OBSERVATION EQUIVALENCE AS A TESTING EQUIVALENCE*

Samson ABRAMSky

Department of Computing, Imperial College of Science and Technology, London SW7 2BZ, United Kingdom

Communicated by R. Milner
Received November 1986
Revised April 1987

Abstract. A notion of testing is developed for transition systems with divergence. The forms of testing include traces, refusals, copying and global testing. Both denotational and operational formulations of testing are given. The equivalence based on this notion of testing is shown to coincide with observation equivalence.

Key words. Concurrency, transition systems, operational semantics, denotational semantics, observation equivalence, testing, powerdomains.

1. Introduction

Observation equivalence was introduced in [7]. It has become recognised as a fundamental notion in the theory of concurrency, for a number of reasons.

(i) Firstly, it has a natural mathematical-logical character (cf. 'Ehrenfeucht Games' [6] and also Aczel's work [2]).

(ii) It corresponds to a very natural modal logic on transition systems, also introduced in [7]—'Hennessy-Milner Logic' or HML.

(iii) It seems to be the finest extensional behavioural equivalence one would want to impose—i.e., it incorporates all distinctions which could reasonably be made by external observation.

As against this, objections have been made to observation equivalence on the grounds that it goes beyond those distinctions that can really be made by an observer, and is therefore too fine; also, that it is dubious on grounds of effectiveness or constructivity. Quite a number of much coarser equivalences have been proposed, e.g., [4]. Among these, the testing equivalences of [5] have been explicitly based on a framework of extracting information about a system by testing it. These testing equivalences are indeed much weaker than observation equivalence in terms of the discriminations they are able to make. Similar comments apply to the refusal testing

* This work was partially supported by a grant from the United Kingdom Science and Engineering Research Council.
of [10], which considerably extends the power of De Nicola–Hennessy testing while remaining in the same framework, but still falls far short of observation equivalence.

This leaves open the question of the extent to which observation equivalence can be put on a testing basis, and also the profile of the terrain between testing equivalence and observation equivalence. That is, what kinds of increments to the power of testing are needed to obtain observation equivalence, if indeed this equivalence can be put on a testing basis at all.

The aim of the present paper is to provide some answers to these questions. We shall develop a notion of testing which incorporates a hierarchy of increasingly powerful constructs: traces, refusals, copying and global testing. This notion of testing is formalised in two different ways: 'denotationally', by induction on the structure of tests and using some operators on the domain of outcomes of tests; and 'operationally', via a transition system. These two definitions are shown to coincide. Then we prove our main result: the equivalence based on our notion of testing is exactly observation equivalence.

2. Transition systems

We begin by reviewing some basic ideas and notations. We shall be following [9] rather closely in the content, if not always the form, of our definitions. The reader is referred to that paper for background and motivation.

Definition 2.1. A labelled transition system (with divergence) is a structure \((Q, A, \delta, \uparrow)\) where
- \(Q\) is a set of states, agents or processes;
- \(A\) is a set of atomic actions or experiments;
- \(\delta: Q \times A \rightarrow 2^Q\) is the transition function;
and \(\uparrow \subseteq Q\) is the divergence predicate.

We write
\[ p \rightarrow^a q \equiv q \in \delta(p, a) \]

to be read "\(p\) performs \(a\) and becomes \(q\);

\[ p \uparrow \equiv p \in \uparrow \]

to be read "\(p\) may diverge";

\[ p \downarrow = \neg(p \uparrow) \]

to be read "\(p\) converges".

Two-important properties which transition systems may possess are:
(i) Image-finiteness:
\[ \forall a \in A. \forall p \in Q. \{ q \in Q \mid p \rightarrow^a q \} \text{ is finite.} \]
(ii) **Sort-finiteness:**

\[ \forall p \in Q. \{ a \in A | \exists s \in A^* \cdot p \overset{a}{\rightarrow} s \} \text{ is finite} \]

where for \( s = a_1 \ldots a_n, p \overset{s}{\rightarrow} a \equiv \exists p_1, \ldots, p_n, q. p \overset{a_1}{\rightarrow} p_1 \cdots \overset{a_n}{\rightarrow} p_n \overset{a}{\rightarrow} q \).

Now, given a transition system \((Q, A, \delta, \uparrow)\), we define a family of relations \( \sqsubseteq_n \) over \( Q \) as follows:

\[
p \sqsubseteq_n q \quad \text{always:}
\]

\[
p \sqsubseteq_{n+1} q = \forall a \in A.
\]

\[
(\forall p'. p \overset{a}{\rightarrow} p' \Rightarrow \exists q'. q \overset{a}{\rightarrow} q' & p' \sqsubseteq_n q')
\]

\[
\text{and } \ p \downarrow \Rightarrow
\]

\[
\left\{ \begin{array}{ll}
(i) & q \downarrow \& \\
(ii) & \forall q'. q \overset{a}{\rightarrow} q' \Rightarrow \exists p'. p \overset{a}{\rightarrow} p' & p' \sqsubseteq_n q' \end{array} \right.
\]

Then \( \sqsubseteq^0 \) is the *observational preorder* on \((Q, A, \delta, \uparrow)\). It is clearly reflexive and transitive; the associated equivalence,

\[
p \equiv q = p \sqsubseteq^0 q \& q \sqsubseteq^0 p
\]

is *observational equivalence*.

Given a set of actions \( A \), we now define a modal logic for talking about behaviours built from these actions. This is *Hennessy–Milner logic* (HML).

The syntax of HML is

\[
\phi ::= \top | \bot | \phi \land \psi | \phi \lor \psi | [a] \phi | (a) \phi.
\]

Given a transition system \((Q, A, \delta, \uparrow)\), we now define a satisfaction relation \( p \models \phi \) between processes \( p \in Q \) and formulas \( \phi \in \text{HML} \).

\[
p \models \top = \text{true},
\]

\[
p \models \bot = \text{false},
\]

\[
p \models \phi \land \psi = p \models \phi \text{ and } p \models \psi,
\]

\[
p \models \phi \lor \psi = p \models \phi \text{ or } p \models \psi,
\]

\[
p \models [a] \phi = p \downarrow \text{ and } \forall p'. p \overset{a}{\rightarrow} p' \Rightarrow p' \models \phi,
\]

\[
p \models (a) \phi = \exists p'. p \overset{a}{\rightarrow} p' & p' \models \phi.
\]

**Remark.** This is not exactly the definition given in [9], but it is equivalent, at least for our purposes.

**Definition.** \( \text{Aff}(p) = \{ \phi \in \text{HML} | p \models \phi \} \).

**Modal Characterisation Theorem ([9], sharpened version).** If the transition system \((Q, A, \delta, \uparrow)\) is sort-finite, then

\[
\forall p, q \in Q. \ p \sqsubseteq^0 q \Leftrightarrow \text{Aff}(p) \subseteq \text{Aff}(q).
\]
We now consider transition systems built up in a certain way, which is a straightforward generalisation of the definition of observation equivalence over CCS.

Assume we are given a basic transition system $B = (Q, A \cup \{\tau\}, \delta_0, \uparrow_0)$ where $\tau \notin A$. We now wish to regard $\tau$ as unobservable. We therefore define a derived transition system $D = (Q, A \cup \{\varepsilon, \delta, \uparrow\})$ as follows:

(i) $p \Rightarrow^a q = p \Rightarrow^{\tau \ast} \Rightarrow^a q$; $\delta(p, a) = \{q | p \Rightarrow^a q\}$

(ii) $p \Rightarrow^\varepsilon q = p \Rightarrow^{\tau \ast} q$; $\delta(p, \varepsilon) = \{q | p \Rightarrow^\varepsilon q\}$.

(iii) $p\uparrow = (\exists q. p \Rightarrow^q q \& q \uparrow) \vee p \Rightarrow^\omega$.

where

$p \Rightarrow^\omega = \exists\{p_n\}. p = p_0 \& \forall n. p_n \Rightarrow^\ast p_{n+1}$.

For the rest of this paper, we shall assume we are working over some given sort-finite basic transition system $B = (Q, A \cup \{\tau\}, \delta_0, \uparrow_0)$. We then let the observation preorder $\preceq^O$ and the HML satisfaction relation $\models$ be defined over the derived transition system $D = (Q, A \cup \{\varepsilon, \delta, \uparrow\})$. Note that sort-finiteness of $B$ implies that of $D$; hence, the Modal Characterisation Theorem holds for $D$.

Remark. All our work will thus be done in 'syntax-free' form over an arbitrary transition system. Note that the sort-finiteness condition holds for CCS (at least if mild restrictions are imposed on the renaming operation), while neither of the conditions in Milner's statement of the Modal Characterisation Theorem in [9] do.

Remark. We have followed [9] in our definitions since that paper addresses the experimental and effective content of observation equivalence with a depth and explicitness not found elsewhere in the literature. However, the approach we are about to develop could be applied to other variants of observation equivalence; see the last section of this paper.

3. Testing: denotational formulation

We begin with some preliminary notions about refusals, (cf [9]).

Definition 3.1

$p \text{ ref } a \equiv p \downarrow_0 \& p \neq^\tau \& p \neq^a$.

Lemma 3.2. If $p \Rightarrow^a$ and $p\downarrow$, then $\exists p'. p \Rightarrow^\varepsilon p' \& p' \text{ ref } a$.

Proof. Define a sequence $\{p_n\}$ as follows: $p_0 = p$. If $p_n \neq^\tau$, the sequence stops at $p_n$. Otherwise, choose $p_{n+1}$ such that $p_n \Rightarrow^\tau p_{n+1}$. If this sequence is infinite, $p \Rightarrow^\tau \omega$, contradicting $p\downarrow$. Hence, it must terminate at some $p_n$, with $p_n \neq'$. Also, $p \Rightarrow^\varepsilon p_n$ and $p\downarrow$ implies $p_n\downarrow_0$, and $p \Rightarrow^\varepsilon p_n$ and $p \Rightarrow^a$ implies $p_n \neq^a$. So $p_n \text{ ref } a$. \(\square\)
We now turn to the notion of outcome of a test. Following [5], we take outcomes of particular runs of a test to be success or failure, represented as elements of the two-point domain:

\[
\begin{array}{c}
T \\
\mathcal{O} = \\
T
\end{array}
\]

Thus this notion of 'failure' incorporates divergence. Since processes are nondeterministic, there may be many different runs of a given test on a process; hence, sets of outcomes are required to give the results of all possible runs.

We are therefore led to use a powerdomain construction [11]. There are in fact three standard powerdomain constructions: the Hoare powerdomain, incorporating may information; the Smyth powerdomain, incorporating must information and the Plotkin powerdomain, which combines both sorts of information (see, e.g., [1, 13]. For the simple case of the two-point domain, these three constructions can be represented explicitly as follows:

\[
\begin{align*}
\mathcal{P}_H[\mathcal{O}] & \quad \{T\} = \{\bot, T\} \\
\mathcal{P}_S[\mathcal{O}] & \quad \{\bot\} = \{\bot, T\} \\
\mathcal{P}_P[\mathcal{O}] & \quad \{\bot, T\}
\end{align*}
\]

It will turn out to be essential to use both may and must information to obtain a correspondence with observation equivalence, so we shall use the Plotkin powerdomain, which henceforth will be written just as \(\mathcal{P}[\mathcal{O}]\).

We now consider operations over this powerdomain. Note that for finite domains, monotone functions are automatically continuous and computable.

1. Linear operations

Given any monotone function \(f: \mathcal{O}^n \to \mathcal{O} (n \geq 0)\), there is the pointwise extension \(f': (\mathcal{P}[\mathcal{O}])^n \to (\mathcal{P}[\mathcal{O}])\) defined by

\[
f'(X_1, \ldots, X_n) = \{f(x_1, \ldots, x_n) | x_i \in X_i, 1 \leq i \leq n\}.
\]

This is obviously monotone. Such operations are said to be (multi)linear because they preserve union in each argument separately:

\[
f'(X_1, \ldots, X_i \cup X_i'', \ldots, X_n) = f'(X_1, \ldots, X'_i, \ldots, X_n) \cup f'(X_1, \ldots, X''_i, \ldots, X_n).
\]
We shall be interested in two such operations in particular, those arising from the binary operations \( \land, \lor \) on \( 0 \) defined as in the tables of Fig. 1.

By abuse of notation, we shall write the pointwise extensions of these operations as \( \land, \lor \) also. These have the tables shown in Fig. 2.

2. Nonlinear operations

More generally, we can consider monotone (but not necessarily linear) operations \( \phi : P[0]^n \rightarrow P[0] \). We will only require two unary nonlinear operations shown in Fig. 3.

These functions are clearly monotone. We have denoted them by the quantifier symbols because they can be read as some and all, i.e., 'some run of the experiment succeeds' and 'all runs of the experiment succeed'. We could also read them as may and must, since \( \exists \) is the closure which picks out \( P_\exists[0] \) as a subdomain of \( P_[0] \) (i.e., ignoring must information and only retaining may information) while \( \forall \) is the projection which picks out \( P_\forall[0] \) as a subdomain of \( P_[0] \) (ignoring may information and only retaining must information).

We are now ready to introduce the syntax of tests. We have a syntactic category \( T \) of test expressions, ranged over by \( t \).

\[
t ::= \text{Succ} | \text{Fail} | \text{at} | \diamond t | t_1 \land t_2 | t_1 \lor t_2 | \forall t | \exists t.
\]

For the 'semantics' of tests, we define a function \( O : T \times Q \rightarrow P[0] \) such that \( O(t, p) \)
is the set of possible outcomes of a test $t$ performed on a process $p$.

\[
O(\text{SUCC}, p) = \{\top\},
\]

\[
O(\text{FAIL}, p) = \{\bot\},
\]

\[
O(at, p) = \bigcup \{ O(t, p') \mid p \Rightarrow^a p' \}
\]
\[
\cup \{ \bot \mid p \uparrow \} \cup \{ \bot \mid p \Rightarrow^i p' \& p' \text{ ref a} \},
\]

\[
O(\tilde{a}t, p) = \bigcup \{ O(t, p') \mid p \Rightarrow^a p' \}
\]
\[
\cup \{ \bot \mid p \uparrow \} \cup \{ \top \mid p \Rightarrow^i p' \& p' \text{ ref a} \},
\]

\[
O(\varepsilon t, p) = \bigcup \{ O(t, p') \mid p \Rightarrow^a p' \} \cup \{ \bot \mid p \uparrow \},
\]

\[
O(t_1 \land t_2, p) = O(t_1, p) \land O(t_2, p),
\]

\[
O(t_1 \lor t_2, p) = O(t_1, p) \lor O(t_2, p),
\]

\[
O(\forall t, p) = \forall O(t, p),
\]

\[
O(\exists t, p) = \exists O(t, p).
\]

**Remark.** Lemma 3.2 is needed to show that the clauses for $at, \tilde{a}t$ are well-defined.

We now provide some discussion of these testing contracts, which fall naturally into five groups:

1. **Traces:** SUCC, FAIL, $at$. This fragment of the language of tests just comprises strings of the form $a_1 \ldots a_n \text{SUCC}$ or $a_1 \ldots a_n \text{FAIL}$ ($n \geq 0$). However, even here there is more to testing than just the set of traces in the sense of ‘the language accepted by the process’ (which would correspond to using the Hoare powerdomain for our domain of outcomes). For example, we can distinguish $a(b\text{NIL} + c\text{NIL})$ from $ab\text{NIL} + ac\text{NIL}$ (cf. [8]) since

\[
O(ab\text{SUCC}, a(b\text{NIL} + c\text{NIL})) = \{\top\},
\]

\[
O(ab\text{SUCC}, ab\text{NIL} + ac\text{NIL}) = \{\bot, \top\}.
\]

2. **Refusals:** $\tilde{a}t$ (cf. [10]). The addition of refusals allows the experimenter to use *finitely obtainable information* about the failure to perform some action. (In terms of the discussion in [9], this is ‘the green light going off after we press the $a$-button’.) The definition of $p \text{ ref } a$ ensures that $p$ determinately refuses $a$, and no possible future actions or ‘improvement of information’ can change this fact. It should be up to the experimenter whether this state of affairs comprises success or failure; hence the introduction of $\tilde{a}t$ alongside $at$.

3. **Epsilon:** $\varepsilon t$. This corresponds to a form of nondeterminism in the test, of a kind already present in De Nicola–Hennessy testing (e.g., with $\varepsilon t = \mu x. t + \tau x$). However, we allow the nondeterminacy to be bounded by the process in the sense that if the process is convergent, and hence must eventually reach a stable state, then we will eventually proceed to test $t$. In terms of [9], this could be implemented
by the following 'procedure'. Flip a coin. If the green light has gone off, or the coin falls heads, proceed to the remainder of the test \( t \). Otherwise, repeat.

(4) Copying: \( t_1 \land t_2 \), \( t_1 \lor t_2 \). De Nicola–Hennessy testing implies the ability to make copies of the machine in its initial state. If we extend this idea to allow copies to be made at intermediate points in a test, then we need some idea of performing separate tests on the copies, and then combining the information from the outcomes of the subtests in some way to obtain an outcome of the overall test. We formalise this in terms of monotone operators \( f: \mathcal{O}^n \rightarrow \mathcal{O} \) (monotonicity guaranteeing that we combine information in a computable fashion), which are then pointwise extended to \( f^*: P[\mathcal{O}]^n \rightarrow P[\mathcal{O}] \) where the multilinearity corresponds to the intuition that each subtest is independent. This seems to be the simplest model of pure copying, and at the same time a rather comprehensive and mathematically natural one. Our choice of the particular operators \( \land, \lor \) is motivated by the desire to have a minimal complete set; this point will be taken up in Section 5.

(5) Global testing: \( \forall t, \exists t \). The use of linear operators as in (4) preserves the local character of De Nicola–Hennessy testing, and also of tests based on (1)–(3). That is, information is obtained independently on each possible run of the test, and only collected as a whole 'at the end'. This is not sufficient for observation equivalence. We need to be able, at some point inside a test, to enumerate all runs of some subtest. The information gathered from this enumeration can still be combined in a computable way; this is the force of the monotonicity of \( \forall \) and \( \exists \). However, the ability to do the enumeration in the first place may be taken as objectionable, not so much on the grounds of effectiveness, as the results in [9] and those in the present paper show, as on the grounds that we are in some way making what should be unobservable observable.

In more detail, what seems to be required is the ability to enumerate all (of finitely many) possible 'operating environments' at each stage of the test, so as to guarantee that all nondeterministic branches will be pursued by various copies of the subject process. Information about the various outcomes of the test on these copies can then be combined by a standard dovetailing construction. This is of course exactly the point of the 'weathers' in [9].

The extent to which global testing is really acceptable clearly needs further discussion. The present paper aims to place the cards on the table as a basis for such a discussion. Our development to date already suggests quite a few possibilities for imposing a coherent structure on testing, with many natural intermediate points, e.g.:

(i) traces: (1),
(ii) traces + refusals: (1) + (2) (+ 'duals'—cf. [10]),
(iii) traces + pure copying: (1) + (4),
(iv) traces + refusal + pure copying: (1) + (2) + (4),

etc. Our results in Section 5 will give strong evidence that our notion of testing is at the top of the lattice of reasonable testing notions since it suffices to characterise observational equivalence, and is also 'operator complete'.

\[\text{Theorem 1:}\]
\[\forall t, \exists t \text{ such that } f^*: P[\mathcal{O}]^n \rightarrow P[\mathcal{O}] \text{ where the multilinearity corresponds to the intuition that each subtest is independent. This seems to be the simplest model of pure copying, and at the same time a rather comprehensive and mathematically natural one. Our choice of the particular operators } \land, \lor \text{ is motivated by the desire to have a minimal complete set; this point will be taken up in Section 5.}\]
Observation equivalence as a testing equivalence

4. Testing: operational formulation

In this section, we shall introduce a transition system to describe how the evaluation of test outcomes may be implemented in a step-by-step fashion. This will also make the computational intuitions appealed to in the previous section somewhat more explicit.

We shall be making two additional assumptions on the basic transition system \( B \) we are working over:

(i) \( B \) is image-finite;
(ii) \( \uparrow_0 = \emptyset \), i.e., \( p \uparrow \) (in the derived transition system \( D \)) iff \( p \rightarrow^\omega \).

Of these, (i) appears essential if we are to keep the system on a finitary basis (and is made in \([9]\)). However, (ii) is merely a minor technical convenience to keep the definitions a little simpler.

The configurations or states of the transition system we shall specify will be experiment expressions, with the following syntax:

\[
E ::= T | \bot | (t|p)| E_1 \land E_2 | E_1 \lor E_2 | \forall E | \exists E
\]

Here \( T, \bot \) represent finite outcomes of tests; \( t|p \), for \( t \) a test, \( p \) a process, represents 'application' of a test to a process (the notation follows \([5]\) and \([10]\) here); \( E_1 \land E_2 \), \( E_1 \lor E_2 \), \( \forall E \), \( \exists E \) represent the evaluation of operator over the (outcomes of) subexperiments.

We shall define a transition relation \( \rightarrow \) over experiment expressions. This relation will have the following important properties:

(i) \( \rightarrow \) is image-finite, i.e., \( \forall E. \{ E' | E \rightarrow E' \} \) is finite,
(ii) \( E \Leftrightarrow \Leftrightarrow E = \top \lor E = \bot \).

Definition. \( \delta_1(E) \equiv \{ E' | E \rightarrow E' \} \).

The transition relation \( \rightarrow \) is defined to be the least satisfying the following axioms and rules:

(1) \( \text{Succ} | p \rightarrow \top \)
(2) \( \text{Fail} | p \rightarrow \bot \)

(3) (i) \( \frac{p \rightarrow^* p'}{at \mid p \rightarrow at \mid p'} \)
(ii) \( \frac{p \rightarrow^* p'}{at \mid p \rightarrow t \mid p'} \)
(iii) \( \frac{p \text{ ref } \omega}{at \mid p \rightarrow \bot} \)

(4) (i) \( \frac{p \rightarrow^* p'}{at \mid p \rightarrow at \mid p'} \)
(ii) \( \frac{p \rightarrow^* p'}{at \mid p \rightarrow t \mid p'} \)
(iii) \( \frac{p \text{ ref } a}{at \mid p \rightarrow \top} \)

(5) (i) \( \frac{p \rightarrow^* p'}{\varepsilon t \mid p \rightarrow \varepsilon t \mid p'} \)
(ii) \( \varepsilon t \mid p \rightarrow t \mid p \)

(6) (i) \( t_1 \land t_2 \mid p \rightarrow t_1 \mid p \land t_2 \mid p \)
(ii) \( E_1 \rightarrow E_1' \)
\( E_2 \rightarrow E_2' \)
\( E_1 \land E_2 \rightarrow E_1' \land E_2' \)
(iii) $T \land E \rightarrow E$  
(iv) $\bot \land E \rightarrow \bot$  
(v) $E \land T \rightarrow E$  
(vi) $E \land \bot \rightarrow \bot$

(7)  
(i) $t_1 \lor t_2 \big| p \rightarrow t_1 \big| p \lor t_2 \big| p$  
(ii) $E_1 \rightarrow E'_1 \quad E_2 \rightarrow E'_2 \quad \frac{E_1 \lor E_2 \rightarrow E'_1 \lor E'_2}{E_1 \lor E_2 \rightarrow E'_1 \lor E'_2}$

(iii) $T \lor E \rightarrow T$  
(iv) $\bot \lor E \rightarrow E$  
(v) $E \lor T \rightarrow T$  
(vi) $E \lor \bot \rightarrow E$

(8)  
(i) $\forall t \big| p \rightarrow \forall (t \big| p)$  
(ii) $\forall \bot \rightarrow \bot$  
(iii) $\forall T \rightarrow T$

(iv) $\delta_i(E) = \{E_1, \ldots, E_n\}$  
$\forall E \rightarrow \land_{i=1}^n \forall E_i$

(9)  
(i) $\exists t \big| p \rightarrow \exists (t \big| p)$  
(ii) $\exists \bot \rightarrow \bot$  
(iii) $\exists T \rightarrow T$

(iv) $\delta_i(E) = \{E_1, \ldots, E_n\}$  
$\exists E \rightarrow \lor_{i=1}^n \exists E_i$

We can now use this transition system to define the set of results of evaluating a test on a process.

**Definition.** $R(t, p) = \{T \mid (t \big| p) \rightarrow^* T\} \cup \{T \mid (t \big| p) \rightarrow^* \bot\} \cup \{\bot \mid (t \big| p) \rightarrow^\omega\}$, where $E \rightarrow^\omega = \exists \{E_n\}. E = E_0 \& \forall n. E_n \rightarrow E_{n+1}$.

As expected, we have the following theorem.

**Theorem 4.1.** $\forall t \in T, p \in Q. O(t, p) = R(t, p)$.

**Proof.** By induction on $t$. The cases for SUCC, FAIL are trivial. For $t = \alpha t_1$, note that

$R(t, p) = \bigcup \{R(t_1, p') \mid p \rightarrow^* \rightarrow^\alpha p'\}$

$\cup \{\bot \mid p \rightarrow^* p' \& p' \text{ ref } a\} \cup \{\bot \mid p \rightarrow^\omega\}$

$= \bigcup \{O(t_1, p') \mid p \Rightarrow^\alpha p'\}$

$\cup \{\bot \mid p \Rightarrow^\alpha p' \& p' \text{ ref } a\} \cup \{\bot \mid p\}$

by induction hypothesis

$= O(\alpha t_1, p)$.

The cases for $\alpha t_1, \varepsilon t_1$ are similar.
For $t = t_1 \land t_2$, note that
\[
R(t, p) = \{T | (t_1, p) \rightarrow^* T \land (t_2, p) \rightarrow^* T \}
\cup \{\perp | (t_1, p) \rightarrow^* \perp \lor (t_2, p) \rightarrow^* \perp \}
\cup \{\perp | (t_1, p) \rightarrow^w \lor (t_2, p) \rightarrow^w \}
= \{T | T \in O(t_1, p) \land T \in O(t_2, p) \}
\cup \{\perp | \perp \in O(t_2, p) \lor \perp \in O(t_2, p) \}
\]
by induction hypothesis
\[
= O(t_1 \land t_2, p).
\]
The case for $t_1 \lor t_2$ is similar.
For $t = \forall t_1$, note that:
\[
R(t, p) = \{T | \text{all computations of } t_1, p \text{ end in } T \}
\cup \{\perp | (t_1, p) \rightarrow^* \perp \}
\cup \{\perp | (t_1, p) \rightarrow^w \}
= \{T | R(t_1, p) = \{T \}\}
\cup \{\perp | \perp \in R(t_1, p) \}
= \{T | O(t_1, p) = \{T \}\}
\cup \{\perp | \perp \in O(t_1, p) \}
\]
(by induction hypothesis)
\[
= O(\forall t_1, p).
\]
The case for $\exists t_1$ is similar.

5. Results

In this section we establish our main results which show that observation equivalence, as defined in Section 2, corresponds exactly to indistinguishability under testing.

Definition 5.1. $T'$ is the operator completion of $T$ defined by adding, for each monotone function $op: P[O]^n \rightarrow P[O]$ ($n \geq 0$), a construct $OP(t_1, \ldots, t_n)$ to the syntax of tests. $O$ is then extended by
\[
O(\text{OP}(t_1, \ldots, t_n), p) = \text{op}(O(t_1, p), \ldots, O(t_n, p)).
\]
Definition 5.2. The depth of a test \( t \in T' \), denoted \( d \), is defined as follows:

\[
\begin{align*}
    d(\text{SUCC}) &= d(\text{FAIL}) = 0, \\
    d(at) &= d(\tilde{a}t) = d(\varepsilon t) = 1 + d(t), \\
    d(t_1 \land t_2) &= d(t_1 \lor t_2) = \max\{d(t_1), d(t_2)\}, \\
    d(\forall t) &= d(\exists t) = d(t), \\
    d(\text{OP}(t_1, \ldots, t_n)) &= \max\{d(t_i) | 1 \leq i \leq n\}.
\end{align*}
\]

Lemma 5.3. \( \forall t \in T' \forall n > d(t). \forall p, q \in Q. \)

\[ p \leq_n q \Rightarrow O(t, p) \subseteq O(t, q). \]

Proof. By induction on the structure of \( t \). The cases for SUCC, FAIL are trivial. For cases of the form \( t = \text{OP}(t_1, \ldots, t_n) \) (including \( t_1 \land t_2, t_1 \lor t_2, \forall t_1, \exists t_1 \)), by induction hypothesis, \( O(t_i, p) \subseteq O(t_i, q), 1 < i < n \); hence, by monotonicity of \( \text{op} \), \( O(\text{OP}(t_1, \ldots, t_n), p) = \text{op}(O(t_1, p), \ldots, O(t_n, p)) \subseteq \text{op}(O(t_1, q), \ldots, O(t_n, q)) = O(\text{OP}(t_1, \ldots, t_n), q) \).

For \( t = \tilde{a}t_1 \), suppose \( T \in O(\tilde{a}t_1, p) \). Then, either

(i) \( p \Rightarrow^a p' \) and \( T \subseteq O(t_1, p') \), or

(ii) \( p \Rightarrow^f p' \) and \( p' \text{ ref } a \).

In case (i), \( 0 < d(t_1) < n \) and \( p \leq_n q \) implies

\[ q \Rightarrow^a q' \quad \text{and} \quad p' \leq_{n-1} q', \quad \text{where } d(t_1) < n - 1. \]

By induction hypothesis, \( O(t_1, p') \subseteq O(t_1, q') \); hence \( T \subseteq O(t_1, q') \), and so \( T \subseteq O(\tilde{a}t_1, q) \).

In case (ii), \( 0 < d(t_1) < n \) and \( p \leq_n q \) implies \( q \Rightarrow^a q' \) and \( p' \leq_{n-1} q' \), with \( 0 < n - 1 \). Then \( p' \text{ ref } a \) implies \( p' \downarrow \) implies (I): \( q' \downarrow \), and (II): \( p' \Rightarrow^a \) implies \( q' \Rightarrow^a \). Hence, by Lemma 3.2, \( \exists q'' \). \( q' \Rightarrow^f q'' \) and \( q'' \text{ ref } a \). Thus, \( q \Rightarrow^f q'' \) and \( q'' \text{ ref } a \); hence, \( T \subseteq O(\tilde{a}t_1, q) \) in this case, too.

Now, suppose \( \bot \in O(\tilde{a}t_1, q) \). Then, either (i): \( p \uparrow \), in which case \( \bot \in O(\tilde{a}t_1, p) \), or (ii): \( p \downarrow \). In case (ii), \( p \leq_n q \) implies \( q \downarrow \). So, in this case, \( \bot \in O(\tilde{a}t_1, q) \) implies \( q \Rightarrow^a q' \) and \( \bot \in O(t_1, q') \). But then \( p \downarrow \) implies \( p \Rightarrow^a p' \) and \( p' \leq_{n-1} q' \). By induction hypothesis, \( O(t_1, p') \subseteq O(t_1, q') \), and so \( \bot \in O(t_1, p') \), and so \( \bot \in O(\tilde{a}t_1, p) \). The cases for \( at \) and \( \varepsilon t \) are similar to \( \tilde{a}t \). \( \square \)

Corollary 5.4. \( p \leq^\circ q \Rightarrow \forall t \in T'. O(t, p) \subseteq O(t, q). \)

This lemma tells us that, whatever monotone operators we add to our language of tests, we shall still not go beyond the observation preorder. In fact, a similar argument would show that we could add arbitrary functions \( f : P[O]^a \rightarrow P[O] \) (not necessarily monotone), and still preserve observation equivalence. This gives further evidence for the importance of the preorder in filtering out computationally unrealistic notions.

We now define a translation from HML formulas to tests.
Definition 5.5

$(-)^* : \text{HML} \rightarrow T$,

$(T)^* = \text{Succ}, \quad (F)^* = \text{Fail},$

$(\phi \land \chi)^* = (\phi)^* \land (\chi)^*, \quad (\phi \lor \chi)^* = (\phi)^* \lor (\chi)^*,$

$([a]\phi)^* = \forall \alpha(\phi)^*, \quad ((a)\phi)^* = \exists a(\phi)^*,$

$([\varepsilon]\phi)^* = \forall \varepsilon(\phi)^*, \quad ((\varepsilon)\phi)^* = \exists \varepsilon(\phi).$

Lemma 5.6. $\forall \phi \in \text{HML}. \forall p \in Q.$

$p \models \phi \iff O((\phi)^*, p) = \{T\}$

and

$p \not\models \phi \iff O((\phi)^*, p) = \{\bot\}.$

Proof. By induction on the structure of $\phi$. The cases for $T, \bot$ are trivial. For $\phi \land \chi, \phi \lor \chi$ note that we are coding truth of the satisfaction relation by $\{T\}$ and falsity by $\{\bot\}$ and $\land, \lor$ have their standard truth-table definitions on these singleton sets.

Case $[a]\phi$:

(i) $p \models [a]\phi \iff p \downarrow \land \forall p'. p \Rightarrow^a p' \Rightarrow p' \models \phi$

$\iff p \downarrow \land \forall p'. p \Rightarrow^a p' \Rightarrow O((\phi)^*, p) = \{T\}$

(by the induction hypothesis)

$\iff O(\alpha(\phi)^*, p) = \{T\}$

$\iff \forall O(\alpha(\phi)^*, p) = \{T\}$

$\iff O(([a]\phi)^*, p) = \{T\}.$

(ii) $p \not\models [a]\phi \iff p \uparrow \lor \exists p'. p \Rightarrow^a p' \land p' \not\models \phi$

$\iff p \uparrow \lor \exists p'. p \Rightarrow^a p' \land O((\phi)^*, p') = \{\bot\}$

(by the induction hypothesis)

$\iff \bot \in O(\alpha(\phi)^*, p)$

$\iff \forall O(\alpha(\phi)^*, p) = \{\bot\}$

$\iff O(([a]\phi)^*, p) = \{\bot\}.$

Case $(a)\phi$:

(i) $p \models (a)\phi \iff \exists p'. p \Rightarrow^a p' \land p' \models \phi$

$\iff \exists p'. p \Rightarrow^a p' \land O((\phi)^*, p') = \{T\}$

$\iff O(\alpha(\phi)^*, p) = \{T\}$

$\iff \forall O(\alpha(\phi)^*, p) = \{\bot\}$

$\iff O(([a]\phi)^*, p) = \{\bot\}.$
(by the induction hypothesis)
\[ T \in O(a(\phi)^*, p') \]
\[ \exists O(a(\phi)^*, p) = \{ T \} \]
\[ O(((a)\phi)^*, p) = \{ T \}. \]

(ii) \[ p \not\models (a)\phi \iff \forall p'. p \models^a p' \Rightarrow p' \not\models \phi \]
\[ \forall p'. p \models^a p' \Rightarrow O((\phi)^*, p') = \{ \bot \} \]

(by the induction hypothesis)
\[ \Leftrightarrow O(a(\phi)^*, p) = \{ \bot \} \]
\[ \Leftrightarrow \exists O(a(\phi)^*, p) = \{ \bot \} \]
\[ \Leftrightarrow O(((a)\phi)^*, p) = \{ \bot \}. \]

The cases for \([\varepsilon]\phi, (\varepsilon)\phi\) are similar. \(\square\)

**Definition 5.7.** The testing preorder is defined as follows
\[ p \sqsubseteq^T q \iff \forall t \in T. \; O(t, p) \sqsubseteq O(t, q). \]

We are now ready to prove our main result.

**Theorem 5.8 (Characterisation Theorem)**
\[ \forall p, q \in Q. \; p \sqsubseteq^T q \iff p \sqsubseteq^\omega q. \]

**Proof.** By Corollary 5.4, \(p \sqsubseteq^\omega q \Rightarrow p \sqsubseteq^T q.\)
For the converse,
\[ p \sqsubseteq^T q \Rightarrow \forall \phi \in \text{HML}. \; O((\phi)^*, p) \sqsubseteq O((\phi)^*, q) \]
\[ \Rightarrow \text{Aff}(p) \subseteq \text{Aff}(q) \quad \text{by Lemma 5.6.} \]
\[ \Rightarrow p \sqsubseteq^\omega q \quad \text{by the Modal Characterisation Theorem.} \quad \square \]

**Theorem 5.9 (Operator Completeness of \(T\)).** Extension of \(T\) by any set of monotone operators does not increase the power of testing since
\[ p \sqsubseteq^T q \iff p \sqsubseteq^T q. \]

**Proof.** Clearly, \(p \sqsubseteq^T q \Rightarrow p \sqsubseteq^T q.\) Furthermore,
\[ p \sqsubseteq^T q \Rightarrow p \sqsubseteq^\omega q \quad \text{by Theorem 5.8} \]
\[ \Rightarrow p \sqsubseteq^T q \quad \text{by Corollary 5.4.} \quad \square \]
6. Alternative definitions of observation equivalence

Our definitions in Section 2 followed [9] closely. In particular, we defined observable action by

\( \Rightarrow^a q \equiv p \Rightarrow^* \Rightarrow^a q \)

instead of the more usual

\( \Rightarrow^a q \equiv p \Rightarrow^* \Rightarrow^a q \).

Our aim in this section is to show that we can develop just as satisfactory a treatment of testing based on (**) as we did for (*), provided that we modify the definition of divergence appropriately.

It will be convenient for this purpose to use a more refined notion of divergence in transition systems, also described in [9]. A system \( (Q, A, \delta, \uparrow) \) is a transition system with local divergence if \( \uparrow \subseteq Q \times A \) is a binary relation between processes and actions. We write \( p \uparrow a \) for \((p, a) \in \uparrow\), and \( p \downarrow a \) for \( \neg(p \uparrow a) \). We modify the definition of the observation preorder at the inductive step to:

\[
p \subseteq_{n+1} q \equiv \forall a \in A. \quad (\forall p'. p \Rightarrow^a p' \Rightarrow^a q' \Rightarrow^a q \land p \subseteq_n q') \land p \downarrow a \Rightarrow
\]

\( i \) \( q \downarrow a \land \)

\( ii) \ \forall q'. q \Rightarrow^a q' \Rightarrow^a q \Rightarrow^a p'. p \Rightarrow^a p' \land p' \subseteq_n q'). \)

The definition of the satisfaction relation for HML has the clause for \([a] \phi \) amended to

\[
p \models [a] \phi = p \downarrow a \land \forall p'. p \Rightarrow^a p' \Rightarrow p' \models \phi.
\]

The Modal Characterisation Theorem still holds with these modified definitions.

Now given a basic transition system \( B = (Q, A \cup \{ \tau \}, \delta_0, \uparrow_0) \) (with global divergence) as before, we define the derived system \( D = (Q, A \cup \{ \varepsilon \}, \delta, \uparrow) \) as follows:

\[
p \Rightarrow^\varepsilon q = p \Rightarrow^* q,
\]

\[
p \Rightarrow^a q \equiv p \Rightarrow^* \Rightarrow^a \Rightarrow^* q,
\]

\[
p \uparrow \varepsilon = (\exists p'. p \Rightarrow^\varepsilon p' \land p \uparrow_0) \lor p \Rightarrow^\omega,
\]

\[
p \uparrow a = p \uparrow \varepsilon \lor (\exists p'. p \Rightarrow^a p' \land p' \uparrow \varepsilon).
\]

(This interpretation is not the one mentioned in [9]; a special case—no divergence in the basic system—appears in [12].)
The denotational formulation of testing in this setting only requires amendment to the clauses for $O(at, p)$, $O(\hat{a}t, p)$, $O(\varepsilon t, p)$. These become

$$O(at, p) = \bigcup \{O(t, p') | p \Rightarrow^a p'\}$$

$$\cup \{\bot | p \uparrow a\} \cup \{\bot | p \Rightarrow^\varepsilon p' \& p \text{ ref } a\},$$

$$O(\hat{a}t, p) = \bigcup \{O(t, p') | p \Rightarrow^a p'\}$$

$$\cup \{\bot | p \uparrow a\} \cup \{\top | p \Rightarrow^\varepsilon p' \& p \text{ ref } a\},$$

$$O(\varepsilon t, p) = \bigcup \{O(t, p') | p \Rightarrow^\varepsilon p'\} \cup \{\bot | p \uparrow \varepsilon\}.$$

The operational formulation of testing only requires amendment to rules (3)(ii) and (4)(ii) in the definition of the transition system. These become

$$\frac{p \Rightarrow^a p'}{at | p \Rightarrow \varepsilon t | p'}$$

$$\frac{p \Rightarrow^a p'}{\hat{a}t | p \Rightarrow \varepsilon t | p'}$$

respectively.

Theorem 4.1 and all the results in Section 5 still hold under these amended definitions, with only routine modifications to the proofs.

Why did we introduce the more refined notion of divergence $p \uparrow a$? A more straightforward approach would have been to simply change the definition of $p \Rightarrow^a q$ to (**), while keeping the global notion of divergence unchanged. The problem with this 'straightforward approach' is that it smuggles in a hidden fairness assumption, which destroys the effective character of testing. To clarify this point, we need some general notions.

We can formulate two basic axioms for testing:

(A1) If a test succeeds, it must do so in finite time (thus excluding, e.g., testing for divergence).

(A2) Only a finite amount of information about the subject process can be elicited by a test in a finite period of time (excluding, e.g., 'Zeno testing').

From these axioms, we can derive the following principle:

(P) If a test $t$ succeeds on a process $p$, it must do so on the basis of a finite amount of information about $p$.

This can be formulated in more mathematical terms as

(C) Tests are continuous in their process arguments.

It can in fact be shown that the notions of testing developed in this paper are continuous in a precise sense; we shall not elaborate on that here. However, we can make the point that the naive approach to interpreting $\Rightarrow^a$ by (**) leads to an evidently discontinuous notion of testing (and thus violates (A1) or (A2)). This can be illustrated by the following example.
Example. Define $p_0 = \Omega$, $p_{n+1} = c\text{NIL} + \tau p_n$ (in CCS with 'partial terms', cf. e.g., [5, 10]). Then the sequence $\{p_n\}$ forms a chain in the syntactic ordering, with $\sqcup_n a p_n = a(\sqcup_n p_n) = a p = a(\mu x. c\text{NIL} + \tau x)$.

Now, consider the test $t = \forall \alpha \exists \epsilon \text{SUC}C$ (i.e., the translation of the HML formula $[a](c)\top$). Under the definitions of the naive approach,

$$\forall n. O(t, a p_n) = \{\bot\} \quad \text{but} \quad O(t, a p) = \{\top\}.$$

Note that with our original definitions based on (*), $O(t, a p_n) = \{\top\}$ ($n \geq 1$), while with the modified definitions given at the start of this section, $O(t, a p) = \{\bot\}$.

As a further indication of the mismatch of concepts in the naive approach, note that under it we have $p \not\equiv [\epsilon](c)\top$ (since $p \uparrow$), but $a p \models [a](c)\top$ (since $a \uparrow$) while $a p \not\equiv [a][\epsilon](c)\top$ (since $p \uparrow$); and hence $[a][\epsilon]\phi$ is not in general equivalent to $[a]\phi$, which we would expect with (**) in force.

Note that with the definitions given at the start of this Section, this equivalence does indeed hold.

Acknowledgment

I would like to thank Iain Phillips for many stimulating discussions on testing and concurrency.

References