Consiglio Nazionale delle Ricerche

ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE

PISA

BEHAVIOURAL EQUIVALENCES FOR TRANSITION SYSTEMS

R. De Nicola

Nota interna B84-13

Settembre 1984

BEHAVIOURAL EQUIVALENCES
FOR TRANSITION SYSTEMS

by

Rocco De Nicola*

Istituto di Elaborazione della Informazione
Consiglio Nazionale delle Ricerche - Pisa (Italy)

and

Department of Computer Science
University of Edinburgh - Edinburgh (U.K.)

* Author's address: I.E.I. - Via S. Maria, 46 - I-56100 Pisa (ITALY)

ABSTRACT

In general one may try to use the same formalism to describe what is required of a system (its specification) and how it can be built from smaller components (its implementation), then the theory of systems equivalences can be very helpful to prove that a particular implementation satisfies a given specification. The kind of equivalence one is interested in depends very heavily on the particular behavioural aspects one is willing to capture. The choice is particularly debated in the case of parallel systems due to the large number of properties which may be relevant for their analysis.

In this paper we discuss an compare various proposed theories of equivalence for parallel or nondeterministic systems, by adapting them to a common model which underlies many proposed models of parallelism: labelled transition systems. The stress is over operational significance of the various equivalences and over the properties they preserve.

CONTENTS

§0. Introduction

The reasons one wants to use formal models to describe systems is to be able to analyze them, prove their properties and discuss their design; to this end the notions of simulation, equivalence and approximation between systems are of fundamental importance.

In general one may try to use the same formalism to describe what is required of a system (its specification) and how it can be built from smaller components (its implementation) and use the theories of simulation or approximations to prove that a particular implementation is correct with respect to a given specification. Another possible use of the theory of equivalence or simulation is to arbitrarily interchange subsystems proved equivalent. That is one subsystem may replace the other as part of a larger system without affecting the external behaviour of the overall system. This turns out to be very helpful to support a stepwise development method.

Anyway the equivalence approaches to systems development and verification can be very dangerous since they could lead to erroneous conclusions and every reduction or equivalence notion needs to be studied together with the properties it preserves, i.e. together with the kind of conclusions arrived at about the reduced systems which are also valid for the original systems.

Roughly speaking we say that a system S_1 simulates (is equivalent to) a system S_2 whenever "some" aspects of their behaviour are compatible. The kind of equivalences, or simulations one is interested in depends very heavily on the particular behavioural aspects one is willing to capture. This choice is particularly debated in the case of parallel systems due to the large number of properties which may be relevant for their analysis.

There have been very different proposals in the literature, as mentioned above, the different choices depend on the use one as in mind for the systems he is modelling. The main idea is to consider equivalent two systems when no external observations can distinguish them. But there is still disagreement on what are reasonable observations and on how they can be used to distinguish systems. The major proposals have been made for various CCS-like languages but they can be extended easily to other formalisms. In the sequel we will present and discuss some of these proposals, by adapting them to labelled transitions systems, /Kel76, Plo81/, trying to stress the aspects of systems they ignore and the identifications they force. Moreover by defining all these equivalences over a unique model we will be able to study their interrelations. A similar comparison has been attempted in /BR83/ but there a different class of equivalences is considered and the stress was on their logical implications more then on their operational significance.

§1. Labelled Transition Systems

Since their appereance Keller's transition systems /Kel76/ have proved to be a model which in some sense underlies many proposed models of parallelism. We will present a particular class of transitions systems with a particular emphasis on the methodologies for formal verification it supports. In particular we will discuss methods for reducing systems to simpler ones and for proving equivalence of two given systems.

Transitions Systems are an abstract relational model based on two primitive notions, namely those of state and transition. Given any other model for which it is possible to define a notion of global state and a notion of indivisible action causing a state transition we can define for each object of the model a corresponding transition system. This correspondence determines an "interleaving" semantics for the model and any property of systems which is preserved under interleaving may be studied purely in terms of transition systems. In this way the latters constitute a significant part of most models for parallelism and many of their properties can be formulated in this highly abstract conceptual model.

We will consider a particular class of nondeterministic transition systems which can be used to model systems controllable through interactions with a surrounding environment, but also capable of making internal or hidden moves which cannot be influenced or even seen by any outside agent. This model is named labelled transition systems (LTS) and is a slight modification of the model used by Keller.

Definition 1.1

A labelled transition system is a quadruple $(Q, A, \xrightarrow{\mu}, q_0)$ where Q is a countable set of states, A is a countable set of elementary actions (a, b, c...), $\xrightarrow{\mu}$ with $\mu \in A \cup \{\tau\}$ is a set of binary relations on Q, and q_0 Q is the initial state. □

In this definition each of the relations $\xrightarrow{a}$ describes the effect of the execution of the elementary action a and $q \xrightarrow{a} q'$ indicates that by performing the action a the system when in the state q can reach the state q'. After Milner /Mil80/ the special simbol τ is used to denote the internal actions and $q \xrightarrow{\tau} q'$ indicates that a system in the state q can perform a silent move to the state q'.

As noted in /BrR83/ a transition system can be "unrolled" into a tree in the usual way. The initial state is the root and the transition relation is represented by the arcs labelled with elements from $A \cup \{\tau\}$, the various nodes will identify the other states. Most of the examples will be done in terms of this, more intuitive, model.

Very often it is necessary to abstract from internal actions and also to consider sequences of actions. We need to introduce notation to represent such possibilities. In the sequal A^* will be used to denote strings over the set of actions A and will be ranged over by $s, t, \ldots$, and ε will be used to denote the empty string. A sequence of actions can be represented in a single transition by $p \xrightarrow{\mu_0 \mu_1 \ldots \mu_n} q$, which means that there is some sequence of states p_i such that p_0 is p, p_n is q and for $i \in \{0, \ldots, n-1\}$ $p_i \xrightarrow{\mu_i} p_{i+1}$. To allow absorption of τ-actions $p \xrightarrow{\tau^n} q$, $n \geq 0$, may be abbreviated to $p \Longrightarrow q$, and $p \xrightarrow{\tau^m a \tau^n} q$ to $p \overset{a}{\Longrightarrow} q$. We will also use $p \longrightarrow q$ to denote $p \xrightarrow{\tau} q$.

For strings $s \in A^*$, $p \overset{s}{\Longrightarrow} q$ is defined in the obvious way so that if $s = a_0 \ldots a_{n-1}$ then $p \overset{a_0}{\Longrightarrow} p_1 \ldots \overset{a_{n-1}}{\Longrightarrow} q$, $p \overset{\varepsilon}{\Longrightarrow} q$ will be rendered as $p \Longrightarrow q$. Moreover $p \overset{s}{\Longrightarrow}$ and $p \xrightarrow{\mu}$ will abbreviate there exists $p' \in P$ such that $p \overset{s}{\Longrightarrow} p'$ and $p \xrightarrow{\mu} p'$ respectively; not exists $p' \in P$ such that $p \overset{s}{\Longrightarrow} p'$ ($p \xrightarrow{\mu} p'$) will be rendered as $p \overset{s}{\not\Longrightarrow}$ ($p \overset{\mu}{\not\longrightarrow}$). In the sequel it will be useful also to be able to talk about the immediate moves of T when in a particular state and of the possibility for a system to perform an infinite number of internal moves without ever performing a visible action. These and others interesting properties of transition systems are captured by the following definitions.

<u>Definition 1.2</u>

If $T = (Q, A, \xrightarrow{\mu}, q_0)$ is an LTS and $q \in Q$ we have

i. $\mathrm{Init}(q) = \{ a \in A \mid q \overset{a}{\Longrightarrow} \}$

ii. $\mathrm{Traces}(q) = \{ s \in A^* \mid q \overset{s}{\Longrightarrow} \}$ □

<u>Definition 1.3</u>

If p is any state of a transition system T then we say

i. $p\Downarrow$ (read p converges) if either $p \not\longrightarrow$ or $p \longrightarrow p'$ implies $p'\Downarrow$.

ii. $p\Uparrow$ (read p diverges) if not $p\Downarrow$. □

<u>Definition 1.4</u>

A relation R over a transition system T is <u>image finite</u> if for each state q of T $\{ q' \mid q \, R \, q' \}$ is finite. □

The induced transition systems whose binary relation is $\overset{s}{\Longrightarrow}$ are particularly interesting since they allow us to analyse the aspects of systems which can be inferred by considering only their externally visible actions. Many of the proposed approaches to systems simulation or systems reduction are based or can be reduced to ignoring particular actions or particular sets of actions which for one reason or another have to be (or can be) considered internal. In fact all the equivalences we will discuss take this possibility into account. For simplicity reasons we will consider only transition systems such that $-\mu\rightarrow$ is <u>image finite</u>. However note that the induced relation $=s\Rightarrow$ is not necessarily image finite. Indeed any divergent state p is such that $\{p' \mid p \Longrightarrow p'\}$ is infinite.

The rest of the paper will be dedicated to define and discuss various behavioural equivalences over labelled transition systems.

§2. Strings Equivalence

A natural proposal for systems equivalence is to consider equivalent two systems which can perform exactly the same sequences of <u>visible</u> actions /Hoa81/. In this way we can abstract from the internal (invisible) actions of a system.

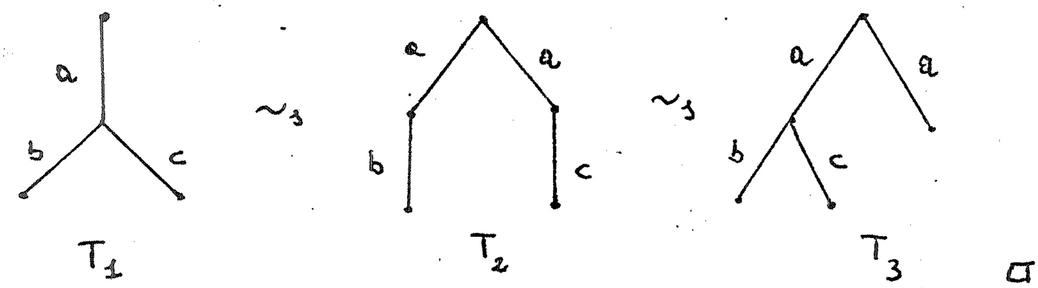
<u>Definition 2.1</u>

If $T_1 = (P, A, -\mu\rightarrow, p_o)$ and $T_2 = (Q, A, -\mu\rightarrow, q_o)$ are two transition systems then we have

$T_1 \sim_s T_2$ iff for all $s \in A^*$ $q_o{=}s{\Rightarrow}q$ if and only if $p_o{=}s{\Rightarrow}p$) □

It is easy to prove that $T_1 \sim_s T_2$ iff $\text{Traces}(q_o) = \text{Traces}(p_o)$.

<u>Example</u>



It can be easily proved that $\sim_s$ is an equivalence relation, it is indeed the equivalence used in automata and formal languages theory over the years and it is the basis of many semantics for CSP /Hoa81, HBR81, OH83/. However when considering systems not running in isolation but exchanging informations or synchronizing with other systems it is important to know whether some communications will always take place or there is the possibility of deadlock.

If we want to be able to model and distinguish such situations we have that, despite its simplicity, string equivalence is not a useful notion. In fact we have that if we try to exchange the sequence of messages "ab" with the system T_1 we will be always successfull while this will not be the case when we consider T_2 or T_3, moreover T_2 and T_3 might exhibit very different reactions as well, in the sense that while the former after the acceptance of the message "a" would always accept the pair of messages "b,c" the latter might not.

§3. Observational Equivalences

There are various ways to "improve" string equivalence in order to be able to differentiate transitions systems similar to the ones of the previous example. The additional feature a new equivalence needs to have is to be able to take into account not only the sequences of actions a system may perform but also "some" of the intermediate states the system goes through while performing a particular sequence of actions. In fact differing intermediate states can be exploited in different contexts to produce different overall behaviours.

The first proposal in this direction has been made by R. Milner. In /Mil80/ and in previous related works /HM80/, a so called observational equivalence is defined for a Calculus of Communicating Systems (CCS). Though it has been proposed for a particular class of transition systems it can be easily extended to the labelled transition systems we are considering. Observational equivalence ($\approx$) is defined as the intersection of a decreasing sequence of equivalences $\approx_k$ $(k \geq 0)$ where $\approx_0$ is the universal relation and for each k the equivalence $\approx_k$ is defined in terms of its predecessor $\approx_{k-1}$.

Definition 3.1

If $T = (Q, A, \xrightarrow{\mu}, q_0)$ is a transition system and $p, q \in Q$ then

1. $p \approx_0 q$ is always true
2. $p \approx_k q$ iff for all $s \in A^*$
   - i. $\exists p'. \ p \overset{s}{\Longrightarrow} p'$ implies $\exists q'. \ q \overset{s}{\Longrightarrow} q'$ and $p' \approx_{k-1} q'$
   - ii. $\exists q'. \ q \overset{s}{=} q'$ implies $\exists p', \ p \overset{s}{\Longrightarrow} q'$ and $p' \approx_{k-1} q'$
3. $p \approx q$ iff for all $k \geq 0$ $p \approx_k q$ $\square$

This relation between states of a particular transition systems can be easily extended to a relation between transition systems.

Definition 3.2

If $T = (P \cup Q, A, \xrightarrow{\mu}_1 \cup \xrightarrow{\mu}_2, \tau_0)$ is the transition system obtained from the union of $T_1 = (P, A, \xrightarrow{\mu}_1, p_0)$ and $T_2 = (Q, A, \xrightarrow{\mu}_2, q_0)$ we have that

i. $T_1 \approx_k T_2$ if and only if $p_0 \approx_k q_0$ and

ii. $T_1 \approx T_2$ if and only if $p_0 \approx q_0$

$\square$

An alternative way of defining observational equivalence has been suggested by D. Park which has been inspired by the homorphisms of automata theory (e.g. see the notion of <u>weak</u> homorphism in /Ginz68/). These alternative definition has been used and discussed in /Mil84/.

<u>Definition 3.3</u>

If T_1 and T_2 are two transition systems as in the previous definition then we say that T_1 <u>simulates</u> T_2 via $R \subseteq P \times Q$ if

i. $(p_o, q_o) \in R$

ii. $(p, q) \in R$ and $p \overset{s}{\Longrightarrow} p'$ implies $q \overset{s}{\Longrightarrow} q'$ and $(p', q') \in R$ □

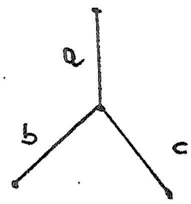
<u>Definition 3.4</u>

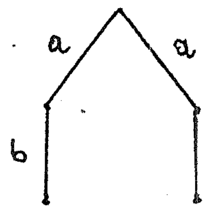
If T_1 and T_2 are as in the previous definition we say T_1 <u>bisimulates</u> T_2 via R ($T_1 \approx_R T_2$) if there exists a relation R such that T_1 <u>simulates</u> T_2 via R and T_2 <u>simulates</u> T_1 via R. □

The two alternative ways of characterizing observational equivalence are discussed extensively in /San82/ and /HM83/; in particular they show that $\approx$ and bisimulates ($\approx_R$) do not coincide when considering infinite systems. Indeed we have $T_1 \approx_R T_2$ implies $T_1 \approx T_2$ but not viceversa. They are the same only for <u>image finite</u> transitions systems.

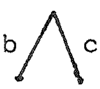
In the sequel we give some examples of transition systems (represented as trees) which either are both observation equivalent and bisimilar (the relation R will be evident) or are both neither observation equivalent nor bisimilar, we will use $\approx$ and $\not\approx$ to express this.

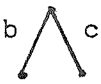
<u>Examples</u>

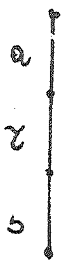
T_1 = 

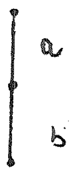
$\not\approx$


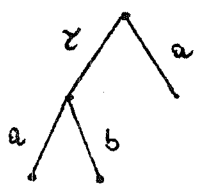
= T_2

Since T_1 =a=>  b c and T_2 =a=>  b or T_2 =a=>  c

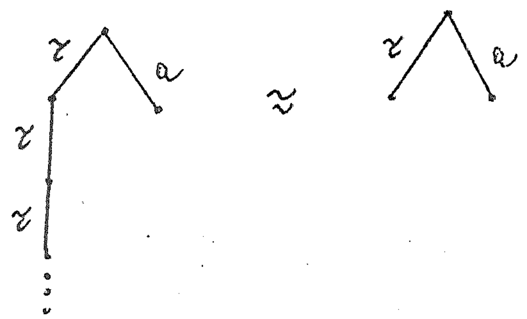
and  b $\not\approx_1$  b c $\not\approx_1$  c



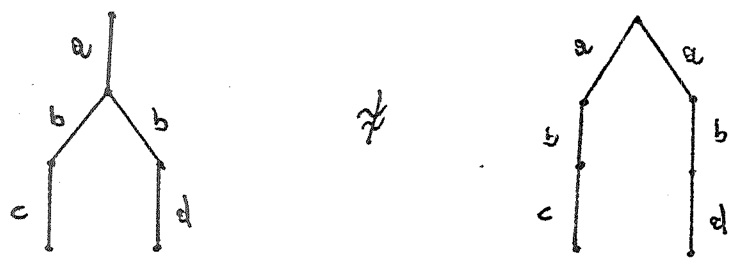
$\approx$




$\approx$

The last identification shows that both observation equivalence and bisimulation have problems in coping with infinite transition systems, in particular with transition systems with an infinite number of internal moves, in the sense that they consider equivalent systems which computationally are very different. We have in fact that a system which can either compute for ever or perform the action "a" and then stop is considered equivalent to a system which can either perform a silent move and then stop or perform the action "a" and then stop. On the other hand we have:



Yet it seems intuitively clear that there is no way to distinguish this two processes by only considering the visible actions they may perform and the way they may react to external experiments. So it seems that the notion of observational equivalence may be discriminating too much from some points of view.

§4. Weak Equivalence

The main reason for $\approx$ being finer (more discriminating) than one would like and expect seems to be the recursive nature of its definition. In some sense in order to decide if two agents are observationally equivalent one needs to check that they can perform the same sequences of actions and that the subagents reached after each sequence still have equivalent behaviour. Some of the resulting distinctions are concerned only with the internal structure of processes and an interesting critique of observational equivalence is given in /Dar82/. There the author gives an alternative equivalence. However it applies only to finite terms and there is no obvious extension to infinite terms.

A similar critique is put forward by John Kenneway in /Ken81/, where another equivalence is proposed. Also this equivalence is based on recursively proposing external experiments to processes and comparing their outcomes. In this case the particular kind of allowed experiments give us a smaller insight into the structure of systems. The equivalence obtained in this way is very similar to the one proposed by Darondeau /Dar82/ and it is the basis of one induced by the denotational approach of Hoare, Brookes and Roscoe /HBR81/ which is discussed in detail in /DeN83/.

In the present section we will describe and discuss Kennaway's "weak" equivalence by adapting the definition given for his calculus (NSCP) to transition systems. Moreover we will show that though defined recursively weak equivalence does not exploit the full power of recursion, in fact we will prove that it is possible to give (and indeed we give) a non recursive characterization for it.

We start with some definitions based on those of section 1. If A is the set of visible actions we will let L, M to range over the finite subsets of A moreover we will let R, P denote sets of states as before.

If $T = (Q, A, -\mu\rightarrow, q_0)$ is a transition system, $p, q \in Q$ and $P, R \subseteq Q$, $L \subseteq A$, $s \in A^*$ and $a \in A$ then we have:

Definition 4.1

i. p after $s = \{p' \mid p = s \Rightarrow p'\}$

ii. P after $s = \bigcup \{p$ after $s \mid p \in P\}$

From this definition and from the definition of $=s\Rightarrow$ it is very easy to derive:

Proposition 4.2

(P after a) after s = P after as

Definition 4.3

i. p MUST L if and only if for all p' such that $p \Rightarrow p'$, there exists $a \in L$ such that $p' = a \Rightarrow$.

ii. P MUST L if and only if p MUST L for all $p \in P$.

We can now adapt Kenneway's weak equivalence (/Ken81/ def. 4.4.8, pg. 93) to labelled transition systems.

Definition 4.4

If $P, R \subseteq Q$ are subset of states of a transition system T we have

$P \approx_0 R$ is always true

$P \approx_{n+1} R$ if and only if for all finite $L \subseteq A$

i. P MUST L iff R MUST L and

ii. for all $a \in A$. P after $a \approx_n R$ after a

$P \approx R$ if and only if for all $n \geq 0$ $P \approx_n R$. □

As for observational equivalence it is easy to extend weak equivalence between two states to transition systems.

Definition 4.5

If T_1 and T_2 are two transition systems with initial states p_0 and q_0 and T is a transition system obtained from T_1 and T_2 as in definition 3.2 then $T_1 \approx T_2$ iff $p_0 \approx q_0$. □

We can give an alternative characterisation of $\approx$ which does not involve any recurrence.

Theorem 4.6

$P \approx R$ if and only if for all $s \in A^*$, for all finite $L \subseteq A$

(P after s) MUST L iff (R after s) MUST L

Proof

1. ($\Longleftarrow$)

It is enough to prove that $P \not\approx R$ implies there exist $s \in A^*$, and $L \subseteq A$ such that (P after s) MUST L and (R after s) ~~MUST~~ L, or viceversa in P and R. If $P \not\approx R$ then there exists $n \geq 0$ such that $P \not\approx_n R$. We prove the claim by induction on n.

a. induction basis.

$P \not\approx_1 R$ implies there exists some L such that P MUST L and R ~~MUST~~ L.
It follows trivially that (P after ε) MUST L and (R after ε) ~~MUST~~ L, or viceversa in P and R.

b. inductive step.

We have $P \not\approx_{n+1} Q$ if and only if i) $P \not\approx_1 Q$ or ii) there exists $a \in A$ such that P after $a \not\approx_n Q$ after a. In case i) the claim follows from the induction basis. In case ii) we have by the inductive hypothesis that for some $a \in A$, $s \in A^*$

(P after a) after s MUST L and (Q after a) after s ~~MUST~~ L, i.e.
(P after as) MUST L and (Q after as) ~~MUST~~ L.

2. ($\Longrightarrow$)

Suppose there exists some $s \in A^*$ and some finite $L \subseteq A$ such that
(P after s) MUST L and (Q after s) ~~MUST~~ L. We prove by induction on s that $P \not\approx Q$.

a. induction basis, $s = \varepsilon$.

Trivial, since p after $\varepsilon = \{p' \mid p \Longrightarrow p'\}$

b. inductive step, $s = as'$.

Then (P after a) after s' MUST L whereas (Q after a) after s' ~~MUST~~ L. By induction P after $a \not\eqsim Q$ after a and so $P \not\eqsim Q$. $\square$

This result allows us to derive two propositions which relate Kennaway's equivalence to string equivalence and observational equivalence.

Proposition 4.7

$p \eqsim q$ implies Traces(p) = Traces(q).

Proof

Suppose there exists s such that $s \in$ Traces(p) and $s \notin$ Traces(q). Let a be such that $p \overset{sa}{\Longrightarrow}$ (a exists since $\xrightarrow{\mu}$ is image finite and A is infinite). Then (p after s) ~~MUST~~ $\{a\}$ whereas vacuously (q after s) MUST $\{a\}$, i.e. $p \not\eqsim q$.

$\square$

Proposition 4.8

If $p \approx_2 q$ then $p \eqsim q$ $\square$

Proof

Suppose $p \not\eqsim q$ then by theorem 4.6 there exists $s \in A^*$ and $L \subseteq A$ such that without loss of generality (p after s) MUST L while (q after s) ~~MUST~~ L. This implies that q after $s \neq \emptyset$ and either i. p after $s = \emptyset$ or ii. p after $s \neq \emptyset$.

In case i. we have that $s \in$ Traces(q) and $s \notin$ Traces(p), i.e. $p \not\approx_1 q$; in case ii. we have that there exist s, L and q' such that $q \overset{s}{\Longrightarrow} q'$ and $q' \overset{a}{\not\Longrightarrow}$ for all $a \in A$ while $p \overset{s}{\Longrightarrow} p'$ implies $p' \overset{a}{\Longrightarrow}$ for some $a \in A$. The latter implies that for all q' such that $q \overset{s}{\Longrightarrow} q'$ we have $p' \not\approx_1 q'$. $\square$

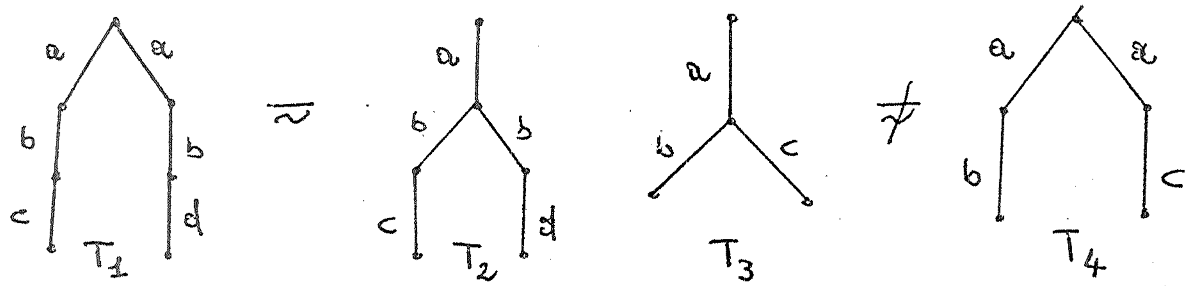
A direct implication of this proposition is

Corollary 4.9

If $p \approx q$ then $p \eqsim q$ $\square$

This corollary shows that weak equivalence is at least as coarse as observational equivalence. The equivalence of next example shows that it is indeed coarser, in the sense that it identifies more systems.

Examples



In fact we have that if we let p_i denote the initial state of T_i $(i = 1,2,3,4)$ then we have $Traces(p_1) = Traces(p_2)$ and that for every $s \in A^*$ the sets of possible moves of T_1 and T_2, after they have performed s, do coincide; this is sufficient to show that $T_1 \approx T_2$. On the other hand we have (q_4 <u>after</u> a) MUST $\{b\}$ while (q_3 <u>after</u> a) $\not$MUST $\{b\}$, i.e. $T_3 \not\approx T_4$. □

Corollary 7.9 together with the above examples shows that weak equivalence abstracts from the internal structure of systems better then observational equivalence and that it still keeps the ability to detect potential deadlocks. However the problem of the identification of systems which perform an infinite number of internal actions with systems which do not perform any action is left unsolved.

In the next section we propose a different approach to behavioural equivalence explicitely based on the notion of testing by external observers which takes care of this problem.

§5. A Theory of Testing

The external behaviour of programs or processes, in general of systems, can be investigated by a series of tests, /Moo56/. For example with sequential systems we can associate a test with a pair consisting of a predicate on the input domain and a predicate on the output domain. It is very easy to see how the input-output function of a program can be characterised by a set of such tests. For more general systems more general kinds of tests are needed. When the processes involved may be nondeterministic it is important to know not only whether given a particular test a process responds favourably or unfavourably to it but also if the process responds consistently all the times the test is performed. In fact all the approaches to systems equivalences discussed in the previous sections are based on the notions of external observation and presuppose implicitely the existence of a set of observers, a way of observing and of criteria for judging the results of an observation.

In general one can think of a set of processes and a set of relevant tests. Then two processes are equivalent (with respect to this set of tests) if they pass exactly the same set of tests. The rest of the section is an attempt at formalising this notion. The equivalence is based on two preorders on processes. The first is formulated in terms of the ability to respond positively to a test, the second in terms of the inability not to responds positively to a test. In the latter case a process p will be considered "less than" a process q if whenever p must respond positively to a particular test, q must also respond positively. The natural equivalence between processes is obtained by taking the equivalence associated with the conjunction of these two preorders (which is a third preorder).

The rest of the section is devoted to setting up a rather general framework within which we may discuss testing of processes and the tabulation of the possible outcomes. It is essentially a resumé of §1 of /DeH83/. There it is also showed how we can cope with partially specified states. It should be possible to adapt this general setting to various models of computation. In /DeH83/ we showed how to view CCS (/Mil80/) as a particular example of the general setting; in the next section we show how this applies to transition systems.

We assume a predefined set of states, <u>States</u>, and we let s range over <u>States</u>. A <u>computation</u> is any non-empty sequence of states. We let <u>Comp</u> denote the set of computations, ranged over by c. Note that a computation may be finite or infinite.

Let $\mathcal{O}$, $\mathcal{P}$ (ranged over by o, p respectively) be sets of predefined <u>observers</u> and <u>processes</u>. Observers may be thought of as agents which perform tests. The effect of observers performing tests on processes may be formalised by saying that for every o and p there is a non-empty set of computations Comp(o,p). If c Comp(o,p) then the result of o testing p may be the computation c. To indicate that a process passes a test we choose some subset of <u>States</u>, denoted <u>Success</u>, to be <u>successful</u> states. Then a computation is <u>successful</u> if it contains a successful state. On the other hand a computation will be called <u>unsuccessful</u> if it contains no successful state.

We may tabulate the effect of an observer o testing a process p by noting the types of computations in Comp(o,p).

For every $o \in \mathcal{O}$, $p \in \mathcal{P}$ let $R(o,p) \subseteq \{\top, \bot\}$, (the <u>result</u> set), be defined by:

i) $\top \in R(o,p)$ if there exists $c \in Comp(o,p)$ such that c is <u>successful</u>.

ii) $\bot \in R(o,p)$ if there exists $c \in Comp(o,p)$ such that c is <u>unsuccessful</u>

Thus in effect we can distinguish between processes which cannot fail a test (the result set is $\{T\}$) and processes which may pass a test (the result set is $\{T,\bot\}$). This will be elaborated upon shortly. A natural equivalence between processes immediately suggest itself:

$$p \sim^{\mathcal{O}} q \text{ if for every } o \in \mathcal{O}, \quad R(o,p) = R(o,q).$$

However as mentioned above it will be more fruitful to consider instead preorders, i.e. relations which are transitive and reflexive. A preorder $\lesssim$ generates an equivalence $\simeq$ in a natural way, $\simeq\ = (\lesssim \cap \gtrsim)$. Preorders are more primitive than equivalences and therefore we may use them to concentrate on more primitive notions which combine to form the equivalence $\sim^{\mathcal{O}}$. The preorders are based on:

Definition 5.1

a) p may satisfy o if $T \in R(o,p)$
b) p must satisfy o if $\{T\} = R(o,p)$ □

Thus p may satisfy o if there is a resulting successful computation whereas p must satisfy o if every resulting computation is successful.

Definition 5.2 a) $p \sqsubseteq_{\sim 3}^{\mathcal{O}} q$ if for all $o \in \mathcal{O}$ p may satisfy o implies q may satisfy o

b) $p \sqsubseteq_{\sim 2}^{\mathcal{O}} q$ if for all $o \in \mathcal{O}$ p must satisfy o implies q must satisfy o

c) $p \sqsubseteq_{\sim 1}^{\mathcal{O}} q$ if and only if $p \sqsubseteq_{\sim 2}^{\mathcal{O}} q$ and $p \sqsubseteq_{\sim 3}^{\mathcal{O}} q$. □

The following is trivial to establish

Proposition 5.3 $p \sim^{\mathcal{O}} q$ if and only if $p \sim_1^{\mathcal{O}} q$.

In /DeH83/ it is shown how $\sqsubseteq_{\sim 1}^{\mathcal{O}}$, $\sqsubseteq_{\sim 2}^{\mathcal{O}}$, $\sqsubseteq_{\sim 3}^{\mathcal{O}}$ arise in a natural way respectively from the Hoare, Smyth and Egli-Milner Powerdomains, /Plo76, Smy78/.

In the next section we apply this general theory of testing to transition systems. To do so we need to specify.

$\mathcal{P}$ - a set of processes
$\mathcal{O}$ - a set of observers
States - a set of states, together with a subset of successful states.
Comp - a method of assigning to every observer and process a non-empty set of computations (sequences of states).

§6. Testing Transitions Systems

§6.1 Testing Equivalences

In this section we show how to view Labelled Transition Systems as a particular instance of the general setting of the previous section.

Processes will be LTS's over an alphabet A of elementary actions. The set of all such processes will be denoted by $\mathcal{T}$ and ranged over by T, V, T_1, T_2 ....

Observers will be LTS's over the alphabet A U $\{w\}$, where w $\notin$ A and is the event we will use to "report success". The set of observers we will use to experiment on transition systems will be denoted by $\mathcal{E}$ and ranged over by E, E_1, E_2... .

States will be pairs $\langle p, e\rangle$ where p is a state of a process and e is a state of an experiment. A successful state is a state whose right component is able to perform a w-move. We will say that a state diverges if one of its two component does.

Computations: given two transitions systems T, E from $\mathcal{T}$ and $\mathcal{E}$ respectively, with initial states t_o and e_o a computation is a sequence of states such that the initial state is $\langle t_o, e_o\rangle$ and

$\langle t_n, e_n\rangle \overset{\tau}{\longrightarrow} \langle t_{n+1}, e_{n+1}\rangle$ if $t_n \overset{\tau}{\longrightarrow} t_{n+1}$ and $e_n \Longrightarrow e_{n+1}$ or $t_n \Longrightarrow t_{n+1}$ and $e_n \overset{\tau}{\longrightarrow} e_{n+1}$.

$\langle t_n, e_n\rangle \overset{a}{\longrightarrow} \langle t_{n+1}, e_{n+1}\rangle$ if $t_n \overset{a}{\longrightarrow} t_{n+1}$ and $e_n \overset{a}{\longrightarrow} e_{n+1}$.

Moreover every computation is maximal, i.e. it is such that if it is finite (i.e. it contains a finite sequence of states) with final element $\langle t_n, e_n\rangle$ than does not exists a pair $\langle t_k, e_k\rangle$ such that $\langle t_n, e_n\rangle \overset{\mu}{\longrightarrow} \langle t_k, e_k\rangle$ for $\mu \in$ A U$\{\tau\}$. We will let Comp (T,E) denote the set of computations from $\langle t_o, e_o\rangle$, when T and E are clear from the context we will say also Comp(t_o, e_o).

From the general setting and from the previous instantiations we have

Definition 6.1.1

T may satisfy E if there exists s $\in$ A* such that

$\langle t_o, e_o\rangle \overset{s}{\Longrightarrow} \langle t_n, e_n\rangle$ and $e_n \overset{w}{\longrightarrow}$

T must satisfy E if if $\langle t_o, e_o\rangle \overset{\mu_1}{\longrightarrow} \langle t_1, e_1\rangle \overset{\mu_2}{\longrightarrow} \ldots$ is a (finite or infinite) computation then there exists n $\geq$ 0 such that $e_n \overset{w}{\longrightarrow}$. $\square$

After these definitions we can introduce the three preorders on LTS generated by them and the corresponding equivalence relations.

Definition 6.1.2

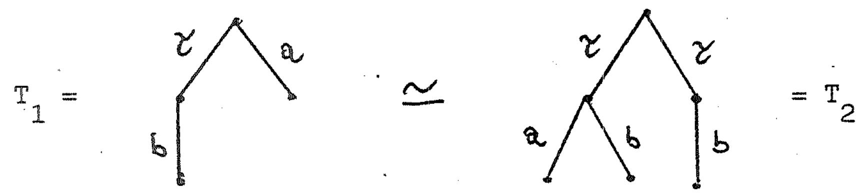
$T_1 \underset{\sim}{\sqsubset}_3 T_2$ if for all $E \in \mathcal{E}$ T may satisfy E implies T_2 may satisfy E

$T_1 \underset{\sim}{\sqsubset}_2 T_2$ if for all $E \in \mathcal{E}$ T must satisfy E implies T_2 must satisfy E

$T_1 \underset{\sim}{\sqsubset}_1 T_2$ if $T_1 \underset{\sim}{\sqsubset}_3 T_2$ and $T_1 \underset{\sim}{\sqsubset}_2 T_2$. $\square$

In the sequel we illustrate with a couple of examples the kind of equivalences induced by the previous definitions. $\simeq_1$ and $\underset{\sim}{\sqsubset}_1$ will be abbreviated as $\simeq$ and $\underset{\sim}{\sqsubset}$.

Example

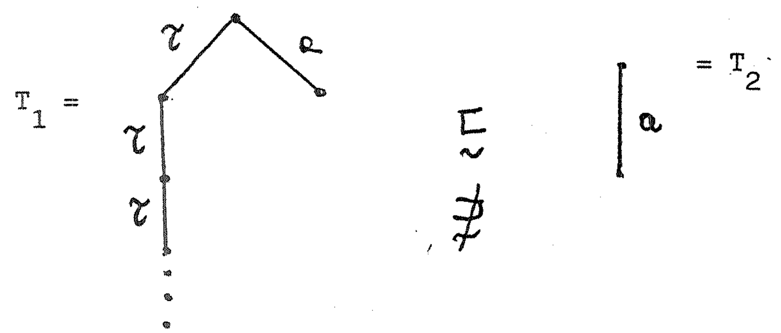


Proof

a) Suppose T_1 may satisfy E. If $e_o \overset{w}{\Longrightarrow}$ then we have T_2 may satisfy E; otherwise T_1 may satisfy E implies $e_o \overset{a}{\Longrightarrow} e' \overset{w}{\longrightarrow}$ or $e_o \overset{b}{\Longrightarrow} e'' \overset{w}{\longrightarrow}$, in both cases we have T_2 may satisfy E. A similar analysis will show that T_2 may satisfy E implies T_1 may satisfy E for all $E \in \mathcal{E}$.
b) Suppose T_1 must satisfy E. Since both T_1 and T_2 do not diverge we have that $e_o \overset{w}{\longrightarrow}$ implies T_2 must satisfy E. If $e_o \overset{w}{\longrightarrow}$ we have that T_1 must satisfy E implies that for every e_1 such that $e_o \Longrightarrow e_1$ we must have that $e_1 \overset{b}{\Longrightarrow} e'_1 \overset{w}{\longrightarrow}$ and that if $e_1 \overset{a}{\Longrightarrow} e''_1$ then $e''_1 \overset{w}{\longrightarrow}$. It is easy to see that in this cases T_2 must satisfy E. Again a similar analysis will show that T_2 must satisfy E implies T_1 must satisfy E for all $E \in \mathcal{E}$.

$\square$

Example



<u>Proof</u>

1. $T_1 \sqsubseteq_{\sim} T_2$

a) T_1 <u>may satisfy</u> E implies $e_o \overset{w}{\Longrightarrow}$ or $e_o \overset{a}{\Longrightarrow} e' \overset{w}{\longrightarrow}$; in both cases we have T_2 <u>may satisfy</u> E.
b) Since the initial state of T_1 diverges we have that T_1 <u>must satisfy</u> E implies that $e_o \overset{w}{\longrightarrow}$ and so we have T_2 <u>must satisfy</u> E.

2. $T_1 \not\sqsupseteq T_2$

It is easy to verify that if E is such that its only transitions are $e_o \overset{a}{\longrightarrow} e_1 \overset{w}{\longrightarrow}$ then T_2 <u>must satisfy</u> E while T_1 <u>must satisfy</u> E.

$\square$

§6.2 Alternative Characterizations

In the previous section we proposed a general approach to investigate the behaviour of a program or a process. The general situation may be expressed as follows. Given a set of processes and a set of "relevant" tests we consider equivalent two process if they pass exactly the same tests. In the first part of this section we have adapted this general setting to a particular model of computation: transition systems, by defining sets of relevant tests and what it means for a system to pass a test. Tough very intuitive the equivalences (preorders) obtained in this way are very difficult to verify. However at least in the case of transition systems it is possible to give alternative characterizations of the equivalences (preorders) which are independent from the notion of experimenter. It is based on the sequences (finite or infinite) of actions each system may perform and on the set of experiments the system must accept. These new characterization allow us to understand the similarities of testing equivalences with Kennaway's weak equivalence and to gain insight into their discriminating power. This is the subject of the rest of the section. The various definitions we give (<u>MUST</u>, <u>after</u>, etc.) will be based on the ones of section 1.

Let $T_1 = (P, A, \overset{\mu}{\longrightarrow}, p_o)$ and $T_2 = (Q, A, \overset{\gamma}{\longrightarrow}, q_o)$ be two transition systems, and $p\Downarrow$ be defined as in definition 1.3.

<u>Definition 6.2.1</u>

Given any state p of a transition system T we say

i. $p\Downarrow\varepsilon$ if $p\Downarrow$

ii. $p\Downarrow as$ if $p\Downarrow$ and $p \overset{a}{\Longrightarrow} p'$ implies $p'\Downarrow s$

As might be expected $\Uparrow s$ denotes the negation of $\Downarrow s$.

$\square$

Definition 6.2.2

$T_1 \underset{\sim}{\sqsubset}'_3 T_2$ if $Traces(p_o) \subseteq Traces(q_o)$

$T_1 \underset{\sim}{\sqsubset}'_2 T_2$ if for all $s \in A^*$, for all finite $L \subseteq A$, $p_o \Downarrow s$ implies

i. $q_o \Downarrow s$ and

ii. (p_o after s) MUST L implies (q_o after s) MUST L

$T_1 \underset{\sim}{\sqsubset}'_1 T_2$ if $T_1 \underset{\sim}{\sqsubset}'_3 T_2$ and $T_1 \underset{\sim}{\sqsubset}'_2 T_2$ □

Before proving the main characterization theorem we need some lemmas.

Lemma 6.2.3 If $T_1 \underset{\sim}{\sqsubset}_2 T_2$ then for all $s \in A^*$ $p_o \Downarrow s$ implies

i. $q_o \Downarrow s$

ii. $s \in Traces(q_o)$ implies $s \in Traces(p_o)$

Proof

i. Suppose there exists $s = a_1 \dots a_n$ such that $p_o \Downarrow s$ and $q_o \Uparrow s$. Then if we choose E such that its set of states is given by $\{e_i \mid 0 \leq i \leq n\} \cup \{e_w, e_f\}$, its initial state is e_o and its transition relation is given by $\{e_i \text{-}\tau\text{->} e_w,\ e_i \text{-}a_i\text{->} e_{i+1} \mid 0 \leq i < n\} \cup \{e_n \text{-}\tau\text{->} e_w,\ e_w \text{-}w\text{->} e_f\}$ then T_1 must satisfy E and T_2 ~~must satisfy~~ E i.e. $T_1 \not\underset{\sim}{\sqsubset}_2 T_2$.

ii. Suppose there exists $s = a_1 \dots\dots a_n$ such that $p_o \Downarrow s$, $s \in Traces(q_o)$ and $s \notin Traces(p_o)$. Then if we choose E similar to the one of case i. but such that the transition relation is extended with $e_n \text{-}a_n\text{->} e_f$ we have again T_1 must satisfy E while T_2 ~~must satisfy~~ E. □

Lemma 6.2.4 If (q after s) MUST L for some finite $L \subseteq A$ then $s \in Traces(q)$.

Proof Suppose $s \notin Traces(q)$, then q after s $= \emptyset$ and we have by definition $\emptyset$ MUST L for every finite $L \subseteq A$. □

Lemma 6.2.5

If $p_o \Downarrow s$ and $T_1 \underset{\sim}{\sqsubset}'_2 T_2$ then $s \in Traces(q_o)$ implies $s \in Traces(p_o)$.

Proof Suppose there exists some s such that $s \in Traces(q_o)$, $p_o \Downarrow s$ and $s \notin Traces(p_o)$. By the previous lemma (p_o after s) MUST L for every finite $L \subseteq A$. Since $q_o \Downarrow s$, by definition of $\underset{\sim}{\sqsubset}'_2$, and moreover from every state there is a finite number of outgoing arcs we have that $\cup \{Init(q') \mid q' \in q_o \text{ after } s\}$ is finite. Consequently, since A is infinite, we can find an a such that $q_o =sa\Rightarrow$. Then (p_o after s) MUST $\{a\}$ while (q_o after s) ~~MUST~~ $\{a\}$, which contraddicts the fact that $T_1 \underset{\sim}{\sqsubset}'_2 T_2$. □

We are now ready to prove the main characterization theorem.

Theorem 6.2.6

$T_1 \sqsubseteq_{\sim i} T_2$ if and only if $T_1 \sqsubseteq'_{\sim i} T_2$ for i=1,2,3.

Proof Because of the way $\sqsubseteq_{\sim 1}$ and $\sqsubseteq'_{\sim 1}$ have been defined we need only to prove the theorem for i = 2,3.

i = 3:
Let $s = a_1 \ldots a_n$, a_i A and E be such that the set of its states is given by $\{e_i \mid 0 \leq i \leq n\} \cup \{q_f\}$, the initial state is e_o and the transition relation is $\{e_i -a_i\to e_{i+1} \mid 0 \leq i < n\} \cup \{e_n -w\to q_f\}$. Then $s \in$ Traces(p_o) if and only if T may satisfy E for any $T \in \mathcal{T}$. The claim is an easy corollary of this fact.

i = 2:
a. We prove first that $T_1 \sqsubseteq_{\sim 2} T_2$ implies $T_1 \sqsubseteq'_{\sim 2} T_2$.
From lemma 6.2.3 we have that $p_o \Downarrow s$ implies $q_o \Downarrow s$ for all $s \in A^*$. We are left to prove that for all finite $L \subseteq A$, for all $s \in A^*$ we have (p_o after s) MUST L implies (q_o after s) MUST L. Let s be $a_1 \ldots . a_n$ and E be a transition systems whose set of states is given by $\{e_i \mid 0 \leq i \leq n\}$ $\cup \{e_w, e_f\}$ with e_o as initial state and whose transition relation is given by $\{e_i -\tau\to e_w, e_i -a_i\to e_{i+1} \mid 0 \leq i < n\} \cup \{e_n -a\to e_w \mid a \in L\} \cup$ $\{e_w -w\to e_f\}$. It is easy to check that (p_o after s) MUST L implies T_1 must satisfy E which in turn implies T_2 must satisfy E by hypothesis. The latter implies (q_o after s) MUST L since we have either that $s \notin$ Traces(q_o) or that for all q such that $q_o =s\Rightarrow q$, $q = a \Rightarrow$ for some $a \in L$.

b. $T_1 \sqsubseteq'_{\sim 2} T_2$ implies $T_1 \sqsubseteq_{\sim 2} T_2$.
Suppose there exist $E \in \mathcal{E}$ such that T_1 must satisfy E, we have to prove that T_2 must satisfy E. We will prove that if there exist c_2, $c_2 \in$ Comp(p_o, e_o), with c_2 unsuccessful, then there exists an unsuccessful c_1, $c_1 \in$ Comp(p_o, e_o). We have that c_2 may be unsuccessful for a number of reasons:

i. $c_2 = \langle q_o, e_o\rangle -\mu_1\to \ldots . -\mu_n\to \langle q_n, e_n\rangle$, and $\langle q_n, e_n\rangle -\mu\not\to$ for all $\mu \in A \cup \{\tau\}$, $e_i -w\not\to$ and $q_i\Downarrow$ and $e_i\Downarrow$ for all $0 \leq i \leq n$.

ii. $c_2 = \langle q_o, e_o\rangle -\mu_1\to \ldots . -\mu_n\to \langle q_n, e_n\rangle -\mu\to \ldots .$ and $q_h \Uparrow$ or $e_k \Uparrow$ for some positive h, $k \leq n$ and $e_i -w\not\to$ for all $0 \leq i < h$ or $0 \leq i < k$ respectively.

iii. c_2 is such that for all states, $\langle q_n, e_n\rangle$, reacheable in a finite number of steps we have $q_n\Downarrow$ and $e_n\Downarrow$, $e_n -w\not\to$ and $\langle q_n, e_n\rangle =a\Rightarrow$ for some $a \in A$.

In all these cases we can prove that there exist $c_1 \in$ Comp(p_o, e_o) which is unsuccessful; this is sufficient to prove the claim.

i. We have that there exists $s \in A^*$ such that $\langle q_o, e_o\rangle =s\Rightarrow \langle q_n, e_n\rangle$, $q_o \Downarrow s$, $e_o \Downarrow s$ and (q_o after s) MUST Init(e_n). We may have either $p_o \Uparrow s$, in which case since $e_o =s\Rightarrow$ there exist an unsuccessful computation from $\langle p_o, e_o\rangle$, or $p_o \Downarrow s$, in which case we have that $T_1 \sqsupseteq'_2 T_2$ implies that if $s \in Traces(q_o)$ than $s \in Traces(p_o)$ and so that there exist c_1 such that $\langle p_o, e_o\rangle =s\Rightarrow \langle p_n, e_n\rangle$, since $T_1 \sqsupseteq'_2 T_2$ and if (q_o after s) MUST Init(e_n) then (p_o after s) MUST Init(e_n), and both in c_1 and c_2 E goes trough the same sequence of states, we have that c_1 is unsuccessful.

ii. We have that there exists $s \in Traces(q_o) \cap Traces(e_o)$ such that $q_o \Uparrow s$ or $e_o \Uparrow s$. We have that this and $T_1 \sqsupseteq'_2 T_2$ imply $p_o \Uparrow s$ or $e_o \Uparrow s$; since E may go trough the same sequences of states we have that there exist and unsuccessful computation from $\langle p_o, e_o\rangle$.

iii. We have that $q_n \Downarrow$ for all states of the computation, i.e. we have that for any $s \in A^*$ such that $\langle q_o, e_o\rangle =s\Rightarrow \langle q_n, e_n\rangle$, $q_o \Downarrow s$ and either $p_o \Uparrow s$ or $\langle p_o, e_o\rangle =s\Rightarrow \langle p_m, e_n\rangle$, for some $m \geq 0$. With reasoning similar to case i. we can prove that there exist an unsuccessful c_1. □

§7. Alternative Forms of Testing

Many of the notions (observer, state, computation) used in §5 to set up the general framework for testing systems seem very natural and correspond to precise intuitions; on the contrary the way of tabulating the possible outcomes of observation and especially the way of noting the types of computations generated by testing a process p with an observer o, is more debatable. In particular there are various possible ways of tabulating the effects of testings (observations) which lead to infinite or divergent computations.

In §5 we chose to consider successful a computation with a diverging state which had gone through a successful state before going through the diverging one. This choice has been vindicated by the simple alternative characterization of the equivalences the derived general framework induces on transition systems. We could have taken an apparently more natural approach by considering successful only those computations which never go through divergent states, i.e. we could have considered successful a computation only if whenever it cannot progress any further it is able to report a success. The present section will be dedicated at discussing this alternative choice by first slightly modifying the general setting of §5 and then applying this to labelled transition systems as in §6. In section 5 we had:

$\perp \in R(o,p)$ if there exists $c \in Comp(o,p)$ such that c is unsuccessful

The new approach will imply that given any observer o and any process p we get a new result set $R'(o,p)$ such that

$\perp \in R'(o,p)$ if there exists $c \in Comp(o,p)$ such that

a. c is unsuccessful

or

b. c is infinite

$T \in R'(o,p)$ if $T \in R(o,p)$

When applied to transition systems this would imply new definition for must satisfy and new preorders (must' satisfy, $\underset{\approx 2}{\sqsubset}$, $\underset{\approx 1}{\sqsubset}$). Note that the definition of may satisfy and $\underset{\sim 3}{\sqsubset}$ are not influenced by the present changes. If we keep the same notation and conventions of the previous section we have:

Definition 7.1

T must' satisfy E if if $\langle t_o, e_o\rangle \overset{\mu_1}{\Longrightarrow} \langle t_1, e_1\rangle \overset{\mu_2}{\Longrightarrow} \cdots \overset{\mu_n}{\Longrightarrow} \langle t_n, e_n\rangle$ is a computation then $e_n \overset{w}{\longrightarrow}$. $\square$

and

Definition 7.2

$T_1 \underset{\approx 2}{\sqsubset} T_2$ if for all $E \in \mathcal{E}$ T_1 must' satisfy E implies T_2 must' satisfy E

$T_1 \underset{\approx 1}{\sqsubset} T_2$ if $T_1 \underset{\approx 2}{\sqsubset} T_2$ and $T_1 \underset{\sim 3}{\sqsubset} T_2$. $\square$

As for $\underset{\sim i}{\sqsubset}$, also for $\underset{\approx 2}{\sqsubset}$ it is possible to give an alternative characterization which is independent from the notion of observers and is based on the set of sequences a process may perform and on the notion of MUST of section 4. The characterization will allow to understand precisely the difference between the equivalences obtained by immersing transitions systems in the two different general settings and in particular the relationships between $\underset{\sim 2}{\sqsubset}$ and $\underset{\approx 2}{\sqsubset}$. Given two transition systems T_1 and T_2, if we let $D(s, p) = \{a \mid a \in A, p \Uparrow sa\}$ we have:

Definition 7.3

$T_1 \underset{\approx}{\sqsubseteq}'_2 T_2$ if for all $s \in A^*$ and for all finite $L \subseteq A$, $L \cap D(s,p_o) = \emptyset$ and $p_o \Downarrow s$ implies

i. $q_o \Downarrow s$

ii. (p_o after s) MUST L implies (q_o after s) MUST L $\square$

As we did in the previous section with $\underset{\approx}{\sqsubseteq}_i$ and $\underset{\approx}{\sqsubseteq}'_i$ we can prove that $\underset{\approx}{\sqsubseteq}_2$ and $\underset{\approx}{\sqsubseteq}'_2$ coincide. As before we need some lemmas.

Lemma 7.4

If T_1 and T_2 are two transition systems with initial state p_o and q_o respectively then $T_1 \underset{\approx}{\sqsubseteq}'_2 T_2$ and $p_o \Downarrow s$ implies

i. $q_o \Downarrow s$

ii. $s \in Traces(q_o)$ implies $s \in Traces(p_o)$

Proof

To prove i. suppose there exists a trace s such that $p_o \Downarrow s$ and q_o s then we can prove there exists an experiment E such that T_1 must satisfy E and T_2 must satisfy E. If $s = a_o \ldots a_n$ it will be enough to have an E such that its set of states R is given by $\{q_i \mid 0 \leq i \leq n+1\} \cup \{q_f\}$, q_o is its initial state and its transition relation is given by $\{q_i \text{-w->} q_f,\ q_i \text{-}a_i\text{->} q_{i+1} \mid 0 \leq i \leq n\} \cup \{q_{n+1} \text{-w->} q_f\}$. To prove ii. suppose there exists $s = a_1 \ldots a_n$ such that $s \in Traces(q_o)$ and $s \notin Traces(p_o)$ then the experimenter E^s equal to E but such that q_{n+1}-w-> would be such that T_1 must satisfy E^s and T_2 must ~~satisfy~~ E^s. $\square$

Lemma 7.5

If $T_1 \underset{\approx}{\sqsubseteq}'_2 T_2$ and $p_o \Downarrow s$ then $s \in Traces(q_o)$ implies $s \in Traces(p_o)$

Proof It follows the same lines of the proof of lemma 6.2.5.. $\square$

Theorem 7.6

$T_1 \underset{\approx}{\sqsubseteq}_2 T_2$ if and cnly if $T_1 \underset{\approx}{\sqsubseteq}'_2 T_2$.

Proof

a. ($\Rightarrow$)

The proof follows the same pattern of the one for part a. of theorem 6.2.6, with E replaced by E', defined as follows. The set of states of E' is $\{e_i \mid 0 \leq i \leq n\} \cup \{e_w, e_f\}$, the initial state is e_o and the transition relation is given by $\{e_i \text{-w->} e_f,\ e_i \text{-}a_i\text{->} e_{i+1} \mid 0 \leq i < n\} \cup \{e_n \text{-a->} e_w \mid a \in L\}$. In fact it is not difficult to prove that given an s such that $p_o \Downarrow s$ and an L such that $L \cap D(s, p_o) = \emptyset$, we have that ($p_o$ after s) MUST L ⊛ from lemma 7.3 we have also $p_o \Downarrow s$ implies $q_o \Downarrow s$.

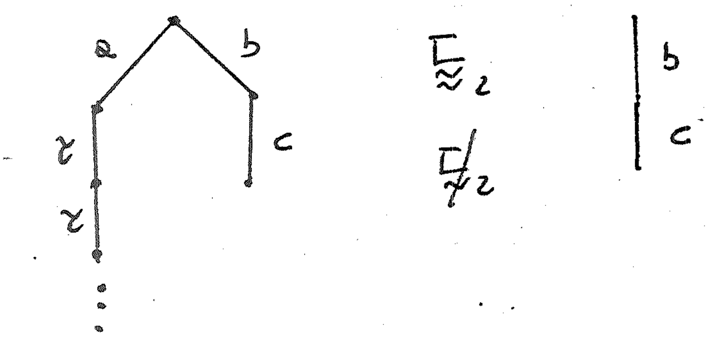
⊛ implies T_1 must' satisfy E' which (by hypothesis) implies T_2 must' satisfy E' and the latter implies (q_o after s) MUST L;

b. ($\Leftarrow$=)
Also this proof follows the same pattern of the one for theorem 6.2.6 (part b.). We have that $c_2 \in Comp(q_o\ e_o)$ may be unsuccessful for a number of reasons; we consider only one of them since the others are treated exactly in same way as in theorem 6.2.6.

We may have that there exists an $s \in Traces(q_o) \cap Traces(e_o)$ such that $\langle q_o, e_o\rangle \overset{s}{\Longrightarrow} \langle q, e\rangle$, $q_o \Downarrow s$, $e_o \Downarrow s$ and $\langle q, e\rangle \overset{\mu}{\not\longrightarrow}$ for all $\mu \in A \cup \{\tau, w\}$. This implies ($q_o$ after s) MUST Init(e). If $Init(e) \cap D(s, p_o) = \emptyset$ then we have also (p_o after s) MUST Init(e) and this implies there exists $c_1 \in Comp(p_o, e_o)$ which is unsuccessful: $\langle q_o, e_o\rangle \overset{s}{\Longrightarrow} \langle p, e\rangle \overset{\mu}{\not\longrightarrow}$ for all $\mu \in A \cup \{\tau, w\}$. If $Init(e) \cap D(s, p_o) \neq \emptyset$ then we would have there exists $p \in p_o$ after s such that there exist an $a \in Init(e) \cap Init(p)$ such that $\langle p_o\ e_o\rangle \overset{sa}{\Longrightarrow} \langle p', e'\rangle$ and $p' \Uparrow$ i.e. there is an infinite computation from $\langle p_o, e_o\rangle$ and so an unsuccessful one. $\square$

The alternative characterizations of $\underset{\sim}{\sqsubset}_2$ and $\underset{\approx}{\sqsubset}_2$ given in definition 6.2.2 and definition 7.2 should suffice to convince the reader that, in case we consider only transition systems which do not contain divergent states, the two preorders coincide. In the next section we will state and prove this interrelations formally. Anyway in case the systems considered have diverging states the two preorders are very different. In particular the preorder $\underset{\approx}{\sqsubset}_2$ seems to overestimate the fact that after performing a particular action a system may diverge, in fact it overstimate divergence up to the point to ignore the fact that the system may perform that action. Some examples will help to understand this point.

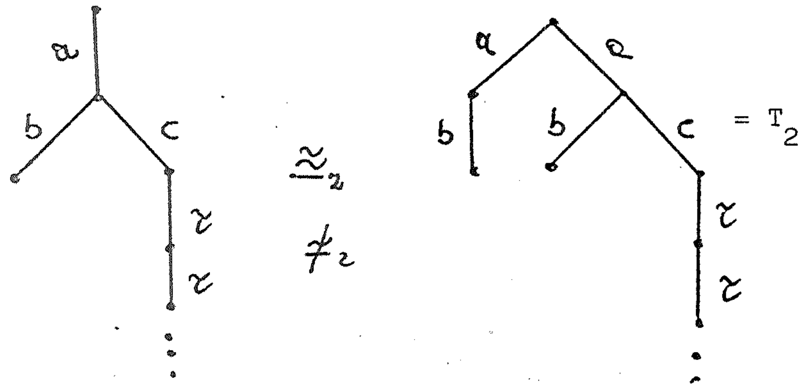
Example

T =  = V

In fact we have (T after s) MUST L implies (V after s) MUST L for all $s \in A^*$ and for all finite $L \subseteq A - \{a\}$. These result does not correspond to any intuitive notion of approximation. $\square$

Example

T_1 =



= T_2

This result can be proved with reasoning similar to the previous ones. Anyway also this equivalence does not match intuitions about the behaviour of systems. In fact we have that T_1, after it has accepted an a-experiment , will certainly accept the c-experiment while T_2 may or may not accept c depending on which a-experiment it has accepted. Note that we have T $\not\underset{\sim}{\sqsubset}_2$ V, since T MUST {a} while V ~~MUST~~ {a}, and $T_1 \not\underset{\sim}{\sqsubset}_2 T_2$, since ($T_1$ after a) MUST {c} while (T_2 after a) ~~MUST~~ {c}. In fact we started with the general setting which generates $\underset{\approx}{\sqsubset}_2$, since it seemed more natural than the general setting which generates $\underset{\sim}{\sqsubset}_2$, but the difficulties $\underset{\approx}{\sqsubset}_2$ has in handling divergent terms convinced us to study further only $\underset{\sim}{\sqsubset}_2$.

§8. Comparisons and Discussion

The alternative characterizations for $\underset{\sim}{\sqsubset}_2$ and $\underset{\approx}{\sqsubset}_2$ and the one for Kennaway's weak equivalence suggest that there are strong similarities between them. In this section we relate with each other the various equivalences we have presented by first stressing the similarities between $\eqsim$ and $\underset{\sim}{\sqsubset}_2$, $\underset{\approx}{\sqsubset}_2$ and $\eqsim$ (we prove they coincide if we consider only strongly convergent transition systems) and then by using previous results about the relation between weak equivalence and observational equivalence ($\approx$). We first define formally what we mean by strongly convergent transition systems.

Definition 8.1

A transition system T = (Q, A, $\xrightarrow{\mu}$, q_0) is strongly convergent if and only if for all $s \in A^*$ $q_0 \overset{s}{\Longrightarrow} q$ implies $q\Downarrow$. □

We may now state the main results, in general their proofs will trivially follow from results of the previous sections.

Proposition 8.2

If T_1 and T_2 are two strongly convergent transition systems then $T_1 \underset{\sim}{\sqsubset}_2 T_2$ if and only if $T_1 \underset{\approx}{\sqsubset}_2 T_2$.

Proof Straightforward from theorem 6.2.6 and theorem 7.6. $\square$

Proposition 8.3

If T_1 and T_2 are two strongly convergent transition systems then

i.	$T_1 \eqsim T_2$	implies	$T_1 \simeq_2 T_2$
ii.	$T_1 \simeq_2 T_2$	implies	$T_1 \simeq_1 T_2$
iii.	$T_1 \simeq_1 T_2$	implies	$T_1 \approxeq_1 T_2$
iv.	$T_1 \approxeq_1 T_2$	implies	$T_1 \approxeq_2 T_2$
v.	$T_1 \approxeq_2 T_2$	implies	$T_1 \eqsim T_2$

Proof

i. Follows from theorem 6.2.6 and theorem 4.6, the two theorems which give an alternative characterization of the two equivalences.

ii. From theorem 6.2.6 we have that $p_0 \simeq_3 q_0$ if and only if $Traces(p_0) = Traces(q_0)$ and that $p_0{\Downarrow}s$ together with $T_1 \underset{\sim}{\sqsubset}_2 T_2$ implies $Traces(p_0) = Traces(q_0)$. This implies that $\underset{\sim}{\sqsubset}_3$ does not give any contribution in the definition of $\underset{\sim}{\sqsubset}_1$ ($\underset{\sim}{\sqsubset}_1$ iff $\underset{\sim}{\sqsubset}_2$ and $\underset{\sim}{\sqsubset}_3$) when we consider only strongly convergent transition systems.

iii. Follows from proposition 8.2.

iv. Follows from reasonings similar to the ones for ii.

v. Like i., v. follows from theorem 6.2.6 and theorem 4.6. $\square$

This proposition allow us to conclude that the equivalences obtained by testing transition systems ($\underset{\sim}{\sqsubset}_1$, $\underset{\sim}{\sqsubset}_2$, $\underset{\approx}{\sqsubset}_1$ and $\underset{\approx}{\sqsubset}_2$) they all coincide with Kennaway's weak equivalence when we restrict ourself to strongly convergent systems and that the differences between the two general settings for testing systems shows themselves only in case we have divergent systems. Moreover the last proposition and previous theorems about the relationships between Milner's observational equivalences and weak equivalence allow a precise taxonomy of the equivalences presented up to now, at least for strongly convergent systems.

In the general case we can only prove that $\underset{\approx}{\sqsubset}_2$ is coarser than $\underset{\sim}{\sqsubset}_2$.

Proposition 8.4

If T_1 and T_2 are two transition systems then $T_1 \sqsubseteq_2 T_2$ implies $T_1 \underset{\approx}{\sqsubset}_2 T_2$.

Proof Follows from theorem 6.2.6 and theorem 7.6. In fact we have, (P after s) MUST L implies (Q after s) MUST L for all $L \subseteq A$ implies (P after s) MUST L' implies (Q after s) MUST L' for every $L' \subseteq A'$, $A' \subseteq A$. □

This proposition together with the last two examples is indeed sufficient to shows that $\underset{\approx}{\sqsubset}_2$ forces more identifications than $\underset{\sim}{\sqsubset}_2$.

Another equivalence worth mentioning and relating to the others before concluding our excursus is failure equivalence, introduced in /HBR81/ and studied at lenght in /Br83/. This equivalence as discussed in /DeN83/ has difficulties in coping with divergent terms; we will consider only its restriction to strongly convergent terms. The reason we have not discussed it previously is that it turns out to be simply a reformulation of the alternative characterizations of the testing equivalences generated by $\underset{\sim}{\sqsubset}_2$ and $\underset{\approx}{\sqsubset}_2$.

Definition 8.5

If T_1 and T_2 are two strongly convergent transition systems then $T_1 \simeq_f T_2$ if and only if for all $s \in A^*$, for all finite $L \subseteq A$

$$\exists q. \quad q_o = s \Rightarrow q \ \underline{\text{and}}\ \text{Init}(q) \cap L = \emptyset$$
$$\text{if and only if}$$
$$\exists p. \quad p_o = s \Rightarrow p \ \underline{\text{and}}\ \text{Init}(p) \cap L = \emptyset$$ □

Theorem 8.6

If T_1 and T_2 are two strongly convergent transitions systems then $T_1 \simeq_f T_2$ if and only if $T_1 \simeq_2 T_2$.

Proof From theorem 6.2.6 and definition 6.2.2 we have that

$T_1 \simeq_2 T_2$ if and only if for all $s \in A^*$, for all finite $L \subseteq A$

(p_o after s) MUST L iff (q_o after s) MUST L

and from the definition of MUST given in §4 we have:

$T_1 \simeq_2 T_2$ if and only if for all $s \in A^*$ for all finite $L \subseteq A$

$$\forall q.\ q_o = s \Rightarrow q, \text{ implies } \text{Init}(q) \cap L \neq \emptyset$$
$$\text{if and only if}$$
$$\forall p.\ p_o = s \Rightarrow p, \text{ implies } \text{Init}(p) \cap L \neq \emptyset.$$

The claim follows from simple logical manipulations. □

ACKNOWLEDGMENTS

The author whishes to thank Matthew Hennessy, with whom most of the ideas presented in the paper were developed, for many detailed comments on a first draft. Thanks are due also to Anna Passerotti and Manuela Mennucci Bernardini for their accurate typing.

References

LNCS n denotes Lecture Notes in Computer Science Volume No. Springer--Verlag.

/BR83/ Brookes S.D. A Model for Communicating Sequential Processes, Ph. D. thesis. University of Oxford, 1983.

/BrR83/ Brookes S.D., Rounds W.C. Behavioural Equivalence Relations induced by Programming Logics, Proc. ICALP '83, LNCS 154, 1983.

/Dar82/ Darondeau, Ph. An enlarged definition and complete axiomatization of observational congruence of finite processes, LNCS 137, pp. 47-62, 1982.

/DeN83/ De Nicola, R. A Complete Set of Axioms for a Theory of Communicating Sequential Processes, Proc. FCT '83, LNCS 158, 1983.

/DeH83/ De Nicola, R. and Hennessy, M. Testing Equivalences for Processes, Technical Report CSR-123-82, University of Edinburgh. To appear in Theoretical Computer Science. A Short version in Proc. ICALP '83, LNCS 154, 1983.

/Hen82/ Hennessy, M. Powerdomains and nondeterministic recursive definitions, LNCS 137, pp. 178-193, 1982.

/HM80/ Hennessy, M., Milner, R. On observing Nondeterminism and Concurrency, LNCS 85, pp. 299-309, 1980.

/HM83/ Hennessy, M. Milner, R. Algebraic Laws for Nondeterminism and Concurrency, Technical Report CSR-133-83, University of Edinburgh. To appear in Journal of ACM.

/Hoa82/ Hoare, C.A.R. A Model for Communicating Sequential Processes. Technical Monograph Prg-22, Computing Laboratory, University of Oxford, 1982.

/HBR81/ Hoare, C.A.R., Brookes, S.D., and Roscoe, A.D. A Theory of Communicating Sequential Processes, Technical Monograph Prg-16, Computing Laboratory, University of Oxford, 1981. To appear in Journal of ACM.

/Kel76/ Keller, R. Formal Verification of Parallel Program, Communication of ACM No. 19, Vol. 7, 1986.

/Ken81/ Kennaway, J.K. Formal semantics of nondeterminism and parallelism, Ph.D. thesis, University of Oxford, 1981.

/Mil80/ Milner, R. A Calculus of Communicating Systems, LNCS 92, 1980.

/Mil84/ Milner, R. A Complete Inference System for a Class of Regular Behaviours, J.C.C.S., Vol. 28, No. 3, 1984.

/Moo56/ Moore, E. Gedanken Experiments on Sequential Machines, Automata Studies, edited by Shannon, C.E. and McCarthy, J., Princeton University Press, 1956.

/OH83/ Olderog, E.R. and Hoare, C.A.R. Specification-Oriented Semantics for Communicating Processes, Proc. ICALP '83, LNCS 154, 1983.

/Plo76/ Plotkin, G. A Powerdomain Construction, SIAM J. on Computing, No. 5, pp. 452-486, 1976.

/Plo81/ Plotkin, G. A Structural Approach to Operational Semantics, Lecture Notes Aharus University, DAIMI-FN-19, 1981.

/San82/ Sanderson, M.T. Proof Techniques for CCS, Ph. D. thesis, University of Edinburgh, CST-19-82, 1982.

/Sco76/ Scott, C.S. Data types as lattices. SIAM Journal on Computing, Vol. 5, No. 3, 1976.

/Smy78/ Smyth, M.B. Power Domains, JCCS, Vol. 2, pp. 23-26, 1978.

/Stoy77/ Stoy, J. Denotational Semantics: the Scott-Strachey approach to Programming Language Theory, MIT Press, 1977.