_{i\in I} a_i(\varphi_i)$.

Assume now that $p \xrightarrow{a} p'$, for some action $a \in \mathsf{Act}$ and process p'. As $\langle a, p'\rangle \in C(p)$, it follows that, for some index $i_{a,p'} \in I$, $a = a_{i_{a,p'}}$ and $p' \models \varphi_{i_{a,p'}}$. In particular, this implies that $\mathsf{initials}(p) \subseteq \{a_i \mid i \in I\} \subseteq A$. Therefore, as $p \sqsubseteq^A_{n+1} q$, $p \downarrow$ and $\mathsf{initials}(p) \subseteq A$, we have that:

  * $q \downarrow$,
  * $\mathsf{initials}(q) \subseteq A$, and
  * for every $a \in A$, $q' \in \mathsf{Proc}$, whenever $q \xrightarrow{a} q'$, then $p \xrightarrow{a} p'$ for some p' such that $p' \sqsubseteq^A_n q'$.

As $q \downarrow$, it follows that $\bot \notin C(q)$. Let now $\langle a, q'\rangle \in C(q)$. Then $a \in A$, $q \xrightarrow{a} q'$ and $p \xrightarrow{a} p' \sqsubseteq^A_n q'$ for some p'. As $a = a_{i_{a,p'}}$ and $p' \models \varphi_{i_{a,p'}}$, an application of the inductive hypothesis gives that $\langle a, q'\rangle \models a_{i_{a,p'}}(\varphi_{i_{a,p'}})$. From this observation, $q \models \Box \bigvee_{i\in I} a_i(\varphi_i)$ follows immediately.

This completes the proof of the 'only if' implication.

The proof of the theorem is now complete. □

Remark: As witnessed by the proof of the 'if implication' in the statement of the above result, for each *finite* set of actions A we can construct a formula φ_A with the property that, for every $p \in \mathsf{Proc}$:

$$p \models \varphi_A \quad \Leftrightarrow \quad p \downarrow \text{ and } \mathsf{initials}(p) \subseteq A \ .$$

Such a formula is simply defined thus:

$$\varphi_A \quad \triangleq \quad \Box \bigvee \{a(\mathsf{true}) \mid a \in A\} \ .$$

Note, however, that, if Act is countably infinite, there is no formula $\varphi_\downarrow$ in $\mathcal{L}_\omega$ which is satisfied exactly by all the convergent processes. In fact, using such a formula we should be able to distinguish the behaviour of the synchronization trees p and q on page 11, which are related by the finitary bisimulation preorder. (The obvious candidate for such a formula $\varphi_\downarrow$ is $\Box \bigvee \{a(\mathsf{true}) \mid a \in \mathsf{Act}\}$, which uses infinite disjunction if Act is countably infinite.) In other words, using the finitary domain logic we can only test convergence for processes that have a finite set of initial actions. This is in agreement with the operational considerations prompted by our discussion of the example on page 11.

By contrast with the above considerations, such a formula $\varphi_\downarrow$ *can* be constructed in the version of the domain logic for transition systems presented by Abramsky in [4]. The satisfaction relation for the $\Box$ and $\Diamond$ modalities is *ibid.* given thus:

$$
\begin{aligned}
p \models_\pi \Box\varphi \quad &\triangleq \quad p\downarrow \ \text{ and } \ \forall c \in C(p).\ \ c \models_\kappa \varphi \\
p \models_\pi \Diamond\varphi \quad &\triangleq \quad \exists c \in C(p):\ \ c \models_\kappa \varphi
\end{aligned}
$$

With this interpretation, the formula $\Box\mathsf{true}$ is satisfied exactly by all the convergent processes. In light of the example on page 11, the presence of such a formula invalidates Thm. 5.8 on page 191 of [4], that states the logical characterization theorem for the finitary bisimulation in terms of the version of the domain logic offered *ibidem*.

Using Propn. 3.3, we can now easily prove the promised general behavioural characterization theorem for $\lesssim^F$.

Proof of Theorem 3.2: Let $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ be any lts. Let $p, q \in \mathsf{Proc}$. Then:

$$
\begin{aligned}
p \lesssim^F q \quad &\Leftrightarrow \quad \mathcal{L}_\omega(p) \subseteq \mathcal{L}_\omega(q) && \text{(Thm. 2.4)} \\
&\Leftrightarrow \quad \forall A \subseteq_{fin} \mathsf{Act}, n \geq 0.\ \ \mathcal{L}_\omega^{(A,n)}(p) \subseteq \mathcal{L}_\omega^{(A,n)}(q) && \\
&\Leftrightarrow \quad \forall A \subseteq_{fin} \mathsf{Act}, n \geq 0.\ \ p \sqsubseteq_n^A q && \text{(Propn. 3.3)} \\
&\Leftrightarrow \quad p \sqsubseteq_\omega^{fin} q && \text{(Definition of } \sqsubseteq_\omega^{fin}\text{)}
\end{aligned}
$$

$\Box$

3.1.1 An Operational Proof of the Characterization Theorem

The use of Abramsky's domain logic in establishing the general characterization of the finitary bisimulation offered in Thm. 3.2 has, we believe, led to an elegant proof of that result based on a logical view of the preorders $\sqsubseteq_n^A$, for every finite set of actions A and non-negative integer n. In addition, it has allowed us to discuss the inequivalence of the two formulations of the domain logic $\mathcal{L}_\omega$ given by Abramsky in the references [2, 4]. We trust that this remark will be of independent interest to researchers who want to apply the domain logic for transition systems in their studies.

An alternative proof of Thm. 3.2 can, however, be given by using purely operational considerations. As the type of argument used in this alternative proof recurs, sometimes incorrectly (cf. [10] and the remarks in [4, page 212]), in several papers on process theory for specific classes of transition systems (cf., e.g., [15, 16]), we find it interesting to present its general version below.

Let $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ be an arbitrary lts. We note, first of all, that the finitary bisimulation over such an lts has the property that, for all $p, q \in \mathsf{Proc}$,

$$
p \lesssim^F q \quad \text{iff} \quad \text{for every } t \in \mathsf{ST}(\mathsf{Act}),\ \ t \lesssim^F p \text{ implies } t \lesssim^F q\ . \tag{5}
$$

This is an immediate consequence of the fact that, for every finite synchronization tree t and process p,

$$t \lesssim p \quad \Leftrightarrow \quad t \lesssim^F p \ .$$

A binary relation over processes that enjoys property (5) is usually called *finitary* or *finitely approximable* [10, 6]. It is immediate to see that two finitary relations that coincide over $\mathsf{ST}(\mathsf{Act}) \times \mathsf{Proc}$ do, in fact, coincide over the whole of Proc. To prove Thm. 3.2, it is therefore sufficient to show the following two statements:

(S1) For every $t \in \mathsf{ST}(\mathsf{Act}), p \in \mathsf{Proc}$, $t \lesssim p$ iff $t \sqsubseteq_\omega^{fin} p$.

(S2) The preorder $\sqsubseteq_\omega^{fin}$ is finitary, i.e.,

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

We now proceed to prove these two statements in turn. In what follows, we shall make use of the notion of *height* of a synchronization tree. This is the ordinal defined thus:

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

Proof of Statement S1: We show that, for every $t \in \mathsf{ST}(\mathsf{Act}), p \in \mathsf{Proc}$,

$$t \lesssim p \text{ iff } t \sqsubseteq_\omega^{fin} p.$$

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

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

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 $\mathsf{ht}(t)$ and uses Propn. 3.1(2)-(3). □

We now proceed to prove statement S2. To this end, we shall need a few 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 (3) and (4), 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 $p^{(A,n+1)}$ is a finite synchronization tree even when $\mathsf{derivatives}(p,a)$ 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 S2 above, i.e., that the preorder $\sqsubseteq_\omega^{fin}$ is finitary.

Proof of Statement S2: We prove that

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

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 $t \in \mathsf{ST}(\mathsf{Act})$,

$$t \sqsubseteq_\omega^{fin} p \quad \Rightarrow \quad t \sqsubseteq_\omega^{fin} q \ . \tag{6}$$

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 $p \sqsubseteq_n^A p^{(A,n)} \lesssim p$ (Lem. 3.4). As $p^{(A,n)}$ is a finite synchronization tree, it follows that $p \sqsubseteq_n^A p^{(A,n)} \sqsubseteq_\omega^{fin} p$ (Statement S1). Using (6), 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. □

The operational proof of Thm. 3.2 is now complete.

3.2 Transition Systems over a Countable Action Set

The behavioural characterization of the finitary bisimulation presented in Thm. 3.2, like the logical one given by Abramsky, applies to arbitrary lts's. For example, it can be used as an observation-independent version of $\lesssim^F$ over lts's with an uncountable action set. This means, in particular, that such a characterization applies to the bisimulation preorder for, e.g., timed calculi which postulate an uncountable time domain like, e.g., Wang's TCCS [33].

However, the lts's giving operational semantics to most standard process calculi, e.g. those for ACP [8], CCS [24] and CSP [14], have countable action sets. For such transition systems, it is possible to give an alternative characterization of the preorder $\lesssim^F$ which is, in many ways, easier to work with than the one provided by Thm. 3.2. This we now proceed to present.

In the remainder of this study, we shall assume that the set of actions Act is *countable*.

Let $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ be an lts with $\mathsf{Act} = \{a_1, a_2, \ldots\}$ a countable action set. For every $n \geq 0$, we write Act_n for the set of actions $\{a_1, \ldots, a_n\}$. We now define:

$$
\begin{array}{lll}
p \sqsubseteq_0 q & \Leftrightarrow & \text{true} \\
p \sqsubseteq_{n+1} q & \Leftrightarrow & (1)\ \ \forall a \in \mathsf{Act}_{n+1}, p' \in \mathsf{Proc}.\ p \stackrel{a}{\rightarrow} p' \Rightarrow \exists q' :\ q \stackrel{a}{\rightarrow} q' \text{ and } p' \sqsubseteq_n q' \\
 & & (2)\ \ \text{If } \mathsf{initials}(p) \subseteq \mathsf{Act}_{n+1} \text{ and } p \downarrow \text{ then} \\
 & & \qquad (2.1)\ \ \mathsf{initials}(q) \subseteq \mathsf{Act}_{n+1} \text{ and } q \downarrow \\
 & & \qquad (2.2)\ \ \forall a \in \mathsf{Act}_{n+1}, q' \in \mathsf{Proc}.\ q \stackrel{a}{\rightarrow} q' \Rightarrow \exists p' :\ p \stackrel{a}{\rightarrow} p' \text{ and } p' \sqsubseteq_n q' \ .
\end{array}
$$

Finally, for all $p, q \in \mathsf{Proc}$,

$$p \sqsubseteq_\omega q \ \triangleq\ \forall n \geq 0.\ p \sqsubseteq_n q \ .$$

Proposition 3.5 *Let* $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ *be an lts with* Act *a countable action set. Then:*

1. *The relations* $\sqsubseteq_n$ $(n \geq 0)$ *and* $\sqsubseteq_\omega$ *are preorders.*
2. *For all* $p, q \in \mathsf{Proc}$*,* $n \geq 0$*, if* $p \sqsubseteq_{n+1} q$*, then* $p \sqsubseteq_n q$*.*

The reader familiar with the literature on value-passing versions of Milner's CCS (cf., e.g., [13, 15, 16]) might have noticed that the above definitions are inspired by the characterizations of the finitary bisimulation presented in the aforementioned references. The main difference in the definition we present is the presence of the condition on the set of initial actions of process p in clause (2) of the definition of $\sqsubseteq_{n+1}$. The significance of this change is witnessed by the example on page 11.

Definition 3.6 *Let* $\mathsf{Act} = \{a_1, a_2, \ldots\}$ *be a countable action set. For every* $A \subseteq_{fin} \mathsf{Act}$*, we define*

$$\imath(A) \ \triangleq\ \min\{n \mid A \subseteq \mathsf{Act}_n\} \ .$$

Theorem 3.7 *For* $p, q \in \mathsf{Proc}$ *in any transition system over a countable action set* Act*,*

$$p \lesssim^F q \ \Leftrightarrow\ p \sqsubseteq_\omega q \ .$$

Proof: In view of Thm. 3.2, it is sufficient to prove that, for all $p, q \in \mathsf{Proc}$,

$$p \sqsubseteq_\omega^{fin} q \ \Leftrightarrow\ p \sqsubseteq_\omega q \ .$$

This we now proceed to show by establishing the two implications separately.

- 'Only If Implication'. We prove that, for all $p, q \in \mathsf{Proc}$ and $n \geq 0$,

  $$p \sqsubseteq_n^{\mathsf{Act}_n} q \ \Rightarrow\ p \sqsubseteq_n q$$

  from which the claim follows immediately.

  The above statement is proven by mathematical induction on n. The basis of the inductive argument is trivial as $\sqsubseteq_0$ is the universal relation over Proc. For the inductive step, assume that $p \sqsubseteq_{n+1}^{\mathsf{Act}_{n+1}} q$. It is now a simple matter to show that $p \sqsubseteq_{n+1} q$, using the inductive hypothesis and Propn. 3.1(2)-(3).

- 'IF IMPLICATION'. It is sufficient to prove that, for every finite set of actions A, $n \geq 0$, $p, q \in \mathsf{Proc}$,

$$p \sqsubseteq_{\imath(A)+n} q \quad \Rightarrow \quad p \sqsubseteq_n^A q \ .$$

  The easy proof by mathematical induction on n is left to the reader.

The proof of the theorem is now complete. $\square$

Remark: The characterization of the finitary bisimulation presented in the above theorem applies, for instance, to the versions of Milner's SCCS [23] considered in [10, 4]. To the best of our knowledge, this is the first explicit behavioural characterization of the finitary bisimulation for the class of lts's denotable by SCCS processes presented in the literature. Indeed, as remarked in [4, page 212], the characterization of $\lesssim^F$ for SCCS in [10] was obtained under the implicit, and incorrect, assumption that SCCS processes are sort-finite.

The characterizations of the finitary bisimulation that we have so far presented apply to any lts with arbitrary or countable action sets. However, it is often the case that the lts's specified by standard process description languages satisfy some sort of finiteness conditions, e.g., that they are weakly finite branching or sort-finite. We shall now provide more specific characterizations of $\lesssim^F$ for several interesting classes of lts's.

3.2.1 Transition Systems with Bounded Convergent Sort

We begin by studying a, to the best of our knowledge novel, class of lts's, viz. those with bounded convergent sort (see Def. 3.8). Apart from its intrinsic interest, the characterization of the finitary bisimulation that we shall offer for this kind of lts's will pave the way to a sharpened version of Propn. 6.13 in [4].

Definition 3.8 *Let* $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ *be an lts. For every* $p \in \mathsf{Proc}$, $n \geq 0$, *we define:*

$$\mathsf{Csort}(p,n) \triangleq \bigcup \left\{ \mathsf{initials}(q) \mid q \downarrow \text{ and there exists } \sigma \in \mathsf{Act}^* \text{ of length at most } n\text{: } p \xrightarrow{\sigma} q \right\}$$

We say that an lts has bounded convergent sort *iff* $\mathsf{Csort}(p,n)$ *is finite for all* $p \in \mathsf{Proc}$ *and* $n \geq 0$.

Intuitively, in an lts with bounded convergent sort, if we unfold the behaviour of a process p and cut the resulting synchronization tree at depth n, we obtain a synchronization tree whose convergent processes have capabilities drawn from a finite set of actions. Of course, every lts that is finite branching has *a fortiori* bounded convergent sort. However, there are lts that have bounded convergent sort, but are not finite branching. For example, the synchronization tree

$$t \quad \triangleq \quad \sum_{i \geq 1} \underbrace{a : \cdots : a :}_{i\text{-times}} \mathbb{O} \tag{7}$$

is sort-finite, and therefore has bounded convergent sort, but it is infinite branching. An example of an lts which has bounded convergent sort and has an infinite sort is

$$p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \xrightarrow{a_3} \cdots \tag{8}$$

where each process p_i $(i \geq 0)$ is convergent. This lts is, up to isomorphism, the one associated with the CCS process:

$$\textsc{All-Actions} \stackrel{\text{def}}{=} a_1.(\textsc{All-Actions}[next])$$

where $next$ is the relabelling mapping each action a_i to its successor a_{i+1} $(i \geq 1)$.

In general, the condition of weak finite branching is incomparable with that of having bounded convergent sort. The synchronization tree given in (7) has bounded convergent sort, but is not weakly finite branching. An example of a weakly finite branching synchronization tree which does not have bounded convergent sort is the following:

$$u \triangleq \left(\sum_{i \geq 0} a : b_i : \mathbb{O}\right) + \Omega$$

where we assume that the actions b_i are pairwise distinct. This synchronization tree is, up to isomorphism, the one associated with the CCS process:

$$X \stackrel{\text{def}}{=} X[f] + a.b_0.\textsc{Nil}$$

where f is the relabelling mapping each action b_i to its successor b_{i+1} $(i \geq 0)$, and leaving a fixed.

Both the codings in CCS (with finite summation) of sort-infinite synchronization trees that we have presented make use of infinite relabellings, i.e., relabellings that change the nature of infinitely many actions. This is unavoidable because, as first observed by Abramsky [3, 4], CCS with finite relabellings can only describe sort-finite, and thus convergent sort bounded, lts's. CCS with infinite relabellings, but only guarded recursion, can only specify finite branching, and thus convergent sort bounded, processes. As (8) shows, these processes need not be sort-finite.

After having discussed the bounded convergent sort condition, we now proceed to present a characterization of the finitary bisimulation for this class of lts's.

Let $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ be an lts with $\mathsf{Act} = \{a_1, a_2, \ldots\}$ a countable action set. We now define:

$$\begin{array}{lll} p \sqsubseteq_{\approx 0} q & \Leftrightarrow & \text{true} \\ p \sqsubseteq_{\approx n+1} q & \Leftrightarrow & (1)\ \forall a \in \mathsf{Act}_{n+1}, p' \in \mathsf{Proc}.\ p \xrightarrow{a} p' \Rightarrow \exists q' :\ q \xrightarrow{a} q' \text{ and } p' \sqsubseteq_{\approx n} q' \\ & & (2)\ \text{If } p\downarrow \text{ then} \\ & & \quad (2.1)\ q\downarrow \\ & & \quad (2.2)\ \forall a \in \mathsf{Act}_{n+1}, q' \in \mathsf{Proc}.\ q \xrightarrow{a} q' \Rightarrow \exists p' :\ p \xrightarrow{a} p' \text{ and } p' \sqsubseteq_{\approx n} q' \ . \end{array}$$

Finally, for all $p, q \in \mathsf{Proc}$,

$$p \sqsubseteq_{\approx \omega} q \triangleq \forall n \geq 0.\ p \sqsubseteq_{\approx n} q \ .$$

Proposition 3.9 *Let* $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ *be an lts with* Act *a countable action set. Then:*

1. *The relations* $\sqsubseteq_{\approx n}$ $(n \geq 0)$ *and* $\sqsubseteq_{\approx \omega}$ *are preorders.*

2. *For all* $p, q \in \mathsf{Proc}$, $n \geq 0$, *if* $p \sqsubseteq_{\approx n+1} q$, *then* $p \sqsubseteq_{\approx n} q$.

The definition of the preorder $\underset{\approx}{\sqsubset}_\omega$ is very similar to that of the preorder $\lesssim_\omega$ given on page 6. The only difference consists in the set of actions an observer, in the sense of [3], is allowed to use in experiments of depth n. The requirement that the actions be drawn from the set Act_n, which is finite, guarantees that the resulting experiments be of finite width as well as depth. Note that, unlike in the definition of the preorder $\underset{\sim}{\sqsubset}_{n+1}$, in clause (2) above we need *not* require that $\mathsf{initials}(p)$ is contained in Act_{n+1}. As the proof of the following result shows, this requirement is redundant for convergent sort bounded processes. Indeed, the requirement that $\mathsf{initials}(p) \subseteq \mathsf{Act}_{n+1}$ for some n, was needed to ensure the testability of the convergence of a process. If the lts under consideration has bounded convergent sort, then every convergent process can only initially perform finitely many distinct actions. As argued in the remark on page 14, convergence of these processes can always be detected using finite observations.

Theorem 3.10 *For $p, q \in \mathsf{Proc}$ in any transition system over a countable action set Act which has bounded convergent sort,*

$$p \lesssim^F q \;\Leftrightarrow\; p \underset{\approx}{\sqsubset}_\omega q \;.$$

Proof: Consider a transition system over a countable action set Act which has bounded convergent sort. In view of Thm. 3.7, it is sufficient to prove that, for all $p, q \in \mathsf{Proc}$,

$$p \underset{\sim}{\sqsubset}_\omega q \;\Leftrightarrow\; p \underset{\approx}{\sqsubset}_\omega q \;.$$

We show the two implications in the above statement separately.

- 'ONLY IF IMPLICATION'. To prove that, for all $p, q \in \mathsf{Proc}$,

  $$p \underset{\sim}{\sqsubset}_\omega q \;\Rightarrow\; p \underset{\approx}{\sqsubset}_\omega q \;.$$

  it is sufficient to show that, for all $p, q \in \mathsf{Proc}$, $n \geq 0$:

  $$p \underset{\sim}{\sqsubset}_{\imath(\mathsf{Csort}(p,n))+n}\, q \;\Rightarrow\; p \underset{\approx}{\sqsubset}_n q \;. \tag{9}$$

  Note that $\imath(\mathsf{Csort}(p, n))$ is a well-defined non-negative integer because, as the lts has bounded convergent sort, $\mathsf{Csort}(p, n)$ is a finite set of actions. To ease readability, throughout the proof of (9) we use $\imath(p, n)$ as a shorthand for $\imath(\mathsf{Csort}(p, n))$.

  We prove statement (9) by mathematical induction on n. The basis of the induction is immediate because $\underset{\approx}{\sqsubset}_0$ is the universal relation. For the inductive step, we assume that $p \underset{\sim}{\sqsubset}_{\imath(p,n+1)+n+1} q$, and prove that $p \underset{\approx}{\sqsubset}_{n+1} q$ holds. To this end, it is sufficient to show that the defining clauses of $\underset{\approx}{\sqsubset}_{n+1}$ hold for p and q.

  - CLAUSE (1). Assume that $p \xrightarrow{a} p'$ and $a \in \mathsf{Act}_{n+1}$. Note that this means that $a \in \mathsf{Act}_{\imath(p,n+1)+n+1}$. Therefore, as $p \underset{\sim}{\sqsubset}_{\imath(p,n+1)+n+1} q$, there exists $q' \in \mathsf{derivatives}(q, a)$ such that

    $$p' \quad \underset{\sim}{\sqsubset}_{\imath(p,n+1)+n} \quad q' \;. \tag{10}$$

    Note now that, as $p \xrightarrow{a} p'$, $\mathsf{Csort}(p', n)$ is included in $\mathsf{Csort}(p, n + 1)$. From this observation, it follows that

    $$\imath(p', n) + n \leq \imath(p, n + 1) + n \;.$$

As the sequence of preorders $\{\sqsubseteq_k \mid k \geq 0\}$ is decreasing with respect to set inclusion (Propn. 3.5(2)), (10) implies that

$$p' \quad \sqsubseteq_{\imath(p',n)+n} \quad q' \ . \tag{11}$$

An application of the inductive hypothesis for statement (9) to (11) now yields

$$p' \sqsubseteq_{\approx n} q'$$

as desired.

– CLAUSE (2.1). Assume that $p\downarrow$. We shall prove that $q\downarrow$. First of all, note that, as $p\downarrow$,

$$\mathsf{initials}(p) \subseteq \mathsf{Csort}(p, n+1) \subseteq \mathsf{Act}_{\imath(p,n+1)+n+1} \ .$$

Now, as $p \sqsubseteq_{\imath(p,n+1)+n+1} q$, $q\downarrow$ follows immediately. Moreover, for use in the remainder of the proof, we also observe that $\mathsf{initials}(q) \subseteq \mathsf{Act}_{\imath(p,n+1)+n+1}$ follows.

– CLAUSE (2.2). Assume that $p\downarrow$, $q\downarrow$ and $q \xrightarrow{a} q'$ for some $a \in \mathsf{Act}_{n+1}$. We shall show that $p \xrightarrow{a} p'$ for some $p' \sqsubseteq_{\approx n} q'$. To this end, note, first of all, that we know that $\mathsf{initials}(p) \subseteq \mathsf{Act}_{\imath(p,n+1)+n+1}$ and $\mathsf{initials}(q) \subseteq \mathsf{Act}_{\imath(p,n+1)+n+1}$ hold from our analysis of clause (2.1). Therefore, as $p \sqsubseteq_{\imath(p,n+1)+n+1} q$ and $a \in \mathsf{Act}_{n+1} \subseteq \mathsf{Act}_{\imath(p,n+1)+n+1}$, it follows that there exists $p' \in \mathsf{derivatives}(p, a)$ such that:

$$p' \sqsubseteq_{\imath(p,n+1)+n} q' \ .$$

Reasoning as in our argument for clause (1), we infer that

$$p' \quad \sqsubseteq_{\imath(p',n)+n} \quad q' \tag{12}$$

also holds. An application of the inductive hypothesis for statement (9) to (12) now yields

$$p' \sqsubseteq_{\approx n} q'$$

as desired.

The proof of the inductive step is now complete.

- 'IF IMPLICATION'. To prove that, for all $p, q \in \mathsf{Proc}$,

$$p \sqsubseteq_{\approx\omega} q \quad \Rightarrow \quad p \sqsubseteq_{\omega} q \ .$$

it is sufficient to show that, for all $p, q \in \mathsf{Proc}$, $n \geq 0$:

$$p \sqsubseteq_{\approx \imath(\mathsf{Csort}(q,n))+n} q \quad \Rightarrow \quad p \sqsubseteq_n q \ . \tag{13}$$

Again, note that $\imath(\mathsf{Csort}(q, n))$ is a well-defined non-negative integer because, as the lts has bounded convergent sort, $\mathsf{Csort}(q, n)$ is a finite set of actions. To ease readability, throughout the proof we use $\imath(q, n)$ as a shorthand for $\imath(\mathsf{Csort}(q, n))$.

We prove statement (13) by mathematical induction on n. The basis of the induction is immediate because $\sqsubseteq_0$ is the universal relation. For the inductive step, we assume that $p \sqsubseteq_{\approx \imath(q,n+1)+n+1} q$, and prove that $p \sqsubseteq_{n+1} q$ holds. To this end, it is sufficient to show that the defining clauses of $\sqsubseteq_{n+1}$ hold for p and q.

- Clause (1). Assume that $p \xrightarrow{a} p'$ and $a \in \mathsf{Act}_{n+1}$. Note that this means that $a \in \mathsf{Act}_{\imath(q,n+1)+n+1}$. Therefore, as $p \sqsubseteq_{\approx \imath(q,n+1)+n+1} q$, there exists $q' \in \mathsf{derivatives}(q,a)$ such that

$$p' \quad \sqsubseteq_{\approx \imath(q,n+1)+n} \quad q' \ . \tag{14}$$

Note now that, as $q \xrightarrow{a} q'$, $\mathsf{Csort}(q',n) \subseteq \mathsf{Csort}(q,n+1)$. From this observation, it follows that

$$\imath(q',n) + n \leq \imath(q,n+1) + n \ .$$

As the sequence of preorders $\left\{ \sqsubseteq_{\approx k} \mid k \geq 0 \right\}$ is decreasing with respect to set inclusion (Propn. 3.9(2)), (14) implies that

$$p' \quad \sqsubseteq_{\approx \imath(q',n)+n} \quad q' \ . \tag{15}$$

An application of the inductive hypothesis for statement (13) to (15) now yields

$$p' \sqsubseteq_{\sim n} q'$$

as desired.

- Clause (2.1). Assume that $\mathsf{initials}(p) \subseteq \mathsf{Act}_{n+1}$ and $p \downarrow$. We shall prove that $\mathsf{initials}(q) \subseteq \mathsf{Act}_{n+1}$ and $q \downarrow$. First of all, note that, as $p \sqsubseteq_{\approx \imath(q,n+1)+n+1} q$ and $p \downarrow$, it follows immediately that $q \downarrow$. We shall now argue that $\mathsf{initials}(q) \subseteq \mathsf{Act}_{n+1}$ also holds. To this end, assume that $q \xrightarrow{a} q'$ for some q'. As $q \downarrow$, the action a is contained in $\mathsf{Csort}(q, n+1)$. Therefore a is also contained in $\mathsf{Act}_{\imath(q,n+1)+n+1}$. As $p \sqsubseteq_{\approx \imath(q,n+1)+n+1} q$, $p \downarrow$, $q \downarrow$ and $q \xrightarrow{a} q'$ with $a \in \mathsf{Act}_{\imath(q,n+1)+n+1}$, we infer that

$$p \xrightarrow{a} p', \text{ for some } p' \ .$$

As $\mathsf{initials}(p) \subseteq \mathsf{Act}_{n+1}$ by assumption, it follows that a is contained in Act_{n+1}, which was to be shown.

- Clause (2.2). Assume that $p \downarrow$, $q \downarrow$, $\mathsf{initials}(p) \subseteq \mathsf{Act}_{n+1}$, $\mathsf{initials}(q) \subseteq \mathsf{Act}_{n+1}$ and $q \xrightarrow{a} q'$ for some $a \in \mathsf{Act}_{n+1}$. We shall show that $p \xrightarrow{a} p'$ for some $p' \sqsubseteq_{\sim n} q'$. To this end, note that, as $p \sqsubseteq_{\approx \imath(q,n+1)+n+1} q$ and $a \in \mathsf{Act}_{n+1} \subseteq \mathsf{Act}_{\imath(q,n+1)+n+1}$, it follows that there exists $p' \in \mathsf{derivatives}(p,a)$ such that:

$$p' \sqsubseteq_{\approx \imath(q,n+1)+n} q' \ .$$

Reasoning as in our argument for clause (1), we infer that

$$p' \quad \sqsubseteq_{\approx \imath(q',n)+n} \quad q' \tag{16}$$

also holds. An application of the inductive hypothesis for statement (13) to (16) now yields

$$p' \sqsubseteq_{\sim n} q'$$

as desired.

The proof of the inductive step is now complete.

This completes the proof of the theorem. $\square$

It is interesting to note that above theorem would *not* hold for lts's that do not have bounded convergent sort, as shown by the following example.

Example: Let $\mathsf{Act} = \{a_1, a_2, a_3, \ldots\}$ be a countably infinite set of actions. Define the synchronization trees:

$$\begin{array}{rcl} t_1 & \triangleq & a_1 : \mathbb{O} + \Omega \\ t_\omega & \triangleq & \left(\sum_{n\geq 1} a_1 : a_n : \mathbb{O}\right) + \Omega \end{array}$$

Note that t_ω does *not* have bounded convergent sort. We claim that $t_1 \sqsubseteq_\omega t_\omega$, but $t_1 \not\lesssim^F t_\omega$. First of all, note that, as t_1 is a finite synchronization tree, the fact that $t_1 \not\lesssim^F t_\omega$ follows immediately from the observation that $t_1 \not\lesssim t_\omega$. To see that $t_1 \sqsubseteq_\omega t_\omega$ does instead hold, note that, for every $n \geq 1$,

$$\mathbb{O} \sqsubseteq_n a_{n+1} : \mathbb{O} \ .$$

This fact guarantees that $t_1 \sqsubseteq_n t_\omega$ for every $n \geq 0$. $\square$

3.2.2 Sort-finite Transition Systems

In [4, 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 $\lesssim_\omega$ defined on page 6. Using 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 3.11 *For $p, q \in \mathsf{Proc}$ in any sort-finite transition system over a countable action set Act,*

$$p \lesssim^F q \ \Leftrightarrow \ p \lesssim_\omega q \ .$$

Proof: First of all, note that, in any lts, not necessarily sort-finite, $p \lesssim_\omega q$ implies $p \lesssim^F q$. In fact, for all p, q,

$$\begin{array}{rcll} p \lesssim_\omega q & \Leftrightarrow & p \sqsubseteq_\omega^{\mathsf{Act}} q & \\ & \Rightarrow & p \sqsubseteq_\omega^{fin} q & \text{(Propn. 3.1(2))} \\ & \Leftrightarrow & p \lesssim^F q & \text{(Thm. 3.2)} \ . \end{array}$$

We are therefore left to show that, for $p, q \in \mathsf{Proc}$ in any sort-finite transition system over a countable action set Act,

$$p \lesssim^F q \ \Rightarrow \ p \lesssim_\omega q \ .$$

To this end, note, first of all, that any sort-finite lts has *a fortiori* bounded convergent sort. In view of Thm. 3.10, it is therefore sufficient to show that, for all $p, q \in \mathsf{Proc}$,

$$p \sqsubseteq_\omega q \ \Rightarrow \ p \lesssim_\omega q \ .$$

This will follow if we show that, for all $p, q \in \mathsf{Proc}$, $n \geq 0$,

$$p \sqsubseteq_{\imath(p,q)+n} q \ \Rightarrow \ p \lesssim_n q \tag{17}$$

where we use $\imath(p,q)$ as a shorthand for $\imath(\mathsf{sort}(p) \cup \mathsf{sort}(q))$, which is a well-defined non-negative integer as the lts is sort-finite.

We prove statement (17) by mathematical induction on n. The basis of the induction is trivial because $\lesssim_0$ is the universal relation. For the inductive step, let us assume that $p \sqsubseteq_{\imath(p,q)+n+1} q$. We show that $p \lesssim_{n+1} q$ holds. To this end, we proceed to check that the defining clauses of $\lesssim_{n+1}$ are met by p and q.

1. Assume that $p \xrightarrow{a} p'$. We shall prove that $q \xrightarrow{a} q'$ for some q' such that $p' \lesssim_n q'$.
As $a \in \mathsf{sort}(p) \subseteq \mathsf{Act}_{\imath(p,q)+n+1}$ and $p \sqsubseteq_{\approx\imath(p,q)+n+1} q$, it follows that $q \xrightarrow{a} q'$ for some q' such that
$$p' \quad \sqsubseteq_{\approx\imath(p,q)+n} \quad q' \ . \tag{18}$$
Note now that, as $\mathsf{sort}(p') \subseteq \mathsf{sort}(p)$ and $\mathsf{sort}(q') \subseteq \mathsf{sort}(q)$, it follows that $\imath(p',q') \leq \imath(p,q)$. As the sequence of preorders $\left\{\sqsubseteq_{\approx k} \mid k \geq 0\right\}$ is decreasing with respect to set inclusion (Propn. 3.9(2)), (18) implies
$$p' \quad \sqsubseteq_{\approx\imath(p',q')+n} \quad q' \ . \tag{19}$$
We may now apply the inductive hypothesis for statement (17) to (19) to infer that
$$p' \lesssim_n q'$$
as desired.
2. Assume that $p \downarrow$. Then $q \downarrow$ is immediate from the fact that $p \sqsubseteq_{\approx\imath(p,q)+n+1} q$.
3. Assume that $p \downarrow$, $q \downarrow$ and $q \xrightarrow{a} q'$. We shall prove that $p \xrightarrow{a} p'$ for some p' such that $p' \lesssim_n q'$.
As $a \in \mathsf{sort}(q) \subseteq \mathsf{Act}_{\imath(p,q)+n+1}$ and $p \sqsubseteq_{\approx\imath(p,q)+n+1} q$, it follows that $p \xrightarrow{a} p'$ for some p' such that
$$p' \sqsubseteq_{\approx\imath(p,q)+n} q' \ .$$
Now reasoning as in point 1 above, we may infer that
$$p' \sqsubseteq_{\approx\imath(p',q')+n} q'$$
also holds. Again, an application of the inductive hypothesis gives that $p' \lesssim_n q'$ as desired.

This completes the inductive proof of statement (17). $\square$

Remark: An alternative proof of the above result can be given by mimicking the proof of statement S2 presented in Sect. 3.1.1. 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 $p \lesssim_\omega q$.

Note that the above argument applies regardless of the cardinality of the action set Act and of $\mathsf{sort}(q)$.

3.2.3 A Class of Image Finite Transition Systems

To conclude our taxonomy of characterizations of the finitary bisimulation, we present a class of lts's with countable action set which do not necessarily have bounded convergent sort, but for which the finitary bisimulation coincides with $\mathrel{\underset{\approx}{\sqsubset}}_\omega$. We believe that this result is mainly of theoretical interest, but we find it worthwhile to include it for the sake of completeness.

Proposition 3.12 *Let $p, q \in \mathsf{Proc}$ in any transition system over a countable action set Act. Assume furthermore that the transition system is image finite, and weakly initials finite. Then*

$$p \lesssim^F q \;\Leftrightarrow\; p \mathrel{\underset{\approx}{\sqsubset}}_\omega q \;.$$

Proof: Consider a transition system over a countable action set Act which is image finite and weakly initials finite. In view of Thm. 3.7, it is sufficient to prove that, for all $p, q \in \mathsf{Proc}$,

$$p \mathrel{\underset{\sim}{\sqsubset}}_\omega q \quad\Leftrightarrow\quad p \mathrel{\underset{\approx}{\sqsubset}}_\omega q \;.$$

We show the two implications in the above statement separately.

- 'Only If Implication'. We show that, for all $p, q \in \mathsf{Proc}$, $n \geq 0$

  $$p \mathrel{\underset{\sim}{\sqsubset}}_\omega q \quad\Rightarrow\quad p \mathrel{\underset{\approx}{\sqsubset}}_n q \;.$$

  The proof is by mathematical induction on n. The basis of the inductive argument is trivial as $\mathrel{\underset{\approx}{\sqsubset}}_0$ is the universal relation over Proc. To establish the inductive step, let us assume that $p \mathrel{\underset{\sim}{\sqsubset}}_\omega q$ holds. We prove that $p \mathrel{\underset{\approx}{\sqsubset}}_{n+1} q$ also holds. To this end, it is sufficient to show that the defining clauses of $\mathrel{\underset{\approx}{\sqsubset}}_{n+1}$ are met by p and q.

  - Clause (1). Assume that a is an action in Act_{n+1} and $p \xrightarrow{a} p'$. As $p \mathrel{\underset{\sim}{\sqsubset}}_\omega q$ and a is contained in Act_m for every $m \geq n+1$, for every such positive integer m there exists a process q_m such that

    $$q \xrightarrow{a} q_m \;\text{ and }\; p' \mathrel{\underset{\sim}{\sqsubset}}_m q_m \;.$$

    As the lts is image-finite, the set of processes $\{q_m \mid m \geq n+1\}$ is finite. This implies that, for some q' contained in $\{q_m \mid m \geq n+1\}$, $p' \mathrel{\underset{\sim}{\sqsubset}}_m q'$ for infinitely many positive integers m. As the sequence of preorders $\{\mathrel{\underset{\sim}{\sqsubset}}_k \mid k \geq\}$ is decreasing (Propn. 3.1(2)), it follows that $p' \mathrel{\underset{\sim}{\sqsubset}}_m q'$ for all $m \geq 0$, i.e., that $p' \mathrel{\underset{\sim}{\sqsubset}}_\omega q'$. By induction, we now obtain $p' \mathrel{\underset{\approx}{\sqsubset}}_n q'$ as desired.
  - Clause (2.1). Assume that $p \downarrow$. We shall prove that $q \downarrow$. As the lts under consideration is weakly initials finite, the set of initial actions from p is finite. Let $M \triangleq \imath(\mathsf{initials}(p))+1$. As $p \mathrel{\underset{\sim}{\sqsubset}}_\omega q$, it follows that $p \mathrel{\underset{\sim}{\sqsubset}}_M q$. As $p \downarrow$ and $\mathsf{initials}(p) \subseteq \mathsf{Act}_M$, we infer that $q \downarrow$, which was to be shown. Moreover, for use in the remainder of the proof, we also obtain that $\mathsf{initials}(q) \subseteq \mathsf{Act}_M$.
  - Clause (2.2). Assume that $p \downarrow$, $q \downarrow$, $a \in \mathsf{Act}_{n+1}$ and $q \xrightarrow{a} q'$. By a previous observation, it follows that $a \in \mathsf{Act}_M$, where M is the positive integer defined in the proof for clause (2.1). As $p \mathrel{\underset{\sim}{\sqsubset}}_\omega q$, $p \downarrow$, $q \downarrow$, $\mathsf{initials}(p) \subseteq \mathsf{Act}_M$, $\mathsf{initials}(q) \subseteq \mathsf{Act}_M$ and Act_M is included in Act_m for every $m \geq M$, we infer that, for every such positive integer m, there exists a process p_m such that

    $$p \xrightarrow{a} p_m \;\text{ and }\; p_m \mathrel{\underset{\sim}{\sqsubset}}_m q' \;.$$

    Reasoning as in the proof for clause (1) above, we may conclude that there is a process $p' \in \mathsf{derivatives}(p, a)$ such that $p' \mathrel{\underset{\sim}{\sqsubset}}_\omega q'$. By induction, we now obtain $p' \mathrel{\underset{\approx}{\sqsubset}}_n q'$ as desired.

This completes the proof for the inductive step.

- 'If Implication'. We show that, for all $p, q \in \mathsf{Proc}$ in an image finite lts, $n \geq 0$,

$$p \sqsubsetapprox_{\omega} q \quad \Rightarrow \quad p \sqsubsetsim_n q \ .$$

  The proof is by mathematical induction on n, and is very similar to the one presented above. For this reason, we limit ourselves to presenting the argument for clause (2.1) of the definition of $\sqsubsetsim_{n+1}$. To this end, assume that $p \sqsubsetapprox_{\omega} q$, $p \downarrow$ and $\mathsf{initials}(p) \subseteq \mathsf{Act}_{n+1}$. We prove that $q \downarrow$ and that $\mathsf{initials}(q)$ is also included in Act_{n+1}. The fact that q definitely converges is an immediate consequence of the assumption that $p \downarrow$ and $p \sqsubsetapprox_1 q$. To see that every action a contained in $\mathsf{initials}(q)$ is also in Act_{n+1}, we reason as follows. Assume that $q \xrightarrow{a} q'$. Pick a positive integer N such that a is contained in Act_N. As $p \sqsubsetapprox_N q$, $p \downarrow$, $q \downarrow$, $a \in \mathsf{Act}_N$ and $q \xrightarrow{a} q'$, it follows that $p \xrightarrow{a} p'$ for some $p' \sqsubsetapprox_{N-1} q'$. That is $a \in \mathsf{initials}(p) \subseteq \mathsf{Act}_{n+1}$, which was to be shown.

The proof of the theorem is now complete. $\square$

The examples on pages 11 and 24 show that neither of the constraints in the proviso of the above result can be omitted.

Acknowledgements: We have benefited from illuminating discussions with Samson Abramsky on the two different versions of the domain logic for transition systems presented in [2, 4]. Our overall debt to his seminal work will be evident to the readers of this paper.

References

[1] S. Abramsky, *Experiments, powerdomains and fully abstract models for applicative multiprogramming*, in Foundations of Computation Theory, M. Karpinski, ed., vol. 154 of Lecture Notes in Computer Science, Springer-Verlag, 1983, pp. 1–13.

[2] ——, *Domain Theory and the Logic of Observable Properties*, PhD thesis, University of London, 1987.

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

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

[5] ——, *Domain theory in logical form*, Annals of Pure and Applied Logic, 51 (1991), pp. 1–77.

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

[7] K. Apt and G. Plotkin, *Countable nondeterminism and random assignment*, J. Assoc. Comput. Mach., 33 (1986), pp. 724–767.

[8] J. Baeten and W. Weijland, *Process Algebra*, Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, 1990.

[9] I. Guessarian, *Algebraic Semantics*, vol. 99 of Lecture Notes in Computer Science, Springer-Verlag, 1981.

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

[11] ——, *Algebraic Theory of Processes*, MIT Press, Cambridge, Massachusetts, 1988.

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

[13] 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.

[14] C. Hoare, *Communicating Sequential Processes*, Prentice-Hall International, Englewood Cliffs, 1985.

[15] A. Ingólfsdóttir, *Semantic Models for Communicating Process with Value-Passing*, PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, June 1994. Computer Science Report 8/94. Also available as Report R–94–2044, Department of Mathematics and Computer Science, Aalborg University.

[16] ——, *A semantic theory for value-passing processes late approach — Part II: A behavioural semantics and full abstractness*, Report RS–95–22, BRICS (Basic Research in Computer Science, Centre of the Danish National Research Foundation), Institute for Electronic Systems, Department of Mathematics and Computer Science, Aalborg University Centre, April 1995.

[17] A. Ingólfsdóttir and A. Schalk, *A fully abstract denotational model for observational congruence*, Report RS–95–40, BRICS (Basic Research in Computer Science, Centre of the Danish National Research Foundation), Institute for Electronic Systems, Department of Mathematics and Computer Science, Aalborg University Centre, August 1995. To appear in *Proceedings CSL '95*.

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

[19] A. Meyer, *Semantical paradigms: Notes for an invited lecture*, in Proceedings 3^{th} Annual Symposium on Logic in Computer Science, Edinburgh, IEEE Computer Society Press, 1988, pp. 236–242.

[20] R. Milner, *Processes: A mathematical model of computing agents*, in Proceedings Logic Colloquium 1973, H. Rose and J. Shepherdson, eds., North-Holland, 1973, pp. 158–173.

[21] ——, *Fully abstract models of typed λ-calculi*, Theoretical Comput. Sci., 4 (1977), pp. 1–22.

[22] ——, *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.

[23] ——, *Calculi for synchrony and asynchrony*, Theoretical Comput. Sci., 25 (1983), pp. 267–310.

[24] ——, *Communication and Concurrency*, Prentice-Hall International, Englewood Cliffs, 1989.

[25] 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.

[26] G. Plotkin, *LCF considered as a programming language*, Theoretical Comput. Sci., 5 (1977), pp. 223–256.

[27] ——, *Lecture notes in domain theory*, 1981. University of Edinburgh.

[28] ——, *A structural approach to operational semantics*, Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[29] D. Scott and C. Strachey, *Towards a mathematical semantics for computer languages*, in Proceedings of the Symposium on Computers and Automata, vol. 21 of Microwave Research Institute Symposia Series, 1971.

[30] C. Stirling, *Modal logics for communicating systems*, Theoretical Comput. Sci., 49 (1987), pp. 311–347.

[31] V. Stoltenberg-Hansen, I. Lindström, and E. Griffor, *Mathematical Theory of Domains*, Cambridge Tracts in Theoretical Computer Science 22, Cambridge University Press, 1994.

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

[33] Y. Wang, *Real-time behaviour of asynchronous agents*, in Proceedings CONCUR 90, Amsterdam, J. Baeten and J. Klop, eds., vol. 458 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 502–520.

Recent Publications in the BRICS Report Series

RS-95-59 **Luca Aceto and Anna Ingólfsdóttir.** ***On the Finitary Bisimulation*****. November 1995. 29 pp.**

RS-95-58 **Nils Klarlund, Madhavan Mukund, and Milind Sohoni.** ***Determinizing Asynchronous Automata on Infinite Inputs*****. November 1995. 32 pp.**

RS-95-57 **Jaap van Oosten.** ***Topological Aspects of Traces*****. November 1995. 16 pp.**

RS-95-56 **Luca Aceto, Wan J. Fokkink, Rob J. van Glabbeek, and Anna Ingólfsdóttir.** ***Axiomatizing Prefix Iteration with Silent Steps*****. November 1995. 25 pp.**

RS-95-55 **Mogens Nielsen and Kim Sunesen.** ***Trace Equivalence - Partially Decidable!*** **November 1995.**

RS-95-54 **Nils Klarlund, Mogens Nielsen, and Kim Sunesen.** ***Using Monadic Second-Order Logic with Finite Domains for Specification and Verification*****. November 1995.**

RS-95-53 **Nils Klarlund, Mogens Nielsen, and Kim Sunesen.** ***Automated Logical Verification based on Trace Abstractions*****. November 1995.**

RS-95-52 **Antonín Kučera.** ***Deciding Regularity in Process Algebras*****. October 1995. 42 pp.**

RS-95-51 **Rowan Davies.** ***A Temporal-Logic Approach to Binding-Time Analysis*****. October 1995. 11 pp.**

RS-95-50 **Dany Breslauer.** ***On Competitive On-Line Paging with Lookahead*****. September 1995. 12 pp.**

RS-95-49 **Mayer Goldberg.** ***Solving Equations in the λ-Calculus using Syntactic Encapsulation*****. September 1995. 13 pp.**

RS-95-48 **Devdatt P. Dubhashi.** ***Simple Proofs of Occupancy Tail Bounds*****. September 1995. 7 pp. To appear in** ***Random Structures and Algorithms*****.**

RS-95-47 **Dany Breslauer.** ***The Suffix Tree of a Tree and Minimizing Sequential Transducers*****. September 1995. 15 pp.**