On Observing Non-determinism and Concurrency

Matthew Hennessy and Robin Milner
University of Edinburgh

1. Introduction

The denotational approach to the semantics of programming languages has been well-developed in recent years (see [1],[6]) and applied successfully to many non-trivial languages. Even languages with parallel constructs have been treated in this way, using the powerdomain constructions of [2],[5],[7]. Indeed for such languages there is no shortage of possible denotational models. For example there are several simple variations on the model for processes, introduced in [3].

Faced with such an abundance it is best to recall the motivation for seeking such models. They provide a useful mathematical framework for the analysis of programs, and for developing logical systems for proving their properties. However if either the mathematics or the logic is to have any relevance a link must be made between the denotational model and the behaviour, or operational semantics, of the programs

What exactly is meant by the behaviour of non-deterministic or concurrent programs is far from clear and in this paper we put forth one possible definition. The essence of our approach is that the behaviour of a program is determined by how it communicates with an observer. We begin by assuming that every program action is observable in this way; later we allow that some actions (in particular, internal communications between concurrent components) are not observable.

We apply our definition to a sequence of simple languages for expressing finite behaviours, and show that in each case it can be characterised by algebraic axioms. This leads automatically to fully abstract models. Moreover, a proper understanding of the finite case seems a necessary prelude to a study of programs with infinite behaviour. Such programs may be gained simply by adding recursion to our languages.

In fact with the addition of recursion, and with a natural extension to allow data values to be communicated between concurrently active agents, the simple algebra described here becomes a language for writing a specifying concurrent programs and for proving their properties. This language was introduced in [4]; it was partly the need for a firm basis for the algebraic laws discussed there which led to the present study of observation equivalence.

In section 2 we present our general framework. In section 3 we outline and summarize our results for the languages considered. The proofs will appear in a technical report.
2. Observational Equivalence of Processes

Let us start with the idea that two programs are operationally congruent if it is impossible to distinguish between them operationally. Thus, if they are considered as modules, one can be exchanged for the other in a larger program, without affecting the behaviour of the latter.

In the deterministic case the behaviour of a program is usually taken to mean its input-output relation. To summarise: two programs p,q are behaviourally equivalent, written p ∼ q, if IO(p) = IO(q); the operational congruence generated by ∼ is defined by p ∼ q if for every suitable programming environment E[ż1],ż1[paration]∼ż1[queries]. If the language is defined algebraically, i.e. by operations for constructing new programs from ones already defined, ∼ is the largest congruence contained in ∼.

However, any satisfactory comparison of the behaviour of concurrent programs must take into account their intermediate states as they progress through a computation, because differing intermediate states can be exploited in larger programming environments to produce different overall behaviour (e.g. deadlock). With this in mind, we now outline a more general notion of behaviour equivalence between programs, called observational equivalence.

Let P be a set of objects called programs, or agents, which are capable of some form of communication. An atomic experiment on p can be considered as an attempt to communicate with p. Since the act of communication can change the nature of an agent, and can change it in various different ways depending on its internal structure, the effect of an atomic experiment can be captured by a binary relation over P. Since in general there may be various means of communication we have a set of relations (Ři ∈ P × P, i ∈ I). Using these atomic experiments, we define a sequence of equivalence relations ∼n over P as follows:

Let

p ∼0 q if \( p,q \in P \)

p ∼n+1 q if

i) \( \forall i \in I, \langle p,p' \rangle \in \tilde{R}_i \) implies \( \exists q', \langle q,q' \rangle \in \tilde{R}_i \) \( \implies q' \sim_n q \)

and

ii) \( \forall i \in I, \langle q,q' \rangle \in \tilde{R}_i \) implies \( \exists p', \langle p,p' \rangle \in \tilde{R}_i \) \( \implies p' \sim_n q' \)

Then p is observationally equivalent to q, written p ∼ q, if p ∼ q for every n. Before discussing ∼ we give some of its properties. For any S ⊆ P × P let E(S) be defined by

\[ \langle p,q \rangle \in E(S) \text{ if } \forall i \in I \]

i) \( \langle p,p' \rangle \in \tilde{R}_i \) implies \( \exists q', \langle q,q' \rangle \in \tilde{R}_i \) \( \implies q' \in S \)

ii) \( \langle q,q' \rangle \in \tilde{R}_i \) implies \( \exists p', \langle p,p' \rangle \in \tilde{R}_i \) \( \implies p' \in S \)

We say that a relation R is image-finite if for each p, \( \{ p' | \langle p,p' \rangle \in R \} \) is finite.

Theorem 2.1

If each \( \tilde{R}_i \) is image-finite then ∼ is the maximal solution to \( S = E(S) \).
More complicated experiments can be carried out by applying sequences of atomic experiments. Let $s$ be the sequence $i_1, \ldots, i_n$ in $I(n \geq 1)$. An $s$-experiment on $p$ is a sequence $p_1, \ldots, p_{n+1}$ where $p_1 = p$ and $p_k \vdash p_{k+1}$ for constructing new knowledge contained in $\vdash$.

Concurrent programs progress through a computation in larger programming stage. With this in mind, we define a sequence $q \vdash q'$ where $q' = \vdash q$ for every $n$. We consider as an internal relation $\vdash$ that changes the nature of the language on its interpretation.

Let that language $\mathcal{L}$ of formulae be the least set such that

1. $\forall \alpha \in \mathcal{L}$
2. $\forall \alpha \in \mathcal{L}$
3. $\forall \alpha \in \mathcal{L}$

The satisfaction relation $| \vdash p \in \mathcal{L}$ is the least relation such that

1. $p \vdash \top$ for all $p \in P$.
2. $p \vdash A \land B$ iff $p \vdash A$ and $p \vdash B$.
3. $p \vdash \forall \alpha$ if and only if $p \vdash \alpha$.
4. $p \vdash \Box \alpha$ iff for some $i$-experiment $<p,p'>$, $p' \vdash \alpha$.

For convenience let $F$ stand for $\top \lor B$

$\top \lor B$ is finite.

We say $p$ is $s$-deadlocked if there are no $s$-experiments on $p$.

**Examples**

a) $p \vdash \Box \top$ - it is possible to carry out an $s$-experiment on $p$.
b) $p \vdash \Box F \vdash \top$ is $s$-deadlocked.
c) $p \vdash \Box (F \lor \Box F)$ - it is possible, via an $s_1$-experiment, to get into a state which is either $s_2$-deadlocked or $s_3$-deadlocked.
d) $p \vdash s_1$ $(\Box s_2 F \lor s_3 F)$ - at the end of any $s_1$-experiment an $s_2$-experiment is possible which will leave the program in a state which is $s_3$-deadlocked.

Note that it is the interleaving to arbitrary depth of the two modal operators $\Diamond$. 

\[ S = E(S) \]
that gives the language its power. Although we do not here develop $\mathcal{L}$ into a logic for reasoning about programs, it is worth noting that as a language it is endogenous by Pnueli's classification [8]. This means that a formula states something about the 'world' of a single program, in contrast to exogenous logics such as Dynamic Logic [9] where parts of programs may be constituents of formulae.

**Theorem 2.2** Assume that each $R_i$ is image-finite.

Let $J(p) = (A \in L | p \models A)$. Then $p \sim q$ iff $J(p) = J(q)$.

This characterization theorem, together with our examples which indicate that in $\mathcal{L}$, it is possible to discuss deadlock properties of programs, encourages us to believe that our notion of observation equivalence is natural. Moreover each connective of $\mathcal{L}$ is important; by removing first negation, then conjunction, from $\mathcal{L}$ we obtain characterizations of progressively weaker equivalences.

In the remainder of the paper we study the observational equivalence (and the observational congruence it generates) of finite programs. We will consider two different types of atomic experiments and in each case we show that the congruence can be algebraically characterized.

### 3. Algebraic Characterization

In the previous section we showed how to define observational equivalence over an arbitrary set $P$ of programs or agents, in terms of an indexed family $(R_i | i \in I)$ of binary relations over $P$ with the finite image property.

Here we wish to introduce structure over $P$, by considering it to be the word algebra $W_2$ for a variety of signatures $\mathcal{L}$. In each case, we define the observation relations $R_i$ in two different ways, and hence obtain observational equivalence relations $\sim$ over $W_2$.

Now in general $\sim$ may not be a congruence with respect to the operations of $W_2$; this is to say that a pair of words $w$ and $w'$ may satisfy $w \sim w'$, but there may be a context $\mathcal{L}[\_]$ (that is, a word with a hole in it, or equivalently a derived unary operation over $W$) for which $\mathcal{L}[w] \not\sim \mathcal{L}[w']$. ($\sim$ is a congruence iff $w \sim w'$ implies $\mathcal{L}[w] \sim \mathcal{L}[w']$ for every $\mathcal{L}[\_]$.) Thus observational equivalence of $w$ and $w'$ does not guarantee that one may be exchanged for the other without observable difference.

We therefore define observational congruence $\sim_c$ over $W_2$ as follows:

$w \sim_c w'$ iff for all contexts $\mathcal{L}[\_], \mathcal{L}[w] \sim \mathcal{L}[w']$.

It is easy to check that this is a congruence, and is moreover the largest congruence contained in $\sim$.

Our aim is to find alternative characterization of this congruence relation of

"indistinguishability" that it is exactly the observational equivalence.

By this means we obtain

In the rest of this section we prove the experiment relations as main results. These axioms induce equivalence relations.

#### 3.1 The signature $\mathcal{L}_1$

$\mathcal{L}_1$ is an arbitrary signature.

$\text{NIL}$ is a nullary operation, representing nil.

The following conditions (we write $(\sim)$ for the usual equivalence relation induced by the experiment relations $\sim$) are:

$\begin{array}{l}
(\ast 1) \upsilon(u) \equiv u \\
(\ast 2) \text{if } u \equiv u' \implies \upsilon(u) \equiv \upsilon(u') \\
(\ast 3) \text{if } v \equiv v' \implies \upsilon(v) \equiv \upsilon(v')
\end{array}$

$\upsilon$ may be regarded as a "virtual" operator on programs, being not executable by the unary operations $\mathcal{L}_1$.

The set $W_2$ consists in the observed behavior of a program, and cannot perform the actions of

#### Axioms

$\begin{array}{l}
(\text{A1}) x + (y + z) = (x + y) + z \\
(\text{A2}) x + y = y + x \\
(\text{A3}) x + x = x \\
(\text{A4}) x + \text{NIL} = x
\end{array}$

**Theorem 3.1** The observed equivalence is the finest equivalence relation on $W_2$ induced by $(\text{A1}) - (\text{A4})$.

In this particular signature there is already a congruence, a congruence law $\upsilon(x + y) = \upsilon(x) + \upsilon(y)$ as follows. In view of the rooted, unordered tree representation, a pair of programs is distinct (unequivalent) if any of the descendant nodes differ.
"indistinguishability by observation in all contexts"; specifically, we aim to show that it is exactly the congruence induced by a set of equational axioms over \( \mathcal{I} \).

By this means we obtain an algebraic theory, of which \( \mathcal{W}^+_{\infty}/\sim_c \) is the initial algebra.

In the rest of this section we present three signatures \( \mathcal{I}_1, \mathcal{I}_2 \) and \( \mathcal{I}_3 \), define the experiment relations \( R_i \) for each of them in two distinct ways, and summarise our main results. These results state, for each of the six cases, a set of equational axioms which induce exactly the observational congruence determined by the relations.

3.1 The signature \( \mathcal{I}_1 = M \cup \{\text{NIL}, +\} \)

\( M \) is an arbitrary set of unary operators, whose members \( \mu \) we shall call labels. \( \text{NIL} \) is a nullary operator and \( + \) is a binary operator. Our programs are \( \mathcal{W}_i \), and our experiment relations \( \{R_\mu \mid \mu \in M\} \) are the smallest relations satisfying the following conditions (we write \( \frac{u}{v} \) for \( R_\mu \)).

\[
\begin{align*}
(\ast 1) & \quad \mu(w) = \frac{w}{w} \\
(\ast 2) & \quad \text{if } u = \frac{v}{v'} \text{ then } u + v = \frac{u + v}{v'} \\
(\ast 3) & \quad \text{if } v = \frac{v}{v'} \text{ then } u + v = \frac{u + v}{v'}
\end{align*}
\]

\( \mathcal{W}_i \) may be regarded as perhaps the simplest possible language for finite nondeterministic programs, built from the null program \( \text{NIL} \), the atomic actions \( M \) (represented by the unary operators \( M \), which may be thought of as prefixing an atomic action to a program) and the binary choice (or ambiguity) operator \( + \). An atomic experiment consists in the observation of an atomic action; the experiment fails if the program cannot perform the action.

**Axioms**

\[
\begin{align*}
(\mathcal{A}1) & \quad x + (y + z) = (x + y) + z \\
(\mathcal{A}2) & \quad x + y = y + x \\
(\mathcal{A}3) & \quad x + x = x \\
(\mathcal{A}4) & \quad x + \text{NIL} = x
\end{align*}
\]

**Theorem 3.1** The observational congruence \( \sim_c \) over \( \mathcal{W}_1 \) is exactly the congruence induced by \( \mathcal{A}1 - \mathcal{A}4 \).

In this particularly simple case the observational equivalence \( \sim \) turns out to be already a congruence, and therefore \( \sim_c = \sim \). Note the absence of the distributive law \( \mu(x + y) = \mu(x) + \mu(y) \) from the axioms. That this is natural may be explained as follows. In view of our axioms, the set \( \mathcal{W}_i/\sim_c \) is isomorphic with set of finite, rooted, unordered trees whose arcs are labelled by members of \( M \). Thus we have two distinct (incongruent) programs \( \mathcal{W}_1 = \mu_1(\mu_2(\text{NIL}) + \mu_3(\text{NIL})), \mathcal{W}_2 = \mu_1(\mu_2(\text{NIL})) + \mu_2(\mu_3(\text{NIL})) \) represented by the different trees

\[
\begin{align*}
\begin{array}{c}
\mu_1 \\
\mu_2 \\
\mu_3
\end{array}
& \quad \begin{array}{c}
\mu_1 \\
\mu_2 \\
\mu_3
\end{array}
\end{align*}
\]
Indeed, in terms of our language $L$ we have

$$w_1 \models A, \quad w_2 \not\models A$$

where $A$ is $\mu_1 (\mu_2 (\tau \land T))$. By contrast, if $B$ is $\mu_1 \mu_2 F$ then

$$w_1 \not\models B, \quad w_2 \not\models B.$$  

3.2 Unobservable atomic actions in $\Sigma$

In the above system every atomic action is observable, a program cannot proceed without being observed. Let us now suppose that among $M$ there are atomic actions which cannot be observed; for such an atomic action we shall have no corresponding atomic experiment. Intuitively, we may consider these actions as beyond the observer's control. But their presence may have a bearing upon the observable behaviour of a program, as the following example shows. Suppose that $\tau$ is an unobservable atomic action, and consider the programs $w_1 = \mu_1 (\mu_2 (\tau \land \tau (\text{NIL}))), \quad w_2 = \mu_1 (\mu_2 (\text{NIL})).$

$$w_1 = \mu_1 \mu_2 \tau \quad w_2 = \mu_1 \mu_2  $$

When we have redefined the notion of $\nu$-experiment to allow it to be accompanied by unobservable actions, then one possible result of a $\mu_1$-experiment on $w_1$ is NIL (since $\tau$ may occur unobserved), while the only possible result of the experiment on $w_2$ is $\mu_2 (\text{NIL})$. Thus we have

$$w_1 \models A, \quad w_2 \not\models A$$

where $A$ is $\mu_1 (\tau \land T)$. Notice however that both $w_1$ and $w_2$ satisfy

$$\mu_1 \mu_2 \tau.$$  

For simplicity we assume that $\tau$ is the only unobservable atomic action. (This may be formally justified; if there were two such, $\tau_1$ and $\tau_2$, we would arrive at an axiom $\tau_1 (x) = \tau_2 (x)$ — indicating that the replacement of $\tau_1$ by $\tau_2$ can affect no observation.) We therefore assume $M = \Lambda \cup \{\tau\} \{\tau \land A\}$, and we define a new set $\{\lambda | \lambda \in \Lambda\}$ of experiment relations as follows. First, define $\Sigma$ over $\Lambda$, for any $s = \mu_1 \ldots \mu_n \in M^p (n \geq 0)$, by

$$w \models s \iff w = w_0 \mu_1 w_1 \mu_2 \ldots \mu_n w_n = w'.$$

Then, writing $\tilde{\lambda}$ as $\lambda \models s$, we define for each $\lambda \in \Lambda$

$$w \models \lambda \iff w \models s \lambda n w_n$$

Thus our new atomic observation $\lambda \models s$ may absorb any finite sequence of unobservable actions before or after the action $\lambda$. It is easy to check that each $\lambda \models s$ is image-finite.

We obtain now a new observational equivalence relation $\models$ over $\Lambda$, using the definition of section 2 with the relations $\{\lambda \models s | \lambda \in \Lambda\}$. This induces, as before contained in $\equiv$, but a congruence. For example, replace each of these programs

$$\sigma (\tau (\text{NIL})) \not\subseteq \sigma (\text{NIL})$$

(We will often, as here, express congruence. It involves an important step in deriving consequences of $\tau$ in a program.)

3.3 The Signature $\Sigma$

We now add a binary operator $\bowtie$ which may be considered to correspond to the experiments which $\approx$ are composed of experiments which $\approx$ are composed of atomic actions which may proceed concurrently. The two properties are reflexive, symmetric, and transitive.

$$\bowtie \quad \text{if} \quad \approx \bowtie \approx' \quad \text{if} \quad \approx \bowtie \approx'$$

To express communication among the elements of the alphabet of names, and the bijection with it. We use $\{\lambda, \mu, \gamma\}$ to range over $\Lambda$, $\mu$ to range over $\mu$, $\lambda$ and $\gamma$ may occur where $\lambda$ occurs for some $\lambda$; the result

$$\approx \bowtie \approx'$$

This choice to represent...
This induces, as before, an observational congruence $\approx_c$ (the largest congruence contained in $\approx$), but this is not identical with $\approx$. Indeed, the latter is not a congruence. For example, it is easy to check that $\tau(NIL) \approx NIL$; but if we place each of these programs in the context $s[()] = \lambda_1(\lambda_3(NIL) + [])$ we obtain $s[\tau(NIL)] \neq s[NIL]$ as may be readily checked (this is, in effect the pair $w_1, w_2$ discussed earlier).

**Axioms**

\begin{align*}
(A5) \quad & x + \tau x = \tau x \\
(A6) \quad & \mu(x + \tau y) = \mu(x + y) + \mu y \quad (\mu < M)
\end{align*}

(We will often, as here, omit parentheses and write $\mu(x)$ as $\mu x$.)

**Theorem 3.2** The observational congruence $\approx_c$ over $W_{\tau_1}$ is exactly the congruence induced by (A1)–(A6).

This theorem is not so immediate as the previous one, partly because $\approx$ is not a congruence. It involves defining a normal form for programs in $W_{\tau_1}$; the most important step in deriving a normal form is the use of (A6) to eliminate most occurrences of $\tau$ in a program.

### 3.3 The Signature

$S_2 = \{[]\}$

We now add a binary operator $\mid$ to our signature; it is one of a variety of operators which may be chosen to represent the combination of a pair of programs which may proceed concurrently and may also communicate with one another. These two properties are reflected by separate new conditions upon the experiment relations $\mid$.

One condition (in two parts) expresses that the program $u \mid v$ admits all the experiments which $u$ and $v$ admit separately. (Since an atomic experiment corresponds to a single atomic action, the simultaneous activity of $u$ and $v$ cannot be observed.)

\begin{align*}
(\ast 4) \quad & \text{if } u \mid u' \text{ then } u \mid v \mid u' \mid v \\
(\ast 5) \quad & \text{if } v \mid v' \text{ then } u \mid v \mid u \mid v'
\end{align*}

To express communication we introduce a little structure over $M$. We assume $M = \Lambda \cup \{\tau\}$ as before, and also that $\Lambda = \Lambda \cup \bar{\Lambda}$, where $\Lambda$ is a possibly infinite alphabet of names, and that the alphabet $\bar{\Lambda}$ of co-names is disjoint from $\Lambda$ and in bijection with it. We represent the bijection and its inverse by overbar $\bar{\cdot}$, and use $\bar{a}, \bar{b}, \bar{c}$ to range over $\bar{\Lambda}$. Thus $\bar{a} \in \bar{\Lambda}$, and $\bar{\Lambda} = \bar{a}$. We continue to use $\lambda$ to range over $\Lambda$, and $u, v$ to range over $M = \Lambda \cup \{\tau\}$. Communication between $u$ and $v$ may occur when $u$ admits a $\lambda$-experiment and $v$ admits a $\bar{\lambda}$-experiment, for some $\lambda$; the result is a $\tau$-action of $u \mid v$.

\begin{align*}
(\ast 6) \quad & \text{if } u \mid u' \text{ and } v \mid v' \text{ then } u \mid v \mid u' \mid v'
\end{align*}

This choice to represent communication between components of a program by a $\tau$-action
will allow us in section 3.4 to treat internal communications as unobservable.

Now taking \( \{ I \mid \mu \in M \} \) to be the smallest relations over \( W_2 \) satisfying \( (\rightarrow 1) \) and \( (\rightarrow 6) \), we obtain an observational equivalence \( \sim \) over \( W_2 \), as in section 2. As before, this turns out to be a congruence, so that \( \sim \) is identical with \( \sim \).

Since axioms (A1) - (A4) are satisfied by \( \sim \), we may adopt the notation for any \( n \geq 0 \),

\[
\sum_{1 \leq i \leq n} x_i = \begin{cases} \sum_{1 \leq i \leq n} x_i & \text{if } n > 0 \\ \text{NIL} & \text{if } n = 0 \end{cases}
\]

Now we add an axiom for "|":

(A7) if \( u \) is \( \sum_{1 \leq i \leq n} x_i \) and \( v \) is \( \sum_{1 \leq j \leq m} y_j \) then

\[
u \mid v = \sum_{1 \leq i \leq n} (x_i | v) + \sum_{1 \leq j \leq m} (u | y_j) + \sum_{1 \leq i \leq n} \tau (x_i | y_j)
\]

Examples of (A7) are (for distinct names \( \alpha, \beta, \gamma \))

\[
\begin{align*}
(\alpha x_1 + \beta x_2) | \gamma y &= \alpha (x_1 | yy) + \beta (x_2 | yy) + \gamma ((x_1 + x_2) | y) \\
(\alpha x_1 + \beta x_2) | \beta y &= \alpha (x_1 | \beta y) + \beta (x_2 | \beta y) + \beta ((x_1 + x_2) | y) + \tau (x_2 | y) \\
(\sum_{1 \leq i \leq n} x_i) | \text{NIL} &= \sum_{1 \leq i \leq n} (x_i | \text{NIL}) + \text{NIL} + \text{NIL}
\end{align*}
\]

Note that (A7) allows \( | \) to be eliminated from any word in \( W_2 \).

Theorem 3.3 The observational congruence \( \sim \) over \( W_2 \) is exactly the congruence induced by (A1) - (A4) and (A7).

Remark The following laws for "|" may be proved to hold over \( W_2 \) by induction on the structure of terms (though they are not deducible from (A1) - (A4), (A7) by equational reasoning):

\[
\begin{align*}
x | (y | z) &= (x | y) | z \\
x | y &= y | x \\
x | \text{NIL} &= x
\end{align*}
\]

3.4 Unobservable actions in \( W_2 \)

We now repeat for \( W_2 \) what we did for \( W_1 \); we wish to treat \( \tau \) as an unobservable atomic action (in particular, the intercommunication of \( u \) and \( v \) in \( u | v \) is not an observable action). If we define the experiment relations \( \{ \lambda \mid \lambda \in \Lambda \} \) as previously then we gain again an observational congruence \( \sim_\Lambda \) over \( W_2 \). We then might expect this to be exactly the congruence induced by the axioms (A1) - (A7), but this is not the case, since (A6) is not satisfied by \( \sim_\Lambda \) over \( W_2 \). We shall demonstrate, in particular that the following instance of (A6) is false:

\[
\alpha (\beta \text{NIL} + \tau \text{NIL}) \not\sim_\Lambda \alpha (\beta \text{NIL} + \text{NIL}) + \tau \text{NIL}
\]

For this would imply that \( \gamma \text{NIL} \mid (\beta \text{NIL} + \alpha) \beta \text{NIL} \).

Calling the left and right part

\[
u \not\approx u' = \gamma \text{NIL},
\]

for \( v \not\approx v' \) implies \( v_1 = \gamma \text{NIL} \) and \( v_2 = \gamma \text{NIL} \).

Now if (1) holds, then \( u' \not\approx v_1 \) and \( u' \not\approx v_2 \) are second impossible as \( u' \approx v_1 \). But

\[
u' \not\approx \text{NIL} | \text{NIL}
\]

while the only \( \gamma \)-experiment is \( v_1 \not\approx \text{NIL} | \text{NIL} \).

Hence we must have \( \text{NIL} | \text{NIL} \).

Axion (A6) fails in contexts in which to place \( u' \mid v_1 \) over \( W_2 \) by replacing

Axioms

(A6.1) \( \nu (x + y) = \nu x + \nu y \\
(A6.2) \nu x | y = \nu y | x
\]

These axioms are indeed satisfied by placing \( x = \text{NIL} \) in \( \nu (x + y) \) and \( y \) in (A6).

\[
u (x + \tau y) = \nu x + \nu y
\]

and use two instances of (A7) and (A6).

Theorem 3.4 The observational congruence induced by (A1) - (A5), (A6.1) and (A6.2)

This theorem is the same as (A1) - (A5), (A6.1) and (A6.2) in the presence of extraneous actions.
For this would imply the observational equivalence (not congruence)
\[ \gamma \text{NIL} \mid \alpha (\beta \text{NIL} + \tau \text{NIL}) \approx \gamma \text{NIL} \mid (\alpha \beta \text{NIL} + \tau \text{NIL}) \]  
(1)

Calling the left and right sides of (1) \( u \) and \( v \) respectively, we have
\[ u \overset{\beta}{\rightarrow} u' = \gamma \text{NIL} \mid (\beta \text{NIL} + \tau \text{NIL}) \]
while \( v \overset{\beta}{\rightarrow} v' \) implies that \( v' = v_1 \) or \( v' = v_2 \) where
\[ v_1 = \gamma \text{NIL} \mid (\beta \text{NIL} + \text{NIL}) \]
\[ v_2 = \gamma \text{NIL} \mid \text{NIL} \]

Now if (1) holds, then by definition of \( \approx \) we must have \( u' = v_1 \) or \( u' = v_2 \). The second is impossible since \( u' \overset{\beta}{\rightarrow} \gamma \text{NIL} \mid \text{NIL} \), while \( v_2 \overset{\beta}{\rightarrow} v_2' \) is impossible. Hence
\[ u' \approx v_1 \]. But
\[ u' \not\approx \gamma \text{NIL} \mid \text{NIL} \]

while the only \( \gamma \)-experiment for \( v_1 \) is
\[ v_1 \overset{\beta}{\rightarrow} \gamma \text{NIL} \mid (\beta \text{NIL} + \text{NIL}) \]

Hence we must have \( \text{NIL} \mid \text{NIL} = \gamma \text{NIL} \mid (\beta \text{NIL} + \text{NIL}) \), which is easily false.

Axiom (A6) fails for \( W \), because the operator \( \mu \) provides a richer class of contexts in which to perform experiments. We therefore hope to characterise \( \approx \) over \( W \) by replacing (A6) by something weaker.

Axioms
\[ (\text{A} 6.1) \quad \mu (x + \tau y) = \mu (x + \tau y) + \mu y \]
\[ (\text{A} 6.2) \quad \mu \tau y = \mu y \]

These axioms are indeed implied by (A1) - (A6). First observe that (A6.2) follows by placing \( x = \text{NIL} \) in (A6) and using the other axioms. Then to get (A6.1) place \( \tau y \) for \( y \) in (A6).
\[ \mu (x + \tau y) = \mu (x + \tau y) + \mu \tau y \]

and use two instances of (A6.2).

Theorem 3.4 The observational congruence \( \approx \) over \( W \) is exactly the congruence induced by (A1) - (A5), (A6.1), (A6.2) and (A7).

This theorem is the central result of our paper, since the method not only generalises in a routine manner to the corresponding theorem for our next signature \( \Gamma_3 \), but also applies we believe - with minor adjustments - to many other signatures and experiment relations representing concurrent and communicating activity. The axioms (A1) - (A5), (A6.1) and (A6.2) seem to be what is required for the operators in \( \Gamma_1 \) in the presence of extra operators for communication and concurrency.
3.5 The signature $\Sigma_3 = \Sigma_2 \cup \mathcal{S}$

In [4] we considered operations over behaviours corresponding to $\Sigma_2$, together with two other families of operations called relabelling and restriction; in the present context, these operations may be described as changing (bijectively) the labels for atomic experiments (i.e., permutations of $\Lambda$), and restricting the class of atomic experiments to a subset of $\Lambda$. The approach in [4] was to classify behaviours into sorts; a sort $L$ was a subset of $\Lambda$, and the behaviours $B_L$ of sort $L$ were those which employed only members of $L$ as labels.

Here we do not consider sorts; these may be later introduced, and are indeed useful in providing a stronger basis for reasoning about realistic programs. Moreover, we can treat relabelling and restriction as subclasses of a wider family of operations indexed by a subset of the partial functions $N \rightarrow M$ from $N$ to $M$. To this end we add to the signature $\Sigma_2$ the operators

$$\mathcal{S} = \{ [S] \mid S \in M \rightarrow M, S \tau = \tau \}$$

We shall postfix these operators. We characterise them operationally by adding a further condition for the production relations $\frac{u}{v}$:

$$(\rightarrow 7) \text{ If } w \frac{u}{v} w' \text{ and } S_u \text{ is defined then } w[S] S_u w'[S]$$

Now we take $\frac{u}{v} \mid u \in M$ to be the smallest relations over $W^L_3$, satisfying $(-1) - (-7)$, and again obtain an observational equivalence $\sim$ over $W^L_3$, which is a congruence, so that again $\sim_C$ is identical with $\sim$.

The axioms needed to characterise $\mathcal{S}$ are the obvious ones:

(A8) $(u \cdot x)[S] = S_u(x[S])$ if $S_u$ is defined, NIL otherwise

(A9) $(x + y)[S] = x[S] + y[S]$

(A10) NIL$[S] = NIL$

Theorem 3.5 The observational congruence $\sim_C$ over $W^L_3$ is exactly the congruence induced by (A1) - (A4) and (A7) - (A10).

The treatment of experiment relations $\{ \frac{u}{v} \mid \lambda \in \Lambda \}$ and the corresponding observational congruence $\sim_C$ over $W^L_3$ is exactly as it was for $W^L_2$, and by trivially adapting the proof of Theorem 3.4 we obtain

Theorem 3.6 The observational congruence $\sim_C$ over $W^L_3$ is exactly the congruence induced by (A1) - (A5), (A6.1), A(6.2) and (A7) - (A10).

3.6 Summary

We have characterised observational congruence in six cases by equational axioms. There are three signatures, and in each case two classes of experiment relations:

$\{ \frac{u}{v} \mid u \in M \}$ when the atomic action $\tau$ is observable, and $\{ \Rightarrow \mid \lambda \in \Lambda \}$ when $\tau$ is not directly observable but may "occur" a finite number of times during any atomic experiment. The axioms needed in every case:

<table>
<thead>
<tr>
<th>Signature</th>
<th>$\Sigma_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production rules</td>
<td></td>
</tr>
<tr>
<td>Axioms for $\sim_C$</td>
<td></td>
</tr>
<tr>
<td>Axioms for $\sim_C$</td>
<td></td>
</tr>
</tbody>
</table>

Furthermore, we believe that these results will be needed in the context of the "\" numbers, and that the observable family of partial restrictions is indeed a congruence.

References


experiment. The axioms for each case may be tabulated as follows ((A1) – (A4) are needed in every case):

| Signature | \( L_1 = M \cup \{ \text{NIL},+ \} \) | \( L_2 = L_1 \cup \{ | \} \) | \( L_3 = L_2 \cup \{ S \} \) |
|-----------|---------------------------------|-----------------|-----------------|
| Production rules | \( (+1) - (+3) \) | \( (+1) - (+6) \) | \( (+1) - (+7) \) |
| Axioms for \( \sim_A \) | \(- \) | \( (A7) \) | \( (A7) - (A10) \) |
| Axioms for \( \sim_A \) | \( (A5),(A6) \) | \( (A5),(A6.1),(A6.2) \) | \( (A5),(A6.1),(A6.2),(A7) \) | \( (A5),(A6.1),(A6.2),(A7),(A10) \) |

Furthermore, we believe that the replacement of \( (A6) \) by two axioms \( (A6.1) \) and \( (A6.2) \) will be needed with the introduction of any operator representing concurrent activity, in place of "|"; and that this replacement persists with the addition of any reasonable family of partial relabelling operators (even multi-valued ones, though we restricted consideration to single valued relabelling).

References


