

Back and Forth Bisimulations

Rocco De Nicola

*Istituto di Elaborazione dell' Informazione, C.N.R.
Via S. Maria 46, I-56126 Pisa, Italy*

Ugo Montanari

*Dipartimento di Informatica, Università di Pisa
Corso Italia 40, I-56125 Pisa, Italy*

Frits Vaandrager

*Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands*

This paper is concerned with bisimulation relations which do not only require related agents to simulate each others behavior in the direction of the arrows, but also to simulate each other when going back in history. First it is demonstrated that the back and forth variant of strong bisimulation leads to the same equivalence as the ordinary notion of strong bisimulation. Then it is shown that the back and forth variant of Milner's observation equivalence is different from (and finer than) observation equivalence. In fact we prove that it coincides with the branching bisimulation equivalence of Van Glabbeek & Weijland. Also the back and forth variants of branching, η and delay bisimulation lead to branching bisimulation equivalence. The notion of back and forth bisimulation moreover leads to characterizations of branching bisimulation in terms of abstraction homomorphisms and in terms of Hennessy-Milner logic with backward modalities. In our view these results support the claim that branching bisimulation is a natural and important notion.

1. INTRODUCTION

The notion of bisimulation relation has been introduced by PARK [18]. It leads to an equivalence on labelled transition systems which, in case image finiteness is assumed, coincides with the strong equivalence of MILNER [12]. The great importance and usefulness of bisimulations in the theory of concurrent systems is evident: Mathematically, bisimulation is a very pleasant notion. It is closely related to the non-well-founded sets of ACZEL [1] and leads to a natural first behavioral abstraction from transition systems. Algebraically, in the setting of CCS-like languages, bisimulations lead to elegant and simple laws [10]. Moreover, bisimulation equivalence has a beautiful characterization in terms of Hennessy-Milner logic [10]. Bisimulations are also important from a practical point of view because with the algorithm of PAIGE & TARJAN [17], bisimulation equivalence on finite state automata can be decided extremely fast in $O(m \log n)$ time (where m is the number of transitions and n the number of states).

In this paper we introduce the concept of back and forth bisimulations. In a back and forth bisimulation the agents can not only simulate each others behavior in the direction of the arrows but also when going backward in their history. In general, given the definition of any bisimulation, one can define a corresponding back and forth version of it. We want to explore the relationships between bisimulations and their back and forth variants.

Back and forth bisimulations are interesting for several reasons. First of all, it is always

First and second authors were supported by ESPRIT project 3011 (CEDISYS). The research of the third author was supported by RACE project no. 1046, Specification and Programming Environment for Communication Software (SPECS) and by ESPRIT project no. 3006 (CONCUR).

intriguing to see what are the consequences of small modifications of important definitions. More important is the connection with temporal and modal logics. These logics give rise to equivalences on transition systems and Kripke structures (two states are equivalent iff they satisfy the same formulas) and it appears to be very useful to give operational characterizations of these equivalences. A well known result in this area is that the equivalence induced by Hennessy-Milner logic (HML) formulas coincides with (strong) bisimulation equivalence [10]. Other results of this kind are for instance reported in [4]: two bisimulation-based equivalences over Kripke structures are related to two variants of CTL* [8]. In the world of temporal and modal logic, there has been a lot of interest in past-tense operators (see for instance [11, 20]). If one is looking for operational characterizations of the equivalences induced by logics with a past-tense operator, it seems natural to consider back and forth bisimulations.

In Section 2.1 of this paper, we demonstrate that the back and forth variant of strong bisimulation leads to the same equivalence as the ordinary notion of strong bisimulation. This result clarifies an earlier result of HENNESSY & STIRLING [11]. They showed that in the context of classic labelled transition systems, the extension of HML with a reverse operator does not lead to any increase in discriminating power. HENNESSY & STIRLING [11] did not consider abstraction of silent actions. In Section 2.2, we show that the back and forth variant of the (weak) observation equivalence of MILNER [15] is different from (and finer than) observation equivalence. In fact we prove that it coincides with the branching bisimulation equivalence of VAN GLABBEEK & WEIJLAND [9]. We will play the same game with other equivalences and prove that the back and forth versions of branching bisimulation, the η -bisimulation of BAETEN & VAN GLABBEEK [3] and the *delay bisimulation* of [21] (first introduced by MILNER [13] under the somewhat confusing name *observational equivalence*), all lead to branching bisimulation equivalence. Hence, branching bisimulation equivalence arises as a kind of ‘fixed point’ of the back and forth operation. This result supports the claim that branching bisimulation is a natural and important notion.

In Section 3, we study the relationships of back and forth bisimulation with abstraction homomorphisms. Abstraction homomorphisms have been introduced for the first time in [6], for labelled event structures. The tight relation of abstraction homomorphisms with bisimulation has been discussed in [5], where it is also proved that, under some rather restrictive conditions, given any transition system \mathcal{C} , there is always a unique minimal homomorphic system. Both important properties have been extended in [16] to partial ordering labelings, and they are proved to hold under significantly milder conditions. Similar results (but apparently applicable only to transition systems without τ) have been generalized to saturating quasi-homomorphisms of Φ algebras in [2] (following [19]). In the present paper, the main result about abstraction homomorphisms is the evidence of their flexibility in expressing several notions of equivalence: we show that observation equivalence can be characterized in terms of abstraction homomorphisms that preserve successors, whereas branching bisimulation corresponds with abstraction homomorphisms which preserve both successors and predecessors.

The final section presents a logical characterization of branching bisimulation equivalence in terms of *back and forth logic*, an extension of Hennessy-Milner logic with backward modalities. In order to establish this logical characterization, we essentially use the result of Section 2.2 which says that branching bisimulation and weak back and forth bisimulation coincide. The logical characterization with backward modalities is also reported in [7]. In addition that paper presents two other logics for branching bisimulation equivalence. The first is an extension of HML with a kind of ‘until’ operators, which turn out to be definable in terms of the modalities of back and forth logic. The second is CTL* without the next-time operator interpreted over all paths, not just the maximal ones.

2. OPERATIONAL CHARACTERIZATIONS

2.1. Strong bisimulation.

We start the technical part of this paper with a discussion of a simple, but fundamental case: the back and forth variant of strong bisimulation. Strong bisimulation equivalence, like all other equivalences in this paper, will be defined on the states of *labelled transition systems (LTS's)*. Below we recall the definitions of a LTS and strong bisimulation.

2.1.1. DEFINITION. A *labelled transition system* (or *LTS*) is a triple (S, A, \rightarrow) where:

- S is a set of *states*,
- A is a set of *actions*; the *silent action* τ is not in A ; we write $A_\tau = A \cup \{\tau\}$;
- $\rightarrow \subseteq S \times A_\tau \times S$ is the *transition relation*; an element $(r, \alpha, s) \in \rightarrow$ is called a *transition* and is usually written as $r \xrightarrow{\alpha} s$.

We let r, s, \dots range over S ; a, b, \dots over A and α, β over A_τ .

2.1.2. DEFINITION. Let $\mathcal{L} = (S, A, \rightarrow)$ be a LTS. A relation $R \subseteq S \times S$ is called a (*strong*) *bisimulation* if it is symmetric and satisfies: if $r R s$ and $r \xrightarrow{\alpha} r'$, then there exists an s' such that $s \xrightarrow{\alpha} s'$ and $r' R s'$.

Two states r, s are (*strongly*) *bisimilar*, abbreviated $\mathcal{L}: r \Leftrightarrow s$ or $r \Leftrightarrow s$, if there exists a strong bisimulation relating r and s .

It is well known (and easy to see) that the arbitrary union of strong bisimulation relations is again a strong bisimulation; \Leftrightarrow is the maximal strong bisimulation on S . Moreover \Leftrightarrow is an equivalence relation.

2.1.3. *Back and forth bisimulation on states.* Back and forth bisimulations do not only require that related states can simulate each other in a *forward* direction but also also that they can simulate each others behavior in *backward* direction. The definition which comes to mind first is obtained by adding to the definition of a bisimulation relation the condition¹:

if $r R s$ and $r' \xrightarrow{\alpha} r$, then there exists an s' such that $s' \xrightarrow{\alpha} s$ and $r' R s'$.

The resulting equivalence distinguishes the states r and s in Figure 1 below. From r it is possible to do a , then b , and then go back with a , which is not possible from s .

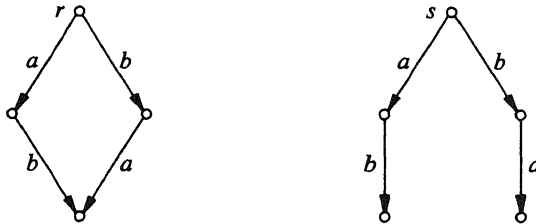


FIGURE 1.

In some non-interleaved models of concurrency ‘diamond’ properties are used to express concurrency of events. In these models the states r and s of Figure 1 are distinguished since the behavior from r corresponds to a pair a and b of concurrent events, whereas s describes a system which either does an a causally followed by a b , or a b causally followed by an a . Back and forth bisimulations cannot capture the intuitions behind these models: if we replace in Figure 1 the labels b by a , then there exists a back and forth bisimulation between r and s , even though

1. This generalization arose from discussion of the first author with Colin Stirling at Edinburgh

from a true concurrency point of view these states are different. At this moment, we do not see how the above back and forth variant of bisimulation equivalence could be useful. Probably everything would work if autoconcurrency is not permitted.

Therefore, in this paper a different type of back and forth bisimulations will be studied: it is possible to move back from a state, but only along the path which represents the *history* that brought one to this state. This means that a bisimulation is no longer a relation between states, but is instead a relation between histories. Below, after some preliminary definitions, we define the new notion of back and forth bisimulation.

2.1.4. DEFINITION. For a set K , the notation K^* denotes the set of finite sequences of elements of K . We denote concatenation of sequences by juxtaposition. With λ we denote the empty sequence and with $|\sigma|$ the length of sequence σ .

2.1.5. DEFINITION. Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. A sequence $(s_0,\alpha_1,s_1)\cdots(s_{n-1},\alpha_n,s_n)\in\rightarrow^*$ is called a *path* from s_0 . A *run* (also called *history* or *computation*) from $s\in S$ is a pair (s,π) , where π is a path from s . We write $run_{\mathcal{Q}}(s)$, or just $run(s)$ for the set of runs from s and $run_{\mathcal{Q}}$ for the set of runs in \mathcal{Q} . We let $\pi,..$ range over paths and $\rho,\sigma,\theta,..$ over runs.

2.1.6. DEFINITION. Let $\rho=(s_0,\pi)\in run(s_0)$ with $\pi=(s_0,\alpha_1,s_1)\cdots(s_{n-1},\alpha_n,s_n)$.

- $first(\rho) = s_0$,
- $path(\rho) = \pi$,
- $last(\rho) = s_n$,
- $states(\rho) = s_0s_1\cdots s_n$,
- $label(\rho) = \alpha_1\cdots\alpha_n$,
- concatenation of runs is denoted by juxtaposition, the result $\sigma\sigma'$ is defined iff $last(\sigma)=first(\sigma')$, the operation is associative and empty runs behave as (left and right) identities,
- $\sigma\stackrel{\alpha}{\rightarrow}\sigma'$ if for some run $\theta=(s,(s,\alpha,s'))$: $\sigma'=\sigma\theta$.

2.1.7. DEFINITION. Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. Two states $r,s\in S$ are *strongly back and forth bisimilar*, abbreviated $\mathcal{Q}:r\leftrightarrow_{bf}s$ or $r\leftrightarrow_{bf}s$, if there exists a symmetric relation $R\subseteq run_{\mathcal{Q}}\times run_{\mathcal{Q}}$, called a *strong back and forth bisimulation*, satisfying

1. $(r,\lambda)R(s,\lambda)$;
2. if $\rho R\sigma$ and $\rho\stackrel{\alpha}{\rightarrow}\rho'$, then there exists a σ' such that $\sigma\stackrel{\alpha}{\rightarrow}\sigma'$ and $\rho'R\sigma'$;
3. if $\rho R\sigma$ and $\rho'\stackrel{\alpha}{\rightarrow}\rho$, then there exists a σ' such that $\sigma'\stackrel{\alpha}{\rightarrow}\sigma$ and $\rho'R\sigma'$.

The following proposition tells us that, when all actions are visible, the possibility to go back 'in ones own history' does not result in any additional distinguishing power: the resulting equivalence is the same as the 'forward only' strong bisimulation equivalence.

2.1.8. PROPOSITION. Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. Then for all $r,s\in S$:

$$\mathcal{Q}:r\leftrightarrow_{bf}s \iff \mathcal{Q}:r\leftrightarrow s.$$

PROOF. " \Leftarrow " Suppose $r\leftrightarrow s$. Let ct be the mapping that associates to each path π in \mathcal{Q} its *colored trace*, i.e. the sequence which is obtained from π by replacing each state by its bisimulation equivalence class (this terminology is borrowed from [9]). So

$$ct((s_0,\alpha_1,s_1)\dots(s_{n-1},\alpha_n,s_n)) = (s_0/\leftrightarrow,\alpha_1,s_1/\leftrightarrow)\dots(s_{n-1}/\leftrightarrow,\alpha_n,s_n/\leftrightarrow).$$

Define relation R by:

$$R = \{(\rho,\sigma),(\sigma,\rho)\mid \rho\in run(r), \sigma\in run(s) \& ct(path(\rho))=ct(path(\sigma))\}.$$

It is straightforward to check that R is a back and forth bisimulation between r and s .

" \Rightarrow " Suppose $r \Leftrightarrow_{bf} s$. Let R be a back and forth bisimulation between r and s . Define

$$R' = \{(last(\rho), last(\sigma)) \mid \rho R \sigma\}.$$

Again, it is straightforward to check that R is a strong bisimulation between r and s . \square

2.2. Weak bisimulation

Weak bisimulation equivalence or observation equivalence is a variant of strong bisimulation equivalence that has been proposed by MILNER [14], to take into account the 'invisible' nature of the silent step τ . In this section we will see that, in contrast to the case of strong bisimulation, weak bisimulation differs from its back and forth variant.

2.2.1. DEFINITION. Let (S, A, \rightarrow) be a LTS. Let $\epsilon \notin A$. Define $\xrightarrow{\epsilon}$ as the transitive and reflexive closure of $\xrightarrow{\tau}$. So $r \xrightarrow{\epsilon} s$ says that there exists a path from r to s consisting of zero or more τ -transitions. Further we define for $a \in A$:

$$r \xrightarrow{a} s \iff \exists r', s' : r \xrightarrow{\epsilon} r' \xrightarrow{a} s' \xrightarrow{\epsilon} s.$$

We let k, l, \dots range over $A_\epsilon = A \cup \{\epsilon\}$.

2.2.2. DEFINITION. Let (S, A, \rightarrow) be a LTS. A relation $R \subseteq S \times S$ is called a *weak bisimulation* if it is symmetric and satisfies: if $r R s$ and $r \xrightarrow{k} r'$, then there exists an s' such that $s \xrightarrow{k} s'$ and $r' R s'$.

Two states r, s are *weakly bisimilar* or *observation equivalent*, abbreviated $@: r \Leftrightarrow_\tau s$ or $r \Leftrightarrow_\tau s$, if there exists a weak bisimulation relating r and s .

Again, it is easy to see that the arbitrary union of weak bisimulation relations is a weak bisimulation; \Leftrightarrow_τ is the maximal weak bisimulation on S . Moreover \Leftrightarrow_τ is an equivalence relation. Since any strong bisimulation is also a weak bisimulation, we have $\Leftrightarrow \subseteq \Leftrightarrow_\tau$.

There is an obvious way to generalize the relations \xrightarrow{k} to runs:

2.2.3. DEFINITION. Let $@=(S, A, \rightarrow)$ be a LTS. Define for $\rho, \sigma \in run_@$:

$$\rho \xrightarrow{\epsilon} \sigma \iff \exists n \geq 0 \exists \rho_0, \dots, \rho_n : \rho = \rho_0, \rho_n = \sigma \ \& \ [\forall 0 \leq i < n : \rho_i \xrightarrow{\tau} \rho_{i+1}].$$

$$\rho \xrightarrow{a} \sigma \iff \exists \rho', \sigma' : \rho \xrightarrow{\epsilon} \rho' \xrightarrow{a} \sigma' \xrightarrow{\epsilon} \sigma.$$

Now consider the following 'weak' variant of the back and forth bisimulation:

2.2.4. DEFINITION. Let $@=(S, A, \rightarrow)$ be a LTS. Two states $r, s \in S$ are *weakly back and forth bisimilar*, abbreviated $@: r \Leftrightarrow_{rbf} s$ or $r \Leftrightarrow_{rbf} s$, if there exists a symmetric relation $R \subseteq run_@ \times run_@$, called a *weak back and forth bisimulation*, satisfying

1. $(r, \lambda) R (s, \lambda)$;
2. if $\rho R \sigma$ and $\rho \xrightarrow{k} \rho'$, then there exists a σ' such that $\sigma \xrightarrow{k} \sigma'$ and $\rho' R \sigma'$;
3. if $\rho R \sigma$ and $\rho' \xrightarrow{k} \rho$, then there exists a σ' such that $\sigma' \xrightarrow{k} \sigma$ and $\rho' R \sigma'$.

Interestingly, weak bisimulation equivalence and weak back and forth bisimulation equivalence are different. In Figures 2 and 3 below, two counterexamples are presented. The states p and q in Figure 2 are not weak back and forth bisimulation equivalent because there exists no weak back and forth bisimulation relating the corresponding empty runs: from q it is possible to do

an a in such a way that always after going back with an a there is a possibility of doing b . This behavior is not possible from p . The counterexample of Figure 3 is similar.

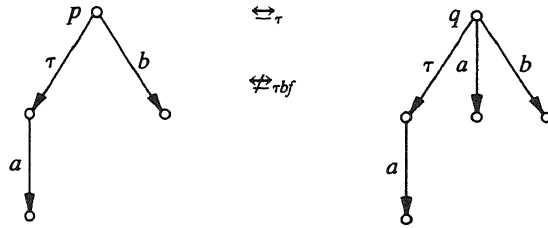


FIGURE 2.

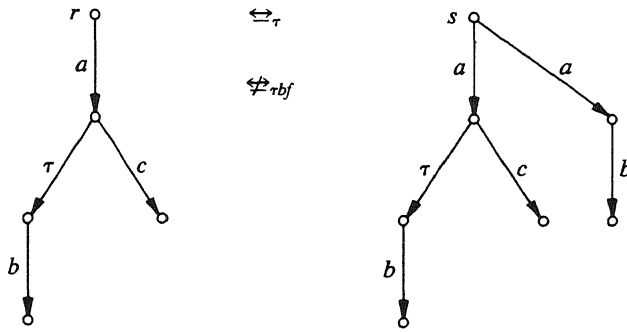


FIGURE 3.

Since any strong back and forth bisimulation is also a weak back and forth bisimulation we have $\Leftrightarrow_{bf} \subseteq \Leftrightarrow_{\tau bf}$ and hence, by Proposition 2.1.8, $\Leftrightarrow \subseteq \Leftrightarrow_{\tau bf}$. The example of Figure 4 shows that this inclusion is strict.

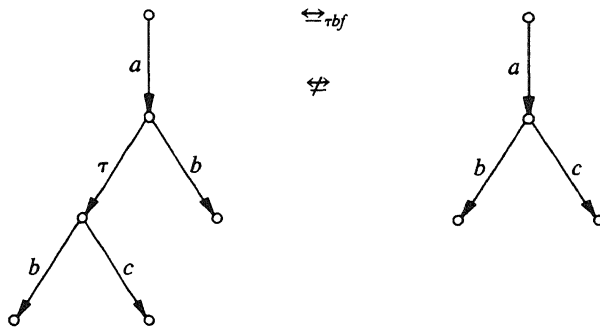


FIGURE 4.

We conclude this section with a technical lemma which will be needed to relate $\Leftrightarrow_{\tau bf}$ with branching bisimulation.

2.2.5. LEMMA. *Let (S, A, \rightarrow) be a LTS and let $r, s \in S$ with $r \Leftrightarrow_{\tau bf} s$. Let $R \subseteq \text{run}_{\mathcal{Q}} \times \text{run}_{\mathcal{Q}}$ be the maximal weak back and forth bisimulation between r and s . Then R has the following X-property:*

$$\forall \rho, \rho' \in \text{run}(r) \forall \sigma, \sigma' \in \text{run}(s) : [\rho \xrightarrow{\epsilon} \rho', \sigma \xrightarrow{\epsilon} \sigma', \rho R \sigma' \ \& \ \rho' R \sigma] \Rightarrow \rho' R \sigma'$$

PROOF. Define relation R' by:

$$R' = R \cup \{(\rho', \sigma'), (\sigma', \rho') \mid \exists \rho \in \text{run}(r) \exists \sigma \in \text{run}(s) : \rho \xrightarrow{\epsilon} \rho', \sigma \xrightarrow{\epsilon} \sigma', \rho R \sigma' \ \& \ \rho' R \sigma\}$$

We prove that R' is a weak back and forth bisimulation. Since R is the maximal back and forth bisimulation and $R \subseteq R'$ by construction, $R = R'$. Thus R has the X-property.

Clearly R' is symmetric. Moreover $(r, \lambda) R' (s, \lambda)$ because $(r, \lambda) R (s, \lambda)$. Suppose $\rho' R' \sigma'$ with $\rho' \in \text{run}(r)$ and $\sigma' \in \text{run}(s)$. If $\rho' R \sigma'$ then the back and forth conditions 2 and 3 are trivially fulfilled. Otherwise there must be a ρ and a σ such that: $\rho \xrightarrow{\epsilon} \rho', \sigma \xrightarrow{\epsilon} \sigma', \rho R \sigma'$ and $\rho' R \sigma$.

We check transfer property 2. Suppose $\rho' \xrightarrow{k} \rho''$. Then $\rho \xrightarrow{k} \rho''$. Since $\rho R \sigma'$, there exists a σ'' such that $\sigma' \xrightarrow{k} \sigma''$ and $\rho'' R' \sigma''$.

Next we check transfer property 3. If $\rho'' \xrightarrow{k} \rho'$, then, since $\rho' R \sigma$, there exists a σ'' such that $\sigma'' \xrightarrow{k} \sigma$ and $\rho'' R' \sigma''$. Now observe that $\sigma'' \xrightarrow{k} \sigma'$.

The remaining case that $\sigma' R' \rho'$ with $\sigma' \in \text{run}(s)$ and $\rho' \in \text{run}(r)$ is symmetric. \square

2.3. Branching bisimulation

In this section we prove the main result of this paper: the back and forth variant of weak bisimulation equivalence coincides with the branching bisimulation of VAN GLABBEK & WEJLAND [9].

2.3.1. DEFINITION. Let $\mathcal{Q} = (S, A, \rightarrow)$ be a LTS. A relation $R \subseteq S \times S$ is called a *branching bisimulation* if it is symmetric and satisfies: if $r R s$ and $r \xrightarrow{\alpha} r'$, then either $\alpha = \tau$ and $r' R s$, or there exist s_1, s' such that $s \xrightarrow{\epsilon} s_1 \xrightarrow{\alpha} s', r R s_1$ and $r' R s'$.

Two states r, s are *branching bisimilar*, abbreviated $\mathcal{Q}: r \Leftrightarrow_b s$ or $r \Leftrightarrow_b s$, if there exists a branching bisimulation relating r and s .

Again the arbitrary union of branching bisimulation relations is a branching bisimulation; \Leftrightarrow_b is the maximal branching bisimulation on S . Moreover \Leftrightarrow_b is an equivalence relation. Since any strong bisimulation is a branching bisimulation and any branching bisimulation is a weak bisimulation, we have $\Leftrightarrow \subseteq \Leftrightarrow_b \subseteq \Leftrightarrow_{\tau}$. It is worth noting that we could have strengthened the above definition by requiring *all* intermediate states in $s \xrightarrow{\epsilon} s_1$ to be related with r . The following lemma implies that this would have led to the same equivalence relation. Moreover, we could have also asked, as in the original definition of [9], that all the states reachable from s' via silent sequences be related with s ; again, by simple considerations, it can be concluded that the equivalence we would obtain would be the same.

2.3.2. LEMMA (cf. Lemma 1.3 of [9]). Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS, let $n>0$ and let $(r_0,\tau,r_1)\dots(r_{n-1},\tau,r_n)$ be a path in \mathcal{Q} with $r_0 \Leftrightarrow_b r_n$. Then for all $0 \leq i \leq n$: $r_0 \Leftrightarrow_b r_i$.

PROOF. Define for $i>0$:

$$R_0 = \Leftrightarrow_b$$

$$R_i = R_{i-1} \cup \{(r,r'), (r',r) \mid \exists r'' : r \xrightarrow{\epsilon} r' \xrightarrow{\tau} r'' \ \& \ r R_{i-1} r''\}$$

$$R_\omega = \bigcup_{i < \omega} R_i$$

First we show that R_ω has the property that we want to prove for \Leftrightarrow_b . Let for some $n>0$, $(r_0,\tau,r_1)\dots(r_{n-1},\tau,r_n)$ be a path with $r_0 R_\omega r_n$. By induction on n we prove that for all $0 \leq i \leq n$: $r_0 R_\omega r_i$.

If $n=1$ the statement is trivially correct.

Now take $n>1$. Since $r_0 R_\omega r_n$, there exists an $m < \omega$ with $r_0 R_m r_n$. By definition of R_{m+1} : $r_0 R_{m+1} r_{n-1}$. Thus $r_0 R_\omega r_{n-1}$ and, by induction hypothesis, $r_0 R_\omega r_i$ for all $0 \leq i \leq n$.

Next we will prove with induction that, for every $n < \omega$, R_n is a branching bisimulation. Thus R_ω is a branching bisimulation and $R_\omega \subseteq \Leftrightarrow_b$. But by construction $\Leftrightarrow_b \subseteq R_\omega$. Hence $\Leftrightarrow_b = R_\omega$ and we have proved the lemma.

R_0 is a branching bisimulation because \Leftrightarrow_b is.

Suppose that, for certain $n>0$, R_{n-1} is a branching bisimulation. We prove that R_n is a branching bisimulation too. By construction R_n is symmetric. Suppose $r R_n r'$ and $r \xrightarrow{\alpha} s$. If $r R_{n-1} r'$, then the transfer property is trivially fulfilled. In the other case there are two possibilities:

1. For some r'' : $r \xrightarrow{\epsilon} r' \xrightarrow{\tau} r''$ and $r R_{n-1} r''$. Using $r R_{n-1} r''$, a first possibility is that $\alpha = \tau$ and $s R_{n-1} r''$. But this means that $r' \xrightarrow{\epsilon} r' \xrightarrow{\tau} r''$ with $r R_n r'$ and $s R_n r''$. Otherwise there are r_1, r_2 such that $r'' \xrightarrow{\epsilon} r_1 \xrightarrow{\alpha} r_2$, $r R_{n-1} r_1$ and $s R_{n-1} r_2$. But then $r' \xrightarrow{\epsilon} r_1 \xrightarrow{\alpha} r_2$, $r R_n r_1$ and $s R_n r_2$.
2. For some r'' : $r' \xrightarrow{\epsilon} r \xrightarrow{\tau} r''$ and $r' R_{n-1} r''$. Then $r' \xrightarrow{\epsilon} r \xrightarrow{\alpha} s$, $r R_n r$ and $s R_n s$. \square

2.3.3. THEOREM. Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. Then for all $r, s \in S$:

$$\mathcal{Q}:r \Leftrightarrow_{\tau b f} s \iff \mathcal{Q}:r \Leftrightarrow_b s.$$

PROOF. " \Leftarrow " Suppose $r \Leftrightarrow_b s$. Let cct be the mapping that associates to each path π in \mathcal{Q} its *concrete colored trace*, i.e. the sequence which is obtained from π by replacing each state by its branching bisimulation equivalence class. So

$$cct((s_0, \alpha_1, s_1) \dots (s_{n-1}, \alpha_n, s_n)) = (s_0 / \equiv_b, \alpha_1, s_1 / \equiv_b) \dots (s_{n-1} / \equiv_b, \alpha_n, s_n / \equiv_b).$$

Let ct be the mapping that associates to each path π in \mathcal{Q} its *(abstract) colored trace*, i.e. the sequence which is obtained from $cct(\pi)$ by removing all elements (C, τ, C) from the sequence. Define relation R by

$$R = \{(\rho, \sigma), (\sigma, \rho) \mid \rho \in run(r), \sigma \in run(s) \ \& \ ct(path(\rho)) = ct(path(\sigma))\}.$$

Using Lemma 2.3.2, it is straightforward to check that R is a weak back and forth bisimulation between r and s .

" \Rightarrow " Suppose $r \Leftrightarrow_{\tau b f} s$. Let $R \subseteq run_{\mathcal{Q}} \times run_{\mathcal{Q}}$ be the maximal weak back and forth bisimulation between r and s . Define

$$R' = \{(last(\rho), last(\sigma)) \mid \rho R \sigma\}.$$

Clearly $r R' s$. We show that R' is a branching bisimulation.

R' is symmetric because R is.

Suppose $r_0 R' s_0$. Then there are ρ, σ with $\rho R \sigma$, $last(\rho)=r_0$ and $last(\sigma)=s_0$. Suppose that $r_0 \xrightarrow{\alpha} r'$. Let $\rho' = \rho(r_0, (r_0, \alpha, r'))$. In the proof of the transfer property we distinguish between two cases.

1. $\alpha \neq \tau$. Since $\rho \xrightarrow{\alpha} \rho'$ and $\rho R \sigma$, there exist $\sigma_1, \sigma_2, \sigma'$ such that $\sigma \xrightarrow{\epsilon} \sigma_1 \xrightarrow{\alpha} \sigma_2 \xrightarrow{\epsilon} \sigma'$ and $\rho' R \sigma'$. Since $\sigma_2 \xrightarrow{\epsilon} \sigma'$, there exists a $\hat{\rho}$ such that $\hat{\rho} \xrightarrow{\epsilon} \rho'$ and $\hat{\rho} R \sigma_2$. But since the last transition of ρ' has label α , $\hat{\rho} = \rho'$ so that $\rho' R \sigma_2$. Because $\sigma_1 \xrightarrow{\alpha} \sigma_2$, there exists a $\bar{\rho}$ such that $\bar{\rho} \xrightarrow{\epsilon} \rho \xrightarrow{\alpha} \rho'$ and $\bar{\rho} R \sigma_1$. Now use that R has the X-property to obtain $\rho R \sigma_1$. But this gives us the transfer property:

$$s_0 \xrightarrow{\epsilon} end(\sigma_1) \xrightarrow{\alpha} end(\sigma_2), \quad r_0 R' end(\sigma_1) \quad \text{and} \quad r' R' end(\sigma_2).$$

2. $\alpha = \tau$. Since $\rho \xrightarrow{\epsilon} \rho'$ and $\rho R \sigma$, there is an $n \geq 0$ and there are σ_i for $0 \leq i \leq n$ such that $\sigma_0 = \sigma$, for $0 < i \leq n$: $\sigma_{i-1} \xrightarrow{\tau} \sigma_i$, and $\rho' R \sigma_n$. If $n = 0$ then $r' R' s_0$ and we have proved the transfer property. If $n > 0$ then we can go back with an ϵ -move from σ_n to σ_{n-1} . A first possibility is that ρ' can simulate this step by doing nothing: $\rho' R \sigma_{n-1}$. If this is the case then either $n = 1$ and we are ready, or we can go back one more ϵ -step from σ_{n-1} to σ_{n-2} . Repeating this, we either find $\rho' R \sigma_0$, in which case we have proved the transfer property for branching bisimulation since $r' R' s_0$, or, for some $m > 0$ with $\rho' R \sigma_m$, we have that a backward step to σ_{m-1} is simulated by a backward step $\hat{\rho} \xrightarrow{\epsilon} \rho \xrightarrow{\tau} \rho'$ with $\hat{\rho} R \sigma_{m-1}$. In this case we use the X-property (Lemma 2.2.5) to obtain $\rho R \sigma_{m-1}$. This gives us the transfer property for branching bisimulation since:

$$s_0 \xrightarrow{\epsilon} end(\sigma_{m-1}) \xrightarrow{\tau} end(\sigma_m), \quad r_0 R' end(\sigma_{m-1}) \quad \text{and} \quad r' R' end(\sigma_m). \quad \square$$

Now we have shown that the back and forth variant of weak bisimulation coincides with branching bisimulation, it becomes natural to consider the back and forth variant of branching bisimulation. Let the symbol \Leftrightarrow_{bbf} denote this equivalence. One can easily prove the following

2.3.4. THEOREM. Let $\mathcal{Q} = (S, A, \rightarrow)$ be a LTS. Then for all $r, s \in S$:

$$\mathcal{Q}: r \Leftrightarrow_{bbf} s \iff \mathcal{Q}: r \Leftrightarrow_b s.$$

PROOF. Similar to but much easier than the proof of Theorem 2.3.3. □

As a corollary of Theorem 2.3.3 and Theorem 2.3.4 one can show that also the back and forth versions of some equivalences in between \Leftrightarrow_{τ} and \Leftrightarrow_b coincide with \Leftrightarrow_b . Consider the following two definitions:

2.3.5. DEFINITION. Let $\mathcal{Q} = (S, A, \rightarrow)$ be a LTS. A relation $R \subseteq S \times S$ is called a η -bisimulation if it is symmetric and satisfies: if $r R s$ and $r \xrightarrow{\alpha} r'$, then either $\alpha = \tau$ and $r' R s$, or there exist s_1, s_2, s' such that $s \xrightarrow{\epsilon} s_1 \xrightarrow{\alpha} s_2 \xrightarrow{\epsilon} s'$, $r R s_1$ and $r' R s'$.

Two states r, s are η -bisimilar, abbreviated $\mathcal{Q}: r \Leftrightarrow_{\eta} s$ or $r \Leftrightarrow_{\eta} s$, if there exists an η -bisimulation relating r and s .

2.3.6. DEFINITION. Let $\mathcal{Q} = (S, A, \rightarrow)$ be a LTS. A relation $R \subseteq S \times S$ is called a *delay bisimulation* if it is symmetric and satisfies: if $r R s$ and $r \xrightarrow{\alpha} r'$, then either $\alpha = \tau$ and $r' R s$, or there exist s_1, s' such that $s \xrightarrow{\epsilon} s_1 \xrightarrow{\alpha} s'$ and $r' R s'$.

Two states r, s are *delay bisimilar*, abbreviated $\mathcal{Q}: r \Leftrightarrow_d s$ or $r \Leftrightarrow_d s$, if there exists a delay bisimulation relating r and s .

The notion of η -bisimulation was first introduced by BAETEN & VAN GLABBEK [3] as a finer

version of observation equivalence. Delay bisimulations are used by WEIJLAND [21]. A variant of delay bisimulation - only differing in the treatment of divergence - first appeared in MILNER [13], also under the name observational equivalence. From the definitions it follows right away that $\Leftrightarrow_b \subseteq \Leftrightarrow_\eta$, $\Leftrightarrow_b \subseteq \Leftrightarrow_d$, $\Leftrightarrow_\eta \subseteq \Leftrightarrow_\tau$ and $\Leftrightarrow_d \subseteq \Leftrightarrow_\tau$. The example of Figure 2 can be used to show that the second and third inclusion are strict. The example of Figure 3 illustrates the strictness of the other two inclusions.

2.3.7. COROLLARY. *Let $\Leftrightarrow_{\eta bf}$ and \Leftrightarrow_{abf} denote the back and forth variants of η - and delay bisimulation respectively. Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. Then for all $r,s \in S$:*

$$\mathcal{Q}:r \Leftrightarrow_{\eta bf} s \iff \mathcal{Q}:r \Leftrightarrow_b s \iff \mathcal{Q}:r \Leftrightarrow_{abf} s.$$

PROOF. Suppose $r \Leftrightarrow_{\eta bf} s$. Then there exists a back and forth η -bisimulation relating r and s . But any back and forth η -bisimulation is also a weak back and forth bisimulation. Therefore $r \Leftrightarrow_{abf} s$. But this implies $r \Leftrightarrow_b s$ by Theorem 2.3.3.

Suppose $r \Leftrightarrow_b s$. Then $r \Leftrightarrow_{bbf} s$ by Theorem 2.3.4. So there exists a back and forth branching bisimulation between r and s . But any back and forth branching bisimulation is also a back and forth η -bisimulation. Therefore $r \Leftrightarrow_{\eta bf} s$.

The remaining two implications can be proved in the same way. \square

3. ALGEBRAIC CHARACTERIZATION

In this section we show that observation equivalence can be characterized in terms of abstraction homomorphisms that preserve successors, whereas branching bisimulation corresponds with abstraction homomorphisms which preserve both successors and predecessors.

3.1. DEFINITION. A *category of labelled computations* (CLAC) $\mathcal{C} = (S, C, src, trg, ;, id, A, o)$ is defined as follows:

- $(S, C, src, trg, ;, id)$ is a category, i.e. S is a set of objects called *states*; C is a set of arrows called *computations*; $src, trg : C \rightarrow S$ are functions associating to every computation its *source* and *target*; the binary operation $;$: $C \times C \rightarrow C$ of *concatenation* is partial: $\rho = \rho'; \rho''$ is defined iff $trg(\rho') = src(\rho'')$, with $src(\rho) = src(\rho')$ and $trg(\rho) = trg(\rho'')$; concatenation has an identity $id(s)$ (both left and right) for each state s and is associative;
- A is a set of *actions*; the silent action τ is not in A ;
- $o : C \rightarrow A^*$ is a *labeling* function which respects concatenation, i.e. $o(\rho; \rho') = o(\rho) o(\rho')$.

On computations we define a prefix preorder: we have $\rho_1 \leq \rho_2$ iff there is a computation σ with $\rho_2 = \rho_1; \sigma$. We let $succ(\rho) = \{\sigma \mid \rho \leq \sigma\}$ and $pred(\rho) = \{\sigma \mid \sigma \leq \rho\}$.

3.2. DEFINITION. Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. Its *associated* CLAC is $\mathcal{C}(\mathcal{Q}) = (S, C, src, trg, ;, id, A, o)$ where C is the set $run_{\mathcal{Q}}$ of runs of \mathcal{Q} , $src(\rho) = first(\rho)$, $trg(\rho) = last(\rho)$, operation $;$ is run concatenation, $id(s)$ is the empty run from s , and $o(\rho)$ is obtained from $label(\rho)$ by removing all τ 's.

3.3. DEFINITION. Let $\mathcal{C} = (S, C, src, trg, ;, id, A, o)$ and $\mathcal{C}' = (S', C', src', trg', ;', id', A', o')$ be CLAC's. A pair of *surjective* functions $k = \langle f, g \rangle$, $f : C \rightarrow C'$ and $g : S \rightarrow S'$ is a *forward abstraction homomorphism*, and we write $\mathcal{C} \xrightarrow{k}_{fw} \mathcal{C}'$, iff:

- i) $g(src(\rho)) = src'(f(\rho))$ and $g(trg(\rho)) = trg'(f(\rho))$,
- ii) $f(\rho; \sigma) = f(\rho);' f(\sigma)$ and $f(id(s)) = id'(g(s))$,
- iii) $o(\rho) = o'(f(\rho))$ and
- iv) $f(succ(\rho)) = succ'(f(\rho))$, where f is extended to sets.

In words, a forward abstraction homomorphism must respect sources, targets, concatenations, identities, observation and successors. A pair $k = \langle f, g \rangle$ is called a *back and forth abstraction homomorphism*, and we write $\mathcal{C} \xrightarrow{k}_{bf} \mathcal{C}'$, iff, besides (i)-(iv), it satisfies also:

- v) $f(pred(\rho)) = pred'(f(\rho))$,

i.e. if it respects predecessors.

3.4. DEFINITION. Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. A pair of states $r,s \in S$ is *forward preserving*, and we write $r \leftrightarrow_{fp} s$, iff there exist \mathcal{C} and $k = \langle f, g \rangle$ with $\mathcal{Q}(\mathcal{C}) \xrightarrow{k}_{fw} \mathcal{C}$ and $g(r) = g(s)$. Similarly, a pair of states $r,s \in S$ is *back and forth preserving*, and we write $r \leftrightarrow_{bfp} s$, iff there exist \mathcal{C} and $k = \langle f, g \rangle$ with $\mathcal{Q}(\mathcal{C}) \xrightarrow{k}_{bf} \mathcal{C}$ and $g(r) = g(s)$.

The following theorem says that forward and back and forth abstraction homomorphisms correspond to weak observation equivalence and branching bisimulation respectively.

3.5. THEOREM. Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. Then, given $r,s \in S$, we have

- a) $r \leftrightarrow_{fp} s$ iff $r \leftrightarrow_r s$;
- b) $r \leftrightarrow_{bfp} s$ iff $r \leftrightarrow_{rbf} s$.

PROOF. We sketch the proof of (a). The proof of (b) is similar. Define a one-to-one correspondence between weak bisimulation equivalences and forward abstraction homomorphisms. When going from homomorphisms $\langle f, g \rangle$ to bisimulations \sim , the construction is straightforward, assuming $s \sim s'$ iff $g(s) = g(s')$. In the other direction it is convenient first to extend \sim to computations letting $\rho \sim \sigma$ iff $src(\rho) \sim src(\sigma)$, $trg(\rho) = trg(\sigma)$ and $o(\rho) = o(\sigma)$, and then to take $f(\rho) = \rho / \sim$ and $g(s) = s / \sim$.

The result follows since if a pair belongs to a bisimulation, it also belongs to a bisimulation equivalence. \square

3.6. REMARK. CLAC's and abstraction homomorphisms have been introduced here for a labeling function o ranging over A^* . However, exactly the same definition (for both the forward and the back and forth case) could be given in terms of a different observation function $o_{\rho o}$, yielding for instance a partial ordering observation $o_{\rho o}(\rho)$ of a computation ρ . This allows us to develop an algebraic theory of observation equivalence, in both its traditional version and its branching variant, also for true concurrency.

4. LOGICAL CHARACTERIZATION

Theorem 2.3.3 suggests a logical characterization of branching bisimulation which is a variant of the Hennessy-Milner logic [10]. The characterization relies on a well-known theorem from HENNESSY & MILNER [10], which we recall first.

4.1. DEFINITION. Let A be a given alphabet of symbols. The set $HML(A)$, often abbreviated to HML , of *Hennessy-Milner logic formulas over A* is given by the following grammar:

$$\phi ::= T \mid \phi \wedge \psi \mid \neg \phi \mid \langle \alpha \rangle \phi.$$

Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. The *satisfaction relation* $\models \subseteq S \times HML$ is the least relation such that:

- $s \models T$ for all $s \in S$,
- $s \models \phi \wedge \psi$ iff $s \models \phi$ and $s \models \psi$,
- $s \models \neg \phi$ iff not $s \models \phi$,
- $s \models \langle \alpha \rangle \phi$ iff for some $t \in S$: $s \xrightarrow{\alpha} t$ and $t \models \phi$.

The following notations are standard:

- F stands for $\neg T$,
- $\phi \vee \psi$ stands for $\neg(\neg \phi \wedge \neg \psi)$,
- $[\alpha] \phi$ stands for $\neg \langle \alpha \rangle \neg \phi$.

Let \mathcal{L} be a set of logical formulas with an associated satisfaction relation. Two states $r,s \in S$ are \mathcal{L} -*equivalent*, abbreviated $\mathcal{Q}: r \sim_{\mathcal{L}} s$ or $r \sim_{\mathcal{L}} s$, if for all formulas ϕ in \mathcal{L} : $s \models \phi \Leftrightarrow t \models \phi$.

Transition system \mathcal{Q} is called *image finite* if for all $s \in S$ and $a \in A$ the set $\{t \mid s \xrightarrow{a} t\}$ is finite.

4.2. THEOREM ([10]). Let $\mathcal{Q}=(S,A,\rightarrow)$ be an image finite LTS. Then for all $r,s \in S$:

$$r \Leftrightarrow s \Leftrightarrow r \sim_{HML} s.$$

4.3. DEFINITION. Let A be a given alphabet and let k range over A_ϵ . The set $BFL(A)$ (or just BFL) of back and forth logic formulas over A is defined by the following grammar:

$$\phi ::= T \mid \phi \wedge \psi \mid \neg \phi \mid \langle k \rangle \phi \mid \langle \leftarrow k \rangle \phi.$$

Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS. The satisfaction relation $\vDash \subseteq run_{\mathcal{Q}} \times BFL$ is the least relation such that:

- $\rho \vDash T$ for all $\rho \in run_{\mathcal{Q}}$,
- $\rho \vDash \phi \wedge \psi$ iff $\rho \vDash \phi$ and $\rho \vDash \psi$,
- $\rho \vDash \neg \phi$ iff not $\rho \vDash \phi$,
- $\rho \vDash \langle k \rangle \phi$ iff for some run ρ' : $\rho \xrightarrow{k} \rho'$ and $\rho' \vDash \phi$,
- $\rho \vDash \langle \leftarrow k \rangle \phi$ iff for some run ρ' : $\rho' \xrightarrow{k} \rho$ and $\rho' \vDash \phi$.

The satisfaction relation $\vDash \subseteq S \times BFL$ is defined by: $s \vDash \phi$ iff $(s,\lambda) \vDash \phi$.

4.4. THEOREM. Let $\mathcal{Q}=(S,A,\rightarrow)$ be a LTS with an image finite double arrow relation (i.e. for all $r \in S$ and $k \in A_\epsilon$ the set $\{s \mid r \xrightarrow{k} s\}$ is finite). Then for all $r,s \in S$:

$$r \Leftrightarrow_b s \Leftrightarrow r \sim_{BFL} s.$$

PROOF. Let $r,s \in S$ with

$$\mathcal{Q}: r \Leftrightarrow_b s. \tag{1}$$

By Theorem 2.3.3, this is equivalent to:

$$\mathcal{Q}: r \Leftrightarrow_{\tau bf} s. \tag{2}$$

Consider the LTS $bf(\mathcal{Q})$, which is obtained by replacing the single step transitions between states in S with the corresponding many step forward and backward arrows between paths in \mathcal{Q} . More precisely, we define:

$$bf(\mathcal{Q}) = (run_{\mathcal{Q}}, A_{bf}, \rightarrow_{bf}),$$

where:

$$A_{bf} = A_\epsilon \cup \{-k \mid k \in A_\epsilon\}$$

and for $\rho, \rho' \in run_{\mathcal{Q}}$ and $k \in A_\epsilon$:

$$\rho \xrightarrow{-k}_{bf} \rho' \Leftrightarrow \rho \xrightarrow{k} \rho', \text{ and}$$

$$\rho \xrightarrow{k}_{bf} \rho' \Leftrightarrow \rho' \xrightarrow{-k} \rho.$$

Observe that any weak back and forth bisimulation on \mathcal{Q} is a strong bisimulation on $bf(\mathcal{Q})$ and vice versa. Thus, (2) is equivalent to:

$$bf(\mathcal{Q}): (r,\lambda) \Leftrightarrow (s,\lambda). \tag{3}$$

Because the double arrow relation in \mathcal{Q} is image finite and because we only consider finite runs, it easily follows that $bf(\mathcal{Q})$ is image finite. This means that we can use Theorem 4.2 to obtain that (3) is equivalent to:

$$bf(\mathcal{Q}): (r,\lambda) \sim_{HML(A_{bf})} (s,\lambda). \tag{4}$$

Let f be the bijective mapping that associates to each Hennessy-Milner formula in $HML(A_{bf})$ a

back and forth formula in $BFL(A)$ by replacing each " $-$ " by a " \leftarrow ". Let $\rho \in \text{run}_{\mathcal{Q}}$ and $\phi \in HML(A_{bf})$. A simple inductive argument gives that $\rho \vDash \phi$ relative to $bf(\mathcal{Q})$ iff $\rho \vDash f(\phi)$ relative to \mathcal{Q} . Therefore we may conclude that (4) is equivalent to:

$$\mathcal{Q}: r \sim_{BFL(A)} s. \quad (5)$$

The theorem now follows from the equivalence of (1)-(5). \square

4.5. EXAMPLE. Let p, q be as in Figure 2, and r, s as in Figure 3. Let $[k] = \neg \langle k \rangle \neg$ and $[\leftarrow k] = \neg \langle \leftarrow k \rangle \neg$.

If $\phi = \langle a \rangle [\leftarrow a] \langle b \rangle T$ then $q \vDash \phi$ while $p \not\vDash \phi$.

If $\phi' = [a][b] \langle \leftarrow b \rangle \langle c \rangle T$ then $r \vDash \phi'$ while $s \not\vDash \phi'$.

REFERENCES

- [1] P. ACZEL (1988): *Non-well-founded sets*, CSLI Lecture Notes No.14, Stanford University.
- [2] A. ARNOLD & A. DICKY (1989): *An algebraic characterization of transition system equivalences*. Information and Computation 82, pp. 198-229.
- [3] J.C.M. BAETEN & R.J. VAN GLABBEEK (1987): *Another look at abstraction in process algebra*. In: Proceedings ICALP 87, Karlsruhe (Th. Ottman, ed.), LNCS 267, Springer-Verlag, pp. 84-94.
- [4] M.C. BROWNE, E.M. CLARKE & O. GRUMBERG (1988): *Characterizing finite Kripke structures in propositional temporal logic*. Theoretical Computer Science 59(1,2), pp. 115-131.
- [5] I. CASTELLANI (1987): *Bisimulations and abstraction homomorphisms*. Journal of Computer and System Sciences 34, pp. 210-235.
- [6] I. CASTELLANI, P. FRANCESCHI & U. MONTANARI (1983): *Labeled event structures: a model for observable concurrency*. In: Proceedings IFIP TC2 Working Conference on Formal Description of Programming Concepts - II, Garmisch (D. Bjørner, ed.), North-Holland, pp. 383-400.
- [7] R. DE NICOLA & F.W. VAANDRAGER (1990): *Three logics for branching bisimulation (extended abstract)*. In: Proceedings 5th Annual Symposium on Logic in Computer Science (LICS 90), Philadelphia, USA, IEEE Computer Society Press, Los Alamitos, CA, pp. 118-129, full version to appear as CWI Report CS-R9012.
- [8] E.A. EMERSON & J.Y. HALPERN (1986): *'Sometimes' and 'Not Never' revisited: on branching time versus linear time temporal logic*. JACM 33(1), pp. 151-178.
- [9] R.J. VAN GLABBEEK & W.P. WEIJLAND (1989): *Branching time and abstraction in bisimulation semantics (extended abstract)*. In: Information Processing 89 (G.X. Ritter, ed.), Elsevier Science Publishers B.V. (North Holland), pp. 613-618.
- [10] M. HENNESSY & R. MILNER (1985): *Algebraic laws for nondeterminism and concurrency*. JACM 32(1), pp. 137-161.
- [11] M. HENNESSY & C. STIRLING (1985): *The power of the future perfect in program logics*. Information and Control 67, pp. 23-52.
- [12] R. MILNER (1980): *A Calculus of Communicating Systems*, LNCS 92, Springer-Verlag.
- [13] R. MILNER (1981): *Modal characterisation of observable machine behaviour*. In: Proceedings CAAP 81 (G. Astesiano & C. Bohm, eds.), LNCS 112, Springer-Verlag, pp. 25-34.
- [14] R. MILNER (1983): *Calculi for synchrony and asynchrony*. Theoretical Computer Science 25, pp. 267-310.
- [15] R. MILNER (1989): *Communication and concurrency*, Prentice-Hall International.
- [16] U. MONTANARI & M. SGAMMA (1989): *Canonical representatives for observational equivalence classes*. In: Resolution Of Equations In Algebraic Structures, Vol. I, Algebraic Techniques (H. Ait-Kaci & M. Nivat, eds.), Academic Press, pp. 293-319.
- [17] R. PAIGE & R. TARJAN (1987): *Three partition refinement algorithms*. SIAM Journal on Computing 16(6), pp. 973-989.
- [18] D.M.R. PARK (1981): *Concurrency and automata on infinite sequences*. In: Proceedings 5th

- GI Conference (P. Deussen, ed.), LNCS 104, Springer-Verlag, pp. 167-183.
- [19] J. SIFAKIS (1984): *Property-preserving homomorphisms of transition systems*. In: Proceedings Logics of Programs, 1983 (E. Clarke & D. Kozen, eds.), LNCS 164, Springer-Verlag, pp. 458-473.
- [20] C. STIRLING (1990): *Modal and temporal logics*. In: Handbook of Logic in Computer Science, Vol I (S. Abramsky, ed.), to appear.
- [21] W.P. WEIJLAND (1989): *Synchrony and asynchrony in process algebra*. Ph.D. Thesis, University of Amsterdam.