Algebraic Laws for Nondeterminism and Concurrency

MATTHEW HENNESSY AND ROBIN MILNER

University of Edinburgh, Edinburgh, Scotland

Abstract. Since a nondeterministic and concurrent program may, in general, communicate repeatedly with its environment, its meaning cannot be presented naturally as an input/output function (as is often done in the denotational approach to semantics). In this paper, an alternative is put forth. First, a definition is given of what it is for two programs or program parts to be equivalent for all observers; then two program parts are said to be *observation congruent* iff they are, in all program contexts, equivalent. The *behavior* of a program part, that is, its meaning, is defined to be its observation congruence class.

The paper demonstrates, for a sequence of simple languages expressing finite (terminating) behaviors, that in each case observation congruence can be axiomatized algebraically. Moreover, with the addition of recursion and another simple extension, the algebraic language described here becomes a calculus for writing and specifying concurrent programs and for proving their properties.

Categories and Subject Descriptors: F.3.2 **[Logics and Meanings of Programs]**: Semantics of Programming Languages—*algebraic approaches to semantics*

General Terms: Theory

Additional Key Words and Phrases: Communicating processes, observational equivalence

1. *Introduction*

The denotational approach to the semantics of programming languages has been well developed in recent years [1, 11] and applied successfully to many nontrivial languages. Even languages with parallel constructs have been treated in this way, using the power-domain constructions of [3], [7], and [10]. 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 [4].

In the face of 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 behavior, or operational semantics, of the programs. One way of making the link is to demand that the denotational model be *fully abstract* with respect to the operational semantics. This means simply that two program phrases should have the same denotation if, and only if, the opera-

A preliminary version of this paper, entitled "On Observing Nondeterminism and Concurrency," appeared in the *Proceedings of the 7th Colloquium on Automata, Languages and Programming*. Lecture Notes in Computer Science, vol. 85. Springer-Verlag, New York, Berlin, Heidelberg, 1980, pp. 299–309. Those portions that appeared in the preliminary are reprinted with permission of Springer-Verlag.

Authors' address: Department of Computer Science, James Clerk Maxwell Building, The Kings Buildings, University of Edinburgh, Mayfield Road, Edinburgh EH9 3JZ, Scotland.

tional meaning of every program remains unchanged when one phrase is replaced by the other.

What exactly is meant by the behavior of nondeterministic 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 behavior 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 programs with finite behavior, and show that in each case it can be characterized by algebraic axioms. This leads automatically to a fully abstract model; it is just the initial algebra generated by the axioms. Moreover, a proper understanding of the finite case seems a necessary prelude to a study of programs with infinite behavior. 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 and specifying concurrent programs and for proving their properties. This language was introduced in [5]; it was partly the need for a firm basis for the algebraic laws discussed there that led to the present study of observation equivalence.

In Section 2 we present our general framework. In Sections 3–6 we outline and summarize our results for the languages considered. Proofs of the main results are contained in Appendixes A–C.

The present paper is a full presentation, complete with proofs, of results first announced without proof in [2].

2. *Observational Equivalence of Processes*

2.1. EQUIVALENCE. In this section we introduce a way of defining equivalence between programs that is based entirely on operational considerations; informally, two programs are *equivalent* when no observations can distinguish them. Further, two subprograms or program phrases are *congruent* if the result of placing each of them in any program context yields two equivalent programs. Then, considering the phrases as modules, one can be exchanged for the other in any program without affecting the observed behavior of the latter.

However, much is left vague by this prescription. First, what are observations? Second, how can they be used to distinguish programs? In this section we answer these two questions, thereby obtaining a precise notion of equivalence, and hence also of congruence. Note that the answer to the first question does not determine the answer to the second; observations may be used in many different ways to distinguish more or fewer programs. We cannot argue that our answer is best, only that it is natural; to this end, we give an alternative characterization of the resulting equivalence relation in Section 2.2.

In the case of deterministic sequential programs, the behavior of a program p is usually taken to be its input–output function $\mathrm{IO}(p)$. Here, an observation of p is taken to be a pair of states (or values): an input state and the resulting output state (if any). Then programs p and q are equivalent, written $p \sim q$, if they yield the same observation sets; that is, if $\mathrm{IO}(p) = \mathrm{IO}(q)$. The corresponding congruence relation $\sim_c$ is then defined as follows: $p \sim_c q$ if for every suitable program context $\mathscr{C}[\ ]$, $\mathscr{C}[p] \sim \mathscr{C}[q]$. If the language is defined algebraically, that is, by operations

for constructing new programs from ones already defined, the $\sim_c$ turns out to be the largest congruence relation included in $\sim$.

However, any satisfactory comparison of the behavior of concurrent programs must take into account their intermediate states as they progress through a computation, because differing intermediate states can be exploited in different program contexts to produce different overall behavior (e.g., deadlock). With this in mind, we now proceed to a more refined notion of behavioral equivalence, which we call *observational equivalence.* This notion may be defined for objects more general than programs, which (for the remainder of the present section) we shall call *processes.*

Let P, then, be a set of objects that we may think of as processes. We take the view that any observation of $p \in P$ entails some participation by p itself; p is an active participant, as well as the observer. Thus, the act of observing a process changes its state. So if we adopt the familiar technique of identifying the state of a process with the process itself, we can say that observation changes the process into a new process. This change may not be deterministic; hence the effect of a particular type of observation—applied to an arbitrary process—may be captured by a binary relation over P. In general, we presuppose a set I of possible types of observation, so we then have a set $\{R_i \subseteq P \times P, i \in I\}$ of observation relations.

Using these relations, we define a sequence of equivalence relations $\sim_n$ over P ($n \geq 0$), in such a way that $\sim_{n+1} \subseteq \sim_n$, as follows:

$p \sim_0 q$ if $p, q \in P$ (i.e., $\sim_0 = P \times P$);
$p \sim_{n+1} q$ if for every $i \in I$,

(i) $\langle p, p' \rangle \in R_i$ implies, for some q', $\langle q, q' \rangle \in R_i$ and $p' \sim_n q'$;
(ii) $\langle q, q' \rangle \in R_i$ implies, for some p', $\langle p, p' \rangle \in R_i$ and $p' \sim_n q'$.

Then p and q are *observationally equivalent*, written $p \sim q$, if $p \sim_n q$ for every n.

Thus, we have defined $\sim$ to be $\bigcap_n \sim_n$. In fact, we have taken $\sim$ to be the limit $\bigcap_n E^n(P \times P)$, where $E(S)$ is defined for any $S \subseteq P \times P$ as follows:

$\langle p, q \rangle \in E(S)$ if for every $i \in I$

(i) $\langle p, p' \rangle \in R_i$ implies, for some q', $\langle q, q' \rangle \in R_i$ and $\langle p', q' \rangle \in S$;
(ii) $\langle q, q' \rangle \in R_i$ implies, for some p', $\langle p, p' \rangle \in R_i$ and $\langle p', q' \rangle \in S$.

Now if E has the property that $E(\bigcap_n S_n) = \bigcap_n (E(S_n))$ for every decreasing sequence S_n of relations, that is if E is *anticontinuous*, then it follows from classical fixed-point theory that $\sim$ is the maximum fixed-point of the map E of relations.

Let us say that $R \subseteq P \times P$ is *image-finite* if, for each $p \in P$, $\{p' \mid \langle p, p' \rangle \in R\}$ is finite. It turns out that the image-finiteness of each R_i, $i \in I$, is sufficient to ensure that E is anticontinuous, so the following theorem holds (see Appendix A for the proof):

THEOREM 2.1. *If R_i is image-finite for each $i \in I$, then $\sim$ is the maximum solution to $S = E(S)$.*

Hitherto we have called $i \in I$ a *type* of observation, and then an instance $\langle p, p' \rangle \in R_i$ is a particular observation (of p). It can also be regarded as a communication between p and an observer; in some of the program languages that we introduce later we exploit this symmetry by representing communication between two processes p and q, running concurrently, as *mutual observation* between the processes.

For the present, we can also regard a single observation $\langle p, p' \rangle \in R_i$ as an *atomic experiment* by the observer on p. A more complicated experiment may consist of a finite sequence of atomic experiments. Let s be the sequence $i_1, \ldots, i_n$ in I ($n \geq 0$); then an *s-experiment* on p is a sequence $p_0, \ldots, p_n$ where $p_0 = p$ and $\langle p_{k-1}, p_k \rangle \in R_{i_k}$. Thus, if $p \sim q$ and some s-experiment leads from p to p', then (assuming image-finiteness) by Theorem 2.1 there exists an s-experiment on q leading to some q' such that $q \sim q'$. If we consider a computation as a sequence of experiments (or communications), then the above remarks show that intermediate states are compared. In fact, if p is to be equivalent to q, there must be a strong relationship between their respective intermediate states. At each intermediate stage in the computations, the respective "potentials" must also be the same. The alternative characterization of observational equivalence given in Section 2.2 will help to shed further light on such intuitive discussions. However the principal reason for introducing this alternative characterization in the present paper is to justify our interest in the notion of observational equivalence despite its rather complicated definition. Moreover, we will find it easier to compare it with simpler forms of equivalence that one might be tempted to define. One such equivalence is

$p \sim_e q$ if for every $s \in I^*$,
p has an s-experiment if and only if q has an s-experiment.

This identifies a process with the set of s-experiments that can be performed on it and reflects the view of classical automata theory that identifies a machine with the language it accepts. The alternative characterization will make apparent the difference between observational equivalence and $\sim_e$ and will underline the deficiencies of the latter.

2.2. LOGICAL CHARACTERIZATION. The alternative characterization depends on the identification of a process with the properties it enjoys. Then we can say that two processes are equivalent if and only if they enjoy exactly the same properties. This is perhaps more illuminating in its negative form: two processes are inequivalent if one enjoys a property that the other does not enjoy.

There are a number of parameters in this definition. First, it presupposes a set $\mathscr{A}$ of properties. Second, we need a notion of a process $p \in P$ enjoying a property $A \in \mathscr{A}$. This can be modeled as a binary relation $\models \subseteq P \times \mathscr{A}$. We write $\models$ in an infix manner and $p \models A$ may be read "p enjoys the property A." Let $\mathscr{A}(p)$ be the set of properties enjoyed by p, that is,

$$\mathscr{A}(p) = \{A, p \models A\}.$$

An equivalence between processes can be defined as follows:

$$p \sim_{\mathscr{A}} q \quad \text{if} \quad \mathscr{A}(p) = \mathscr{A}(q) \quad \text{for} \quad p, q \in P,$$

We now introduce a particular set of properties and a particular satisfaction relation for which it will follow that

$$p \sim q \quad \text{if and only if} \quad p \sim_{\mathscr{A}} q.$$

The properties in question are rather general, but they depend on the set of observation relations $\{R_i \subseteq P \times P, i \in I\}$ given in the previous section. They are best expressed as formulas in a simple modal language $\mathscr{L}$. $\mathscr{L}$ is defined by extending propositional logic with a set of modal operators, $\langle i \rangle$, one for each observation

relation R_i. The connectives of propositional logic have their usual meaning and a process p will enjoy the property $\langle i \rangle A$, that is, $p = \langle i \rangle A$, if there is an i-experiment $\langle p, p' \rangle$ such that p' enjoys the property A. The language looks deceptively simple, but it derives its power from the ability to define the dual modal operators $[i]$ from $\langle i \rangle$ using negation. We now give the relevant definitions.

Let the language $\mathscr{L}$ of formulas be the least set such that

(i) $T \in \mathscr{L}$,
(ii) $A, B \in \mathscr{L} \Rightarrow A \wedge B \in \mathscr{L}, \neg A \in \mathscr{L}$,
(iii) $A \in \mathscr{L}$ and $i \in I \Rightarrow \langle i \rangle A \in \mathscr{L}$.

The satisfaction relation $\models \subseteq P \times \mathscr{L}$ is the least relation such that

(i) $p \models T$ for all $p \in P$,
(ii) $p \models A \wedge B$ iff $p \models A$ and $p \models B$,
(iii) $p \models \neg A$ iff not $p \models A$,
(iv) $p \models \langle i \rangle A$ iff for some i-experiment $\langle p, p' \rangle$, $p' \models A$.

In examples and later discussion we adopt the following convenient notations:

F stands for $\neg T$,
$A \vee B$ stands for $\neg(\neg A \wedge \neg B)$,
$\langle s \rangle A$ stands for $\langle i_1 \rangle \cdots \langle i_n \rangle A$, where $s = i_1 \cdots i_n$, $n \geq 1$,
$[s]A$ stands for $\neg\langle s \rangle \neg A$.

We say p is *s-deadlocked* if there are no s-experiments on p. From the definition of the satisfaction relation we can now interpret many simple sentences as assertions about the possibility of deadlock.

Examples

(a) $p \models \langle s \rangle T$: It is possible to carry out an s-experiment on p.
(b) $p \models [s]F$: p is s-deadlocked.
(c) $p \models \langle s_1 \rangle([s_2]F \vee [s_3]F)$: It is possible, via an s_1-experiment, to get into a state that is either s_2-deadlocked or s_3-deadlocked.
(d) $p \models [s_1](\langle s_2 \rangle [s_3]F)$: At the end of any s_1-experiment, an s_2-experiment that will leave the program in a state that is s_3-deadlocked is possible.

Note that it is the interleaving to arbitrary depth of the two model operators $\Diamond$, $\Box$ that gives the language its power. Although we do not here develop $\mathscr{L}$ 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 formulas.

Let $\mathscr{L}(p) = \{A \in \mathscr{L}, p \models A\}$. Thus, $\mathscr{L}(p)$ is the set of properties enjoyed by the process p.

THEOREM 2.2. *If each R_i is image finite then*

$$p \sim q \quad \text{if and only if} \quad \mathscr{L}(p) = \mathscr{L}(q).$$

This characterization theorem (proved in Appendix A), together with our examples, which indicate that in $\mathscr{L}$ it is possible to discuss deadlocking properties of processes, encourages us to believe that our notion of observation equivalence is natural.

Moreover, we shall see that each connective of $\mathscr{L}$ is important; by removing first negation, then conjunction, from $\mathscr{L}$ we obtain characterizations of progressively weaker equivalences. It is of some interest to examine these weaker equivalences, and the rest of the present section is devoted to this task. However, the reader may note that the work in later sections is only concerned with observational equivalence for various sets R_i.

For any set of formulas $\mathscr{F} \subseteq \mathscr{L}$ let

$$p \sim_{\mathscr{F}} q \quad \text{if for every} \quad A \in \mathscr{F}, \qquad p \models A \quad \text{if and only if} \quad q \models A.$$

If $\mathscr{F}$ is the empty set then $\sim_{\mathscr{F}}$ identifies all processes. The larger we make $\mathscr{F}$, the more discriminating the equivalence relation $\sim_{\mathscr{F}}$ becomes. Theorem 2.2 states that $\sim$ coincides with $\sim_{\mathscr{L}}$. The two sets of formulas just mentioned are

$$\mathscr{M} = \{A \in \mathscr{L}, A \text{ does not contain } \neg\}$$
$$\mathscr{N} = \{A \in \mathscr{M}, A \text{ does not contain } \wedge\}.$$

It is not difficult to establish that

THEOREM 2.3

$$p \sim_e q \qquad \textit{if and only if} \qquad p \sim_{\mathscr{N}} q.$$

This result emphasizes the weakness of $\sim_e$; within $\mathscr{N}$ we cannot define $\boxed{i}$ or F, which are essential to express properties concerning deadlock. In Section 3.2 we will use the experiment relations defined in Section 3.1 to show that, in general, $\sim_e$ is weaker than $\sim$. We will also give an example to show that it is weaker than $\sim_{\mathscr{M}}$ which in turn is weaker than $\sim$. Anticipating these examples, it is reasonable to ask if there is a natural characterization of $\sim_{\mathscr{M}}$ in terms of the experiment relations R_i, similar to the characterizations in Theorems 2.2 and 2.3 for $\sim_{\mathscr{L}}$ and $\sim_{\mathscr{M}}$, respectively. Such a characterization can be obtained by considering the asymmetric version of the relation E, used in Section 2.1 to define $\sim$.

For any $S \subseteq P \times P$ let $AE(S)$ be defined by

$\langle p, q\rangle \in AE(S)$ if for every $i \in I$,

$\langle p, p'\rangle \in R_i$ implies, for some q', $\langle q, q'\rangle \in R_i$ and $\langle p'q'\rangle \in S$.

Let

(i) $\underset{\sim}{\sqsubseteq}_0$ be $P \times P$,

(ii) $\underset{\sim}{\sqsubseteq}_{n+1}$ be $AE(\underset{\sim}{\sqsubseteq}_n)$.

and let

$$p \underset{\sim}{\sqsubseteq} q \qquad \text{if for every } n \geq 0, \quad p \underset{\sim}{\sqsubseteq}_n q.$$

If each R_i is image finite, then we can modify Theorem 2.1 to show that $\underset{\sim}{\sqsubseteq}$ is the maximal solution to $S = AE(S)$. In general the relation $\underset{\sim}{\sqsubseteq}$ is reflexive and transitive but not necessarily symmetric. We let $\approx$ denote the natural equivalence relation it generates, $\underset{\sim}{\sqsubseteq} \cap \underset{\sim}{\sqsupseteq}$. Surprisingly it turns out that in general $\approx$ is much weaker than $\sim$. An example will be given in Section 3.2. However, we do have

THEOREM 2.4. *If each R_i is image finite, then*

$$p \approx q \qquad \textit{if and only if} \qquad p \sim_{\mathscr{M}} q.$$

In Section 3.2 we will also present examples that show that $\sim_n$, $n \geq 0$ and $\underset{\sim}{\sqsubset}_n$, $n \geq 0$ are true hierarchies; that is, we will give processes p_n, q_n, $n \geq 0$ such that $p_n \sim_n q_n$, $p_n \underset{\sim}{\sqsubset}_n q_n$, and $p_n \not\sim_{n+1} q_n$, $p_n \not\underset{\sim}{\sqsubset}_{n+1} q_n$ for every $n \geq 0$.

These remarks show that various notions of equivalence of processes can be defined starting from either experiment relations or sets of properties that one expects processes to enjoy. In the present setting observational equivalence, $\sim$, seems the most natural and in the remainder of the paper we study its application to finite programs. We consider two different types of atomic experiment, and in each case we show that the congruence generated by the equivalence can be algebraically characterized.

3. *Application to a Simple Nondeterministic Language*

In the previous section we showed how to define observational equivalence over an arbitrary set P of processes or agents in terms of an indexed family $\{R_i \mid i \in I\}$ of binary relations over P with the finite-image property.

In this section, we introduce a simple language for defining processes. Intuitively, every program in the language defines a nondeterministic finite machine, and, associated with every possible action i that a machine can make, we have an experiment relation R_i. This relation corresponds to the performance of an action i by the machine. In fact, we have two different sets of experiment relations, depending on whether or not the machines can perform actions that are not observable. This leads to two different observational equivalences $\sim$, $\approx$ over programs.

The language we use for defining machines is simply the word algebra W_Σ over a signature Σ. This approach has certain advantages. It introduces structure on the machines in that each operator in the signature can be viewed as a constructor: a method for defining a new machine in terms of existing machines, which are called its *constituents*. Moreover, the behavior of the new machine is uniquely determined by the behavior of its constituents. This will be reflected in our definition of the experiment relations R_i; the result of applying an experiment to a machine depends entirely on what happens when we apply experiments to its constituents. Another advantage of this structural view of machines is that we can augment the signature, thereby increasing the descriptive power of the language. The definition of the experiment relations on the extended language can be given simply by adding clauses to cover the new constructors. Indeed, this is the approach in the present paper and by Section 5 we have all of the operators of the language CCS [6] in our signature apart from recursion.

In general, $\sim$ (or $\approx$) may not be a congruence with respect to the operations of W_Σ; this is to say that a pair of words p, p' may satisfy $p \sim p'$ but there may be a context $\mathscr{C}[\,]$ (i.e., a word with a hole in it, or equivalently a derived unary operation over W_Σ) for which $\mathscr{C}[p] \not\sim \mathscr{C}[p']$. ($\sim$ is a congruence if and only if $p \sim p'$ implies $\mathscr{C}[p] \sim \mathscr{C}[p']$ for every $\mathscr{C}[\,]$.) Thus, observational equivalence of two words does not guarantee that one may be exchanged for the other without observable difference.

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

$$p \sim_c p' \quad \text{if for all contexts} \quad \mathscr{C}[\,], \quad \mathscr{C}[p] \sim \mathscr{C}[p'].$$

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

The definition of $\sim_c$ is in general complicated and a direct proof that $p \sim_c p'$ for a particular pair of words p, p' is quite difficult. Our aim is to provide an alternative proof method for deriving such statements. We isolate properties of $\sim_c$ (and $\approx_c$) as axioms. One can then derive statements such as $p \sim_c p'$ by using these axioms to transform p into p' or vice versa. In fact, for each of the languages considered we show that $\sim_c$ (and $\approx_c$) can be characterized completely by an appropriate set of axioms; that is, $p \sim_c p'$ if and only if we can derive $p = p'$ from the axioms using substitution. In general, this is false if we add recursion to the languages. However, the axioms together with some form of induction are a powerful proof method for deriving observational congruence.

In the remainder of this section we present a signature Σ_1 and define experiment relations R_i over W_{Σ_1} in two distinct ways. For each way, we give a set of equational axioms that induce exactly the observational congruence determined by the relations.

3.1. THE SIGNATURE $\Sigma_1 = M \cup \{\text{NIL}, +\}$. Let M be an arbitrary set, representing the atomic actions that may be performed by a program. We shall let μ, ν range over M. The words W_{Σ_1} may be regarded as perhaps the simplest language for finite nondeterministic programs built from M, together with the null program NIL, a nullary operator representing termination, and the binary operator + representing choice. The members of M, which are unary operators, may be thought of as prefixing an atomic action to a program. As an example, the program

$$p = \mu_1(\mu_2(\text{NIL}) + \mu_1(\mu_2(\text{NIL})))$$

may first perform μ_1 only; thereafter, it may perform μ_2 and terminate, or perform μ_1 then μ_2 and terminate.

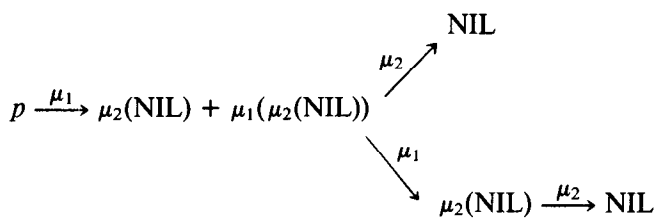
We now suppose that an atomic experiment consists in observing an atomic action; then the relations $\{R_\mu \mid \mu \in M\}$ are defined as the smallest relations over W_{Σ_1} satisfying the following conditions (we write $\xrightarrow{\mu}$ for R_μ):

($\rightarrow$ 1) $\mu(p) \xrightarrow{\mu} p$.

($\rightarrow$ 2) If $p \xrightarrow{\mu} p'$, then $p + q \xrightarrow{\mu} p'$.

($\rightarrow$ 3) If $q \xrightarrow{\mu} q'$, then $p + q \xrightarrow{\mu} q'$.

Thus, the s-experiments ($s \in M^*$) possible for the program p above are the paths of the following tree:



We now proceed to examine the observational equivalence $\sim$ derived from the experiment relations R_μ determined above and its associated congruence $\sim_c$. First, we note that in this particularly simple case $\sim$ itself is indeed a congruence, and therefore identical with $\sim_c$.

PROPOSITION

(1) *$\sim$ is a congruence relation over W_{Σ_1}.*

(2) *$\sim$ is identical with $\sim_c$ over W_{Σ_1}.*

PROOF

(1) It is only necessary to show that $p_1 \sim p_2$ implies $\mu(p_1) \sim \mu(p_2)$ and that $p_1 \sim p_2$, $q_1 \sim q_2$ imply $p_1 + q_1 \sim p_2 + q_2$. The details are straightforward in terms of the definition of $\sim$ in Section 2.

(2) $\sim_c$ is the largest congruence included in $\sim$, which is clearly $\sim$ itself. □

Second, we look for simple "equational" properties of the congruence. It turns out that $(W_{\Sigma_1}/\sim_c, +, \text{NIL})$ is an Abelian monoid, with absorption.

PROPOSITION. *The following hold for all* $p_1, p_2, p_3 \in W_{\Sigma_1}$:

(1) $p_1 + (p_2 + p_3) \sim_c (p_1 + p_2) + p_3$,
(2) $p_1 + p_2 \sim_c p_2 + p_1$,
(3) $p_1 + p_1 \sim_c p_1$,
(4) $p_1 + NIL \sim_c p_1$.

PROOF. In each case, denoting the left and right sides by q_1, q_2 we can show from the definition of $\xrightarrow{\mu}$ that $q_1 \xrightarrow{\mu} q$ iff $q_2 \xrightarrow{\mu} q$; the result then follows directly. □

Note that the distributive law $\mu(p_1 + p_2) \sim_c \mu(p_1) + \mu(p_2)$ fails. For consider the two programs

$$p_1 = \mu_1(\mu_2(\text{NIL}) + \mu_3(\text{NIL})),$$
$$p_2 = \mu_1(\mu_2(\text{NIL})) + \mu_1(\mu_3(\text{NIL})).$$

We have $p_1 \xrightarrow{\mu_1} \mu_2(\text{NIL}) + \mu_3(\text{NIL})$, whereas $p_2 \xrightarrow{\mu_1} \mu_2(\text{NIL})$ and $p_2 \xrightarrow{\mu_1} \mu_3(\text{NIL})$; neither of the two successors of p_2 under $\xrightarrow{\mu_1}$ is equivalent to the successor of p_1.

The first property proved in the proposition justifies the use of the following convenient notation; we write

$$\sum_{1 \le i \le n} \mu_i p_i \quad \text{in place of} \quad \begin{cases} \mu_1 p_1 + \cdots + \mu_n p_n & \text{if } n > 0, \\ \text{NIL} & \text{if } n = 0, \end{cases}$$

knowing that the notation is unambiguous up to $\sim_c$. (From now on, μp stands for $\mu(p)$.) Moreover, the first and fourth properties allows us to assume that any program p can be expressed (up to $\sim_c$) in the form $\sum_{1 \le i \le n} \mu_i p_i$, where each p_i is again of the same form. We shall call such an expression a normal form for p.

This normalization is a necessary tool in proving the main result of the present section, namely that the four "equations" of the last proposition are complete, in the sense that any other valid equation between programs may be derived from them. For we have the following completeness result:

THEOREM 3.1. *The observational congruence* $\sim_c$ *over* W_{Σ_1} *is exactly the congruence induced by the four axioms*

(*A*1) $x + (y + z) = (x + y) + z$,
(*A*2) $x + y = y + x$,
(*A*3) $x + x = x$,
(*A*4) $x + NIL = x$.

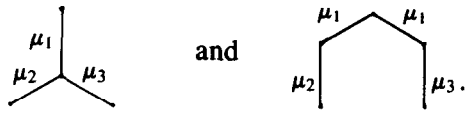
PROOF. Let $\equiv_1$ be the congruence over W_{Σ_1} induced by (A1)–(A4). By the previous proposition, $\sim_c$ satisfies these four axioms, whence $p \equiv_1 q$ implies $p \sim_c q$.

We first prove the converse for normal forms p and q, by induction on their structure, noting that $p \xrightarrow{\mu} p'$ implies that p' is a subterm of p. We therefore assume that p and q take the forms $\sum_m u_i p_i$ and $\sum_n v_j q_j$.

Assume that $p \sim_c q$. Since $p \xrightarrow{\mu_i} p_i$, then, for some q', $q \xrightarrow{\mu_i} q'$ and $p_i \sim_c q'$. But q' must be q_j for some j, with $\nu_j = \mu_i$, and by induction $p_i \equiv_1 q_j$. So such a j must exist for each i, and by symmetry for each j there exists i such that $p_i \equiv_1 q_j$ also. It then follows from axioms (A1)–(A3) that $p \equiv_1 q$.

Finally, in the case that p and q are arbitrary programs, it is enough to note that the four axioms allow any program to be proved congruent ($\equiv_1$) to its normal form. □

In view of our axioms, then, it is intuitively clear that $W_{\Sigma_1}/\sim_c$ is isomorphic with the set of rooted, unordered finite trees whose arcs are labeled by members of M, with the extra requirement that no two identically labeled arcs from a node lead to identical subtrees. As an example, the two programs p_1, q_1 are represented by the distinct trees



Indeed, we may use our language $\mathscr{L}$ of Section 2 to show that these two programs are not congruent (or even equivalent), for in terms of $\mathscr{L}$ we have

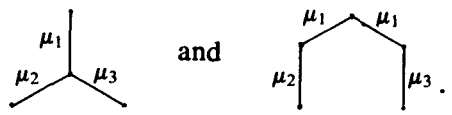
$$p_1 \models A_1, \qquad q_1 \not\models A_1,$$

where A_1 is $\langle\mu_1\rangle(\langle\mu_2\rangle T \wedge \langle\mu_3\rangle T)$.

Theorem 3.1 and the propositions leading up to it set the pattern for the remaining five algebraic characterizations of congruences treated in the paper, though the details usually are more difficult.

3.2. EXAMPLES. This section is devoted entirely to examples that substantiate the remarks at the end of Section 2.2. We use W_{Σ_1} as processes, whose elements are described by trees, and the experiment relations as given by the rules ($\rightarrow$ 1), ($\rightarrow$ 2), ($\rightarrow$ 3).

Example 1. Let p_1, q_1 denote the programs

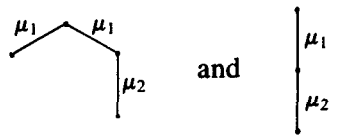


We have seen in Section 3.1 that

$$p_1 \models A_1, \qquad q_1 \not\models A_1,$$

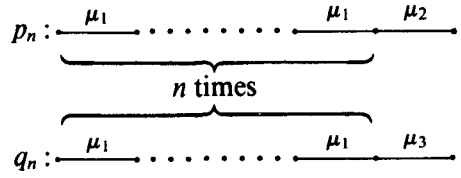
where A_1 is the formula $\langle\mu_1\rangle(\langle\mu_2\rangle T \wedge \langle\mu_3\rangle T)$. Now $A_1 \in \mathscr{M}$, so $p_1 \sim_{\mathscr{M}} q_1$. On the other hand one can show, by induction on A, that if $A \in \mathscr{N}$, then $p_1 \models A$ if and only if $q_1 \models A$. It follows that $p_1 \sim_{\mathscr{N}} q_1$ so that $\sim_{\mathscr{M}}$ is strictly stronger than $\sim_{\mathscr{N}}$. □

Example 2. Let p_2, q_2 denote the programs



If $A \in \mathscr{M}$, then $p_2 \models A$ if and only if $q_2 \models A$. This depends on the fact that if $A \in \mathscr{N}$ and NIL $\models A$ then $p \models A$ for every program p. Thus, $p_2 \sim_{\mathscr{M}} q_2$. On the other hand if A_2 denotes $\langle\mu_2\rangle \neg \langle\mu_2\rangle T$, then $p_2 \models A_2$ and $q_2 \not\models A_2$. From Theorem 2.2 it follows that $p_2 \not\sim q_2$. So $\sim$ is strictly stronger than $\sim_{\mathscr{M}}$. □

Example 3. We now show that the sequences $\sim_n$ and $\underset{\sim}{\sqsubset}_n$, $n \geq 0$, of relations are strictly decreasing. For let the pairs p_n, q_n be defined as follows, for each $n \geq 0$:



Then it is simple to prove that $p_n \sim_n q_n$ and $p_n \underset{\sim}{\sqsubset}_n q_n$, but $p_n \not\sim_{n+1} q_n$ and $p_n \not\underset{\sim}{\sqsubset}_{n+1} q_n$, for each $n \geq 0$. □

Turning to the formulas of $\mathscr{L}$, we remark that for each $n \geq 0$, the relation $\sim_n$ is characterized by the sublanguage $\mathscr{L}_n$ of $\mathscr{L}$ consisting of formulas with nesting at most n of the modal operators $\langle\mu\rangle$; that is, $p \sim_n q$ iff $\mathscr{L}_n(p) = \mathscr{L}_n(q)$. This is in fact shown, by induction on n, in our proof of Theorem 2.2 given in Appendix A.

However, Example 3 leaves something to be desired. For it shows that the weakness of each $\sim_n$ or $\mathscr{L}_n$ follows, in part, from its inability to "examine" a program's behavior beyond its first n actions.

This weakness can be remedied as follows. We may consider, in place of the experiment relations $\xrightarrow{\mu}$, the derived relations $\xrightarrow{s}$ for each $s \in M^*$, where

$$p \xrightarrow{\mu_1 \cdots \mu_k} q \quad \text{iff} \quad p \xrightarrow{\mu_1} \cdots \xrightarrow{\mu_k} q.$$

We may then define a map E^* of relations over P as follows:

$\langle p, q\rangle \in E^*(S)$ if, for all $s \in M^*$,

(i) $p \xrightarrow{s} p'$ implies, for some q', $q \xrightarrow{s} q'$ and $\langle p', q'\rangle \in S$,
(ii) $q \xrightarrow{s} q'$ implies, for some p', $p \xrightarrow{s} p'$ and $\langle p', q'\rangle \in S$.

Then we take $\sim_n^* = E^{*n}(P \times P)$, and $\sim^* = \bigcap_n \sim_n^*$. Now each $\sim_n^*$, even $\sim_1^*$, can "examine" the behavior of programs arbitrarily far into the future. In fact, $\sim_1^*$ is already quite strong; we can state that

$$\sim_1^* = \sim_e.$$

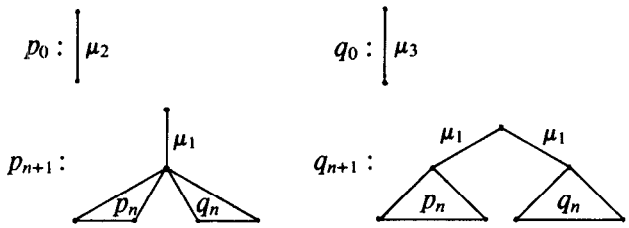
In other words, $p \sim_1^* q$ iff p and q have exactly the same action sequences. On the other hand, in the limit we can show that

$$\sim^* = \sim.$$

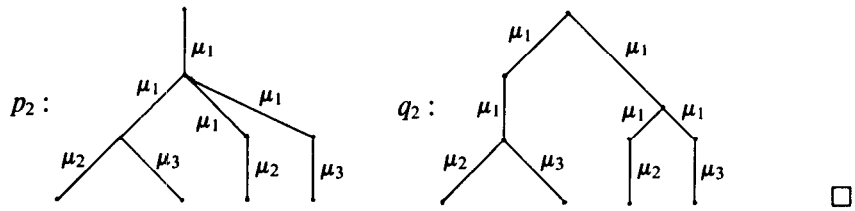
In other words, we have yet another characterization of the observation equivalence.

But is it necessary to proceed to the limit, setting $\sim^* = \bigcap_n \sim_n^*$, or do we have $\sim^* = \sim_n^*$ already for some finite n? If the latter were so, then our "recursive definition" of $\sim$ would have been misleading. But we can indeed show that the sequence of relations $\sim_n^*$, $n \geq 0$, is also strictly decreasing. For this we need a more complex sequence of program pairs than Example 3.

Example 4. Let the pairs p_n, q_n of programs be defined as follows, for each $n \geq 0$:



Note that p_1, q_1 are the programs of Example 1 above. Note also that $p_n \sim_e q_n$ for all $n \geq 1$, since p_n and q_n have the same action sequences. But, as in Example 3, we are able to show (we omit the proof) that $p_n \sim_n^* q_n$ but $p_n \not\sim_{n+1}^* q_n$ for each n. The reader may care to verify this at $n = 2$:



□

Now, finally, it should be clear that if we take $\mathscr{L}^*$ to be the formulas defined as for $\mathscr{L}$ but with modal operators $\langle s \rangle$, $s \in M^*$, then Theorem 2.2 yields

$$p \sim^* q \quad \text{iff} \quad \mathscr{L}^*(p) = \mathscr{L}^*(q).$$

This is no surprise, since $\sim^*$ and $\sim$ are identical and since $\mathscr{L}^*$ is already a derived language of $\mathscr{L}$, by setting

$$\langle s \rangle = \langle \mu_1 \rangle \cdots \langle \mu_k \rangle \quad \text{for} \quad s = \mu_1 \cdots \mu_k.$$

But we also have a characterization of each $\sim_n^*$:

$$p \sim_n^* q \quad \text{iff} \quad \mathscr{L}_n^*(p) = \mathscr{L}_n^*(q)$$

where $\mathscr{L}_n^*$ is the sublanguage of $\mathscr{L}$ in which the modal operators $\langle s \rangle$ may be nested to depth at most n. Thus, arbitrary depth of nesting is required in $\mathscr{L}^*$, even for its more powerful modal operators, in order to characterize $\sim^*$ fully. Clearly a simple nesting of the form $\langle s \rangle \langle s' \rangle$ is no more powerful than the single operator $\langle ss' \rangle$; it is the interleaving of propositional and modal operators that adds power. Indeed, we saw in Section 2.2 how the alternation of $\langle \rangle$ with its dual $[]$ allowed the expression of complex properties concerned with deadlock.

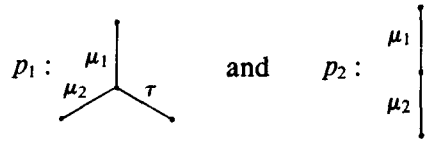
3.3. UNOBSERVABLE ATOMIC ACTIONS IN Σ_1. In the system of Section 3.1, every atomic action is observable; a program cannot proceed without being observed. Let us now suppose that among M there are atomic actions that cannot be observed; such an atomic action had no corresponding atomic experiment. For the moment we are not concerned with how these actions may arise. In the next section we introduce a notion of communication, and then unobservable actions will arise from internal communications between subprocesses of the process being observed. However, we can analyze the effect of their presence on the observable behavior of a program independently of saying how they arise.

That their presence does indeed have an effect on observable behavior may be seen from the following example.

Example. Suppose that τ is an unobservable atomic action, and consider the programs

$$p_1 = \mu_1(\mu_2(\mathrm{NIL}) + \tau(\mathrm{NIL})),$$
$$p_2 = \mu_1(\mu_2(\mathrm{NIL})).$$

Using the remarks at the end of Section 3.1, these may be represented by the trees:



When an atomic μ_1-experiment is performed on p_1, one possible result is that it changes p_1 into NIL. This follows because the execution of the action μ_1 by the program p_1 may be followed by the execution of the unobservable τ action. Intuitively, this is an acceptable μ_1-experiment since of this sequence of actions performed by the program the observer only sees μ_1. However, the only possible result of performing a μ_1-experiment on p_2 is the program $\mu_2(\mathrm{NIL})$. It follows that p_1 is not observationally equivalent to p_2. This inequivalence may also be seen using the language $\mathscr{L}$ since

$$p_1 \models A, \qquad p_2 \not\models A,$$

where A is $\langle\mu_1\rangle(\neg\ \langle\mu_2\rangle T)$. □

For simplicity we assume that τ is the only unobservable atomic action. (This may be formally justified; if there were two such, τ_1 and τ_2, we would arrive at an axiom $\tau_1(x) = \tau_2(x)$—indicating that the replacement of τ_1 by τ_2 can affect no observation.) We therefore assume $M = \Lambda \cup \{\tau\}$ ($\tau \notin \Lambda$), and we define a new set $\{R_\lambda \mid \lambda \in \Lambda\}$ of experiment relations as follows. First, define $\xrightarrow{s}$ over W_{Σ_1}, for any $s = \mu_1 \cdots \mu_n \in M^*(n \geq 0)$, by

$$p \xrightarrow{s} p' \qquad \text{iff} \qquad p = p_0 \xrightarrow{\mu_1} p_1 \xrightarrow{\mu_2} \cdots \xrightarrow{\mu_n} p_n = p'.$$

Then, writing R_λ as $\overset{\lambda}{\Longrightarrow}$, we define for each $\lambda \in \Lambda$

$$p \overset{\lambda}{\Longrightarrow} p' \qquad \text{iff} \qquad p \xrightarrow{\tau^m \lambda \tau^n} p' \qquad \text{for some} \quad m, n \geq 0.$$

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

We obtain now a new observational equivalence relation $\approx$ over W_{Σ_1}, using the definition of Section 2, with the relations $\left\{\overset{\lambda}{\Longrightarrow} \middle| \lambda \in \Lambda\right\}$. 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 in not a congruence. For example, it is easy to check that $\tau(\mathrm{NIL}) \approx \mathrm{NIL}$; but if we place each of these programs in the context $\mathscr{L}[\ ] = \lambda_1(\lambda_2(\mathrm{NIL}) + [\ ])$ we obtain $\mathscr{L}[\tau(\mathrm{NIL})] \not\approx \mathscr{L}[\mathrm{NIL}]$ as may be readily checked (this is in effect the pair $p_1 p_2$ discussed earlier).

The fact that $\approx$ is not identical to $\approx_c$ makes the latter more difficult to analyze than the congruence $\sim_c$ of Section 3.1. However, it is easy to show that $\approx_c$ distinguishes no more programs than $\sim_c$.

PROPOSITION. *For all* $p_1, p_2 \in W_{\Sigma_1}$

(*a*) $p_1 \sim p_2$ *implies* $p_1 \approx p_2$.
(*b*) $p_1 \sim_c p_2$ *implies* $p_1 \approx_c p_2$.

PROOF

(a) A simple induction on n will show that $\sim \subseteq \approx_n$.
(b) We have $\sim_c \subseteq \sim \subseteq \approx$. Hence, since $\approx_c$ is the largest congruence contained in $\approx$, it follows that $\sim_c \subseteq \approx_c$. □

From this proposition it immediately follows that $\approx_c$ satisfies all the axioms that characterize $\sim_c$.

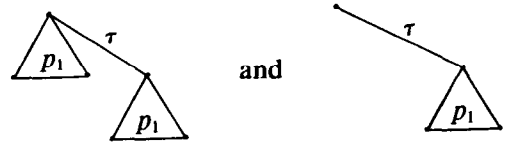
In addition, it enjoys some properties that indicate that certain occurrences of τ may be eliminated from programs.

PROPOSITION. *The following hold for all* $p_1, p_2 \in W_{\Sigma_1}$:

(*a*) $p_1 + \tau p_1 \approx_c \tau p_1$.
(*b*) $\mu(p_1 + \tau p_2) \approx_c \mu(p_1 + p_2) + \mu p_2$.

A direct proof of any one of these properties would involve consideration of the effect an arbitrary context can have on the terms involved. To avoid this we give in Appendix C an alternative characterization of $\approx_c$ that is much easier to deal with. The proof of this proposition then becomes routine.

One may motivate the new properties by seeing how an observer might attempt to distinguish between the two programs in each case. For example, the programs in (a) may be represented as



From these trees it can be seen intuitively that the extra p_1-subtree on the left-hand side does not change its $\stackrel{\lambda}{\Rightarrow}$ experiments because τ is unobservable. Such arguments, however, are fraught with danger and should be treated carefully.

A somewhat surprising result is that these two additional properties are sufficient to characterize the new observational congruence.

THEOREM 3.2. *The observational congruence* $\approx_c$ *over* Σ_1 *is exactly the congruence induced by axioms* (*A*1)–(*A*4) *and*

(*A*5) $x + \tau x = \tau x$,
(*A*6) $\mu(x + \tau y) = \mu(x + y) + \mu y$.

This theorem is not so immediate as Theorem 3.1, partly because $\approx$ is not a congruence. It involves defining a normal form for programs in W_{Σ_1}; the most important step in deriving the normal form is the use of (A6) to eliminate most occurrences of τ in a program.

4. *Application to a Simple Language for Communication*

4.1. EXTENSION OF THE SIGNATURE. We now extend Σ_1 to the signature Σ_2 by adding a binary operator " | "; it is one of a variety of operators that may be chosen to represent the combination of a pair of programs that may proceed concurrently

and may also communicate with one another. These two properties are reflected by separate new conditions upon the experiment relations $\xrightarrow{\mu}$. One condition (in two parts) states that the program $p \mid q$ admits all the experiments that p and q admit separately. (Since an atomic experiment corresponds to a single atomic action, the simultaneous activity of p and q cannot be observed.)

($\rightarrow$ 4) If $p \xrightarrow{\mu} p'$, then $p \mid q \xrightarrow{\mu} p' \mid q$.

($\rightarrow$ 5) If $q \xrightarrow{\mu} q'$, then $p \mid q \xrightarrow{\mu} p \mid q'$.

The next condition upon the relations $\xrightarrow{\mu}$ expresses the capability of p and q to communicate, in the case that two actions—one by p and one by q—complement each other. We take the view that two such actions occurring simultaneously appear to an external observer as a single, unobservable action τ.

To handle the notion of complementary actions, we introduce a little structure over M. We assume $M = \Lambda \cup \{\tau\}$ as before, and also that $\Lambda = \Delta \cup \bar{\Delta}$ where Δ is a possibly infinite alphabet of *names*, and that the alphabet $\bar{\Delta}$ of *conames* is disjoint from Δ and in bijection with it. We represent the bijection and its inverse by an overbar ($\bar{\ }$), and use $\{\alpha, \beta, \gamma\}$ to range over Δ. Thus $\bar{\alpha} \in \bar{\Delta}$, and $\bar{\bar{\alpha}} = \alpha$. We continue to use λ to range over Λ, and μ, ν to range over $M = \Lambda \cup \{\tau\}$. Communication between p and q may occur when p admits a λ-experiment and q admits a $\bar{\lambda}$-experiment, for some λ; the result is a τ-action of $p \mid q$.

($\rightarrow$ 6) If $p \xrightarrow{\lambda} p'$ and $q \xrightarrow{\bar{\lambda}} q'$, then $p \mid q \xrightarrow{\tau} p' \mid q'$.

Now taking $\{\xrightarrow{\mu} \mid \mu \in M\}$ to be the smallest relations over W_{Σ_2} satisfying ($\rightarrow$ 1)–($\rightarrow$ 6), we obtain an observational equivalence $\sim$ over W_{Σ_2} as in Section 3.1. As before, this turns out to be a congruence, so that $\sim_c$ is identical with $\sim$.

Let us now examine properties of $\sim_c$ with respect to the new binary composition operator. Intuitively, the behavior of $p \mid q$ is as follows. Its possible first actions are just those of p independently, those of q independently, and those τ actions resulting from complementary pairs of actions by p and q; after such a first action, p and q continue to act in parallel.

To express this as an equation, we use the notation $\sum \mu_i p_i$ introduced in Section 3.1.

PROPOSITION. *If p is $\sum \mu_i p_i$ and q is $\sum \nu_j q_j$, then*

$$p \mid q \sim_c \sum_i \mu_i (p_i \mid q) + \sum_j \nu_j (p \mid q_j) + \sum_{\mu_i = \bar{\nu}_j} \tau (p_i \mid q_j).$$

A few simple examples makes the proposition clear. Note particularly that, in the second example, the action β may either occur independently or be complemented by the action $\bar{\beta}$; we shall see later how the complementation can—by application of a further operator called *restriction*—be forced to occur, so that the name β is used solely for communication between p and q in the composite $p \mid q$.

Examples

$$(\alpha p_1 + \beta p_2) \mid \gamma q = \alpha (p_1 \mid \gamma q) + \beta (p_2 \mid \gamma q) + \gamma ((\alpha p_1 + \beta p_2) \mid q),$$
$$(\alpha p_1 + \beta p_2) \mid \bar{\beta} q = \alpha (p_1 \mid \bar{\beta} q) + \beta (p_2 \mid \bar{\beta} q) + \bar{\beta} ((\alpha p_1 + \beta p_2 \mid q) + \tau (p_2 \mid q),$$
$$\left(\sum \mu_i p_i\right) \mid \mathrm{NIL} = \sum \mu_i (p_i \mid \mathrm{NIL}) + \mathrm{NIL} + \mathrm{NIL}.$$

Note that the proposition allows " | " to be eliminated, by stages, from any word in W_{Σ_2}. In fact, it is this property of " | " that allows us to prove that the proposition,

taken as an axiom schema, is the only interesting property of "|" with respect to $\sim_c$.

(A7) For any words u and v of form $\sum \mu_i x_i$, $\sum \nu_j y_j$:

$$u \mid v = \sum_i \mu_i (x_i \mid v) + \sum_j \nu_j (u \mid y_j) + \sum_{\mu_i = \bar{\nu}_j} \tau (x_i \mid y_j).$$

Now we may state another theorem of complete axiomization.

THEOREM 4.1. *The observational congruence $\sim_c$ over W_{Σ_2} is exactly the congruence induced by (A1)–(A4) and (A7).*

Remark. The following laws for "|" may be proved to hold over W_{Σ_2} by induction on the structure of terms (though they are not deducible from (A1)–(A4), (A7) by equational reasoning):

$$x \mid (y \mid z) = (x \mid y) \mid z,$$
$$x \mid y = y \mid x,$$
$$x \mid \mathrm{NIL} = x.$$

4.2. UNOBSERVABLE ACTIONS IN Σ_2. We now repeat for Σ_2 what we did for Σ_1; we wish to treat τ as an unobservable atomic action (in particular, the intercommunication of p and q in $p \mid q$ is not an observable action). If we define the experiment relations $\{\overset{\lambda}{\Rightarrow} \mid \lambda \in \Lambda\}$ as we did previously, then we gain an observational congruence $\approx_c$ over W_{Σ_2} again. We 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 $\approx_c$ over W_{Σ_2}. The reason is that, although one side of (A6) may be replaced by the other in any context built from Σ_1, preserving observational equivalence, there are Σ_2 contexts built using "|" in which the replacement does not preserve the equivalence. In fact, we shall demonstrate in particular that the following instance of (A6) is false:

$$\alpha(\beta\mathrm{NIL} + \tau\mathrm{NIL}) \approx_c \alpha(\beta\mathrm{NIL} + \mathrm{NIL}) + \alpha\mathrm{NIL}.$$

For this would imply the observational equivalence

$$\gamma\mathrm{NIL} \mid \alpha(\beta\mathrm{NIL} + \tau\mathrm{NIL}) \approx \gamma\mathrm{NIL} \mid (\alpha(\beta\mathrm{NIL} + \mathrm{NIL}) + \alpha\mathrm{NIL}). \tag{1}$$

Calling the left and right sides of (1) p and q, respectively, we have

$$p \overset{\alpha}{\Rightarrow} p' = \gamma\mathrm{NIL} \mid (\beta\mathrm{NIL} + \tau\mathrm{NIL}),$$

whereas $q \overset{\alpha}{\Rightarrow} q'$ implies that $q' = q_1$ or $q' = q_2$ where

$$q_1 = \gamma\mathrm{NIL} \mid (\beta\mathrm{NIL} + \mathrm{NIL}),$$
$$q_2 = \gamma\mathrm{NIL} \mid \mathrm{NIL}.$$

Now if (1) holds, then by definition of $\approx$ we must have $p' \approx q_1$ or $p' \approx q_2$. The second is impossible since $p' \overset{\beta}{\Rightarrow} \gamma\mathrm{NIL} \mid \mathrm{NIL}$ whereas $q_2 \overset{\beta}{\Rightarrow} q_2'$ is impossible. Hence $p' \not\approx q_2$. On the other hand, we may also show $p' \not\approx q_1$. Since

$$p' \overset{\gamma}{\Rightarrow} \mathrm{NIL} \mid \mathrm{NIL},$$

whereas the only γ-experiment for q_1 is

$$q_1 \overset{\gamma}{\Rightarrow} \mathrm{NIL} \mid (\beta\mathrm{NIL} + \mathrm{NIL}),$$

it is easily seen that $\mathrm{NIL} \mid \mathrm{NIL} \not\approx \mathrm{NIL} \mid (\beta\mathrm{NIL} + \mathrm{NIL})$.

We therefore look for a set of axioms weaker than (A1)–(A7) that characterize $\approx_c$ over W_{Σ_2}. Fortunately, it turns out that only (A6) need be replaced; (A1)–(A5) and (A7) are found to be satisfied by $\approx_c$ over W_{Σ_2}. Our replacement for (A6) is two new axioms:

$$\left.\begin{array}{ll}(\text{A6.1}) & \mu(x + \tau y) = \mu(x + \tau y) + \mu y, \\ (\text{A6.2}) & \mu\tau y = \mu y,\end{array}\right\} (\mu \in M).$$

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 τy for y in (A6):

$$\mu(x + \tau\tau y) = \mu(x + \tau y) + \mu\tau y,$$

and use two instances of (A6.2).

THEOREM 4.2. *The observational congruence $\approx_c$ over W_{Σ_2} 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 generalizes in a routine manner to the corresponding theorem for our next signature Σ_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 Σ_1 in the presence of extra operators for communication and concurrency.

5. *Further Operators on Programs*

In the preceding sections we have dealt with the main technical results of the present paper. This requires only slight extension to cover the operators of CCS [6], and we present the required extension in this section.

In [4] we considered operators over behaviors corresponding to Σ_2, together with two other families of operators called *relabeling* and *restriction*; in the present context, these operators may be described as changing (bijectively) the labels for atomic experiments (i.e., permutations of Λ), and restricting the class of atomic experiments to a subset of Λ. The approach in [6] was to classify behaviors into *sorts*; a sort L was a subset of Λ, and the behaviors B_L of sort L were those that 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 relabeling and restriction as subclasses of a wider family of operators indexed by a subset of the partial functions $M \rightharpoonup M$ from M to M. To this end we extend Σ_2 to the signature Σ_3 by adding operators

$$\mathscr{S} = \{[S] \mid S \in M \rightharpoonup M,\ S\tau = \tau\}.$$

We shall postfix these operators. We characterize them operationally by adding a further condition for the experiment relations $\xrightarrow{\mu}$:

($\rightarrow$ 7) If $p \xrightarrow{\mu} p'$ and $S\mu$ is defined, then $p[S] \xrightarrow{S\mu} p'[S]$.

Now we take $\{\xrightarrow{\mu} \mid \mu \in M\}$ to be the smallest relations over W_{Σ_3} satisfying ($\rightarrow$ 1)–($\rightarrow$ 7), and again obtain an observational equivalence $\sim$ over W_{Σ_3}, which is a congruence, so that again $\sim_c$ is identical with $\sim$.

(A1) $x + (y + z) = (x + y) + z$
(A2) $x + y = y + x$
(A3) $x + x = x$
(A4) $x + \mathrm{NIL} = x$
(A5) $x + \tau x = \tau x$
(A6) $\mu(x + \tau y) = \mu(x + y) + \mu y$
(A6.1) $\mu(x + \tau y) = \mu(x + \tau y) + \mu y$
(A6.2) $\mu\tau y = \mu y$
(A7) if u is $\Sigma\mu_i x_i$ and v is $\Sigma\nu_j y_j$, then $u \mid v = \Sigma\mu_i (x_i \mid v) + \Sigma\nu_j (u \mid y_j) + \Sigma_{\mu=\bar{\nu}_j} \tau(x_i \mid y_j)$
(A8) $(\mu x)[S] = S\mu(x[S])$ if $S\mu$ defined
$= \mathrm{NIL}$ otherwise
(A9) $(x + y)[S] = x[S] + y[S]$
(A10) $\mathrm{NIL}[S] = \mathrm{NIL}$

FIGURE 1

TABLE I. RELATIONSHIP BETWEEN AXIOMS AND CONGRUENCES

Signature	Axioms for $\sim_c$	Axioms for $\approx_c$
$\Sigma_1 = M \cup \{\mathrm{NIL}, +\}$	—	(A5), (A6)
$\Sigma_1 = \Sigma_1 \cup \{\mid\}$	(A7)	(A5), (A6.1), (A6.2), (A7)
$\Sigma_3 = \Sigma_2 \cup \mathscr{S}$	(A7)–(A10)	(A5), (A6.1), (A6.2), (A7)–(A10)

The axioms needed to characterize $\mathscr{S}$ are the obvious ones:

(A8) $(\mu x)[S] = S\mu(x[S])$ if $S\mu$ is defined, NIL otherwise;
(A9) $(x + y)[S] = x[S] + y[S]$;
(A10) $\mathrm{NIL}[S] = \mathrm{NIL}$.

THEOREM 5.1. *The observational congruence $\sim_c$ over W_{Σ_3} is exactly the congruence induced by (A1)–(A4) and (A7)–(A10).*

The treatment of experiment relations $\{\overset{\lambda}{\Longrightarrow} \mid \lambda \in \Lambda\}$ and the corresponding observational congruence $\approx_c$ over W_{Σ_3} is exactly as it was for W_{Σ_2}, and by trivially adapting the proof of Theorem 4.2 we obtain

THEOREM 5.2. *The observational congruence $\approx_c$ over W_{Σ_3} is exactly the congruence induced by (A1)–(A5), (A6.1), (A6.2), and (A7)–(A10).*

6. *Conclusions*

We have characterized observational congruence in six cases by equational axioms. There are three signatures, $\Sigma_1 \subseteq \Sigma_2 \subseteq \Sigma_3$, with Σ_3 being a minor variant of the signature used in the language CCS [6]. For each of those cases, two classes of experiments relations are considered: $\{\overset{\mu}{\longrightarrow} \mid \mu \in M\}$ when the atomic action τ is observable, and $\{\overset{\lambda}{\Longrightarrow} \mid \lambda \in \Lambda\}$ when τ is not directly observable buy may "occur" a finite number of times during any atomic experiment. The set of axioms used in the paper is given in Figure 1. The correspondence between the axioms and various observational congruences may be tabulated as shown in Table I ((A1)–(A4) are needed in every case). 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 " $\mid$ ", and that this replacement persists with the addition of any reasonable family of partial relabeling operators (even multivalued ones, though we restricted consideration to single-valued relabeling).

The following Appendixes provide detailed proofs of the theorems.

Appendix A. Proofs of General Results on Observational Equivalence

This Appendix is devoted to the proofs of Theorems 2.1 and 2.2. We assume the notations and definitions introduced in Section 2.

THEOREM 2.1. *If each R_i is image-finite, then $\sim$ is the maximal solution to $S = E(S)$.*

PROOF.

(i) $\sim \subseteq E(\sim)$
Suppose $p \sim q$ and $\langle p, p' \rangle \in R_i$. Then for each n there exists q_n such that $p' \sim_n q_n$ and $\langle q, q_n \rangle \in R_i$. Since R_i is image finite, there exists q' such that $q' = q_n$ for infinitely many n. But $\sim_n$ is decreasing in n, hence $p' \sim_n q'$ for all n, and so $p' \sim q'$; also $\langle q, q' \rangle \in R_i$. By symmetry it follows that $\langle p, q \rangle \in E(\sim)$; hence $\sim \subseteq E(\sim)$ since p and q were arbitrary.

(ii) $E(\sim) \subseteq \sim$
We prove by induction on n that $\langle p, q \rangle \in E(\sim) \Rightarrow p \sim_{n+1} q$. Let $\langle p, q \rangle \in E(\sim)$ and $\langle p, p' \rangle \in R_i$. Then $\langle q, q' \rangle \in R_i$ and $p' \sim q'$, for some q'. From (i), it follows that $\langle p', q' \rangle \in E(\sim)$. By induction, $p' \sim_n q'$. Similarly, if $\langle q, q' \rangle \in R_i$, there exists p' such that $\langle p, p' \rangle \in R_i$ and $p' \sim_n q'$. Therefore, $p \sim_{n+1} q$.

(iii) Let S be any relation such that $S = E(S)$. We prove by induction on n that $\langle p, q \rangle \in S \Rightarrow p \sim_{n+1} q$. Let $\langle p, q \rangle \in S$ and $\langle p, p' \rangle \in R_i$. Then $\langle p, q \rangle \in E(S)$. So $\langle q, q' \rangle \in R_i$ and $\langle p', q' \rangle \in S$, for some q'. By induction, $p' \sim_n q'$. From symmetry it follows that $p \sim_{n+1} q$. Therefore, $\langle p, q \rangle \in S \Rightarrow p \sim q$. □

THEOREM 2.2. *If each R_i is image-finite, then $p \sim q$ if and only if $\mathscr{L}(p) = \mathscr{L}(q)$.*

PROOF. Let $\mathscr{L}_n \subseteq \mathscr{L}$ be the class of formulas with depth at most n of "modal" operators $\langle i \rangle$. Let $\mathscr{L}_n(p) = \{A \in \mathscr{L}_n \mid p \models A\}$. To establish the theorem it is sufficient to show that $p \sim_n q$ if and only if $\mathscr{L}_n(p) = \mathscr{L}_n(q)$. We use induction on n.

(i) $n = 0$.
For any p and $A \in \mathscr{L}_0$, $p \models A$ iff A is logically equivalent to T. So $\mathscr{L}_0(p) = \mathscr{L}_0(q)$ for every p, q and the result follows since $p \sim_0 q$ for every p, q.

(ii) $p \sim_{n+1} q$ implies $\mathscr{L}_{n+1}(p) = \mathscr{L}_{n+1}(q)$.
We show by structural induction on $A \in \mathscr{L}_{n+1}$ that, if $p \sim_{n+1} q$ and $p \models A$, then $q \models A$. Let $p \models A$. If A is T, the result is trivial. Consider now the other cases for A.

(a) A is $\langle i \rangle B$, $B \in \mathscr{L}_n$.
Then $\langle p, p' \rangle \in R_i$ with $p' \models B$, for some p'. Since $p \sim_{n+1} q$, there exists q' such that $\langle q, q' \rangle \in R_i$ and $p' \sim_n q'$. By induction on n, $q' \models B$. So $q \models A$.

(b) A is $\neg A'$.
Then not $p \models A'$. Now if $q \models A'$. Then by structural induction $p \models A'$, which is a contradiction. Therefore, $q \models \neg A'$.

(c) A is $A_1 \wedge A_2$.
Then $p \models A_1$ and $p \models A_2$. By structural induction, $q \models A_1$ and $q \models A_2$ and therefore $q \models A$.

(iii) $p \not\sim_{n+1} q$ implies $\mathscr{L}_{n+1}(p) \neq \mathscr{L}_{n+1}(q)$.
Since $p \not\sim_{n+1} q$, without loss of generality, we can assume that there is an i and p' such that $\langle p, p' \rangle \in R_i$, and $\langle q, q' \rangle \in R_i \Rightarrow p' \not\sim_n q'$. Since R_i is image-finite, let $\{q_1, \ldots, q_k\} = \{q' \mid \langle q, q' \rangle \in R_i\}$. By induction, $\mathscr{L}_n(p') \neq \mathscr{L}_n(q_i)$ for

each i, $1 \leq i \leq k$. So there are formulas $B_1, \ldots, B_n$ such that $p' \models B_i$ and not $q_i \models B_i$, $B_i \in \mathcal{L}_n$. Then $p' \models B$ and not $q_i \models B$, where B denotes $B_1 \wedge \cdots \wedge B_n$. Therefore, $p \models \langle\!\langle \rangle\!\rangle B$ and not $q \models \langle\!\langle \rangle\!\rangle B$, that is, $\mathcal{L}_{n+1}(p) \neq \mathcal{L}_{n+1}(q)$. □

Appendix B. Proofs of Results Concerning $\sim_c$

This Appendix is devoted to the proofs of the completeness of the axiomatizations of $\sim_c$ over the three signatures Σ_1, Σ_2, and Σ_3. Recall that these results deal with the case in which all actions, including τ, are observable.

The result for Σ_1, concerning axioms (A1)–(A4), was proved as Theorem 3.1. The other two theorems, 4.1 and 5.1, will be reduced to Theorem 3.1 by the following Extension Lemma.

Definition B1. Let $\Sigma \subseteq \Sigma'$; let R be a relation over W_Σ and R' over $W_{\Sigma'}$. Then R' is a *conservative extension* of R if $R' \cap W_\Sigma^2 \subseteq R$.

EXTENSION LEMMA. *Let $\Sigma \subseteq \Sigma'$, and let R and R' be equivalence relations over W_Σ and $W_{\Sigma'}$ such that $R \subseteq R'$. Let S be an equivalence relation over $W_{\Sigma'}$ such that*

(*i*) *S is a conservative extension of R,*
(*ii*) *$R' \subseteq S$,*
(*iii*) *For each t in $W_{\Sigma'}$, there exists a normal form $nf(t)$ in W_Σ such that $\langle t, nf(t)\rangle \in R'$. Then $R' = S$.*

PROOF. Suppose $\langle p, q\rangle \in S$. *Then,*

from (iii), $\langle p, nf(p)\rangle \in R'$ and $\langle q, nf(q)\rangle \in R'$;
from (ii), $\langle nf(p), nf(q)\rangle \in S$;
from (i), $\langle nf(p), nf(q)\rangle \in R$.

Therefore, $\langle p, q\rangle \in R'$ since $R \subseteq R'$. □

Let $\equiv_2$ be the congruence over W_{Σ_2} generated by the axioms (A1)–(A4) and (A7), and $\equiv_3$ over W_{Σ_3} by (A1)–(A4) and (A7)–(A10).

COROLLARY

(*a*) *Theorem* 4.1: *For $p, q \in W_{\Sigma_2}$, $p \equiv_2 q$ iff $p \sim_c q$.*
(*b*) *Theorem* 5.1: *For $p, q \in W_{\Sigma_3}$, $p \equiv_3 q$ iff $p \sim_c q$.*

PROOF. We prove (b) only, the proof of (a) being similar. We apply the Extension Lemma, with $\Sigma = \Sigma_1$, $\Sigma' = \Sigma_3$, $R = \equiv_1$, $R' = \equiv_3$, and S the observational congruence $\sim_c$ over W_{Σ_3}.

As for Theorem 3.1, we leave it to the reader to show that $\sim$ is a congruence and satisfies the axioms (A1)–(A4) and (A7)–(A10). This establishes hypothesis (ii) of the Extension Lemma. Also by using axioms (A7)–(A10) all occurrences of "|" and $[S]$ can be eliminated from terms in W_{Σ_3}. This establishes hypothesis (iii). It remains to show that $\sim$ is a conservative extension of $\equiv_1$. Let $\sim'$ be the observational equivalence over W_{Σ_1}, which uses as experiments the least relations satisfying $(\rightarrow 1)$, $(\rightarrow 2)$, and $(\rightarrow 3)$. From Theorem 3.1, $p \equiv_1 q$ iff $p \sim' q$. A simple proof by structural induction will establish that for all $p, q \in W_{\Sigma_1}$, $p \sim q$ implies $p \sim' q$. □

Appendix C. Proofs of Results Concerning $\approx_c$

This Appendix is mainly devoted to proving the principal result of the paper, Theorem 4.2, which deals with the axiomatization of $\approx_c$ over signature Σ_2; recall

that this differs from Theorem 4.1 in that τ is now unobservable. Section C1 deals with the soundness of the appropriate axioms and Section C2 with their completeness. The proofs of the analogous Theorems 3.2 and 5.2 for signatures Σ_1 and Σ_3 are outlined in Section C3.

C1. SOUNDNESS OF THE AXIOMATIZATION OF $\approx_c$ OVER SIGNATURE Σ_2. Let $\equiv$ denote the congruence over W_{Σ_2} induced by the axioms (A1)–(A5), (A6.1), (A6.2) and (A7).

We show that $p \equiv q$ implies $p \approx_c q$. First, generalize the experiment relations by defining: $p \stackrel{\tau}{\Rightarrow} q$ if $p \stackrel{\tau^n}{\longrightarrow} q$ for some $n > 0$. For $\mu \in \Lambda \cup \{\tau\}$ let $\mathrm{Der}_\mu(p) = \{q \mid p \stackrel{\mu}{\Rightarrow} q\}$. Now let $\approx'$ denote the maximal solution to the equation

$$S = E'(S),$$

where

$\langle p, q\rangle \in E'(S)$ if for all $\mu \in \Lambda \cup \{\tau\}$,

(i) $p' \in \mathrm{Der}_\mu(p)$ implies $\langle p', q'\rangle \in S$, for some $q' \in \mathrm{Der}_\mu(q)$.
(ii) $q' \in \mathrm{Der}_\mu(q)$ implies $\langle p', q'\rangle \in S$, for some $p' \in \mathrm{Der}_\mu(p)$.

The existence of $\approx'$ follows from Theorem 2.1. It must be observed that $\approx'$ differs from $\approx_c$; in particular it will not satisfy axiom (A6.2). But the following lemma is enough for our purpose:

LEMMA

(*a*) $\approx'$ *is a congruence over* W_{Σ_2}.
(*b*) $p \approx' q$ *implies* $p \approx_c q$.

PROOF

(a) By structural induction on terms.
(b) By structural induction, we can prove that $p \approx' q$ implies $p \approx q$. The result then follows from (a) since $\approx_c$ is the largest congruence contained in $\approx$. □

THEOREM 4.2, PART (i) (SOUNDNESS). *For* $p, q \in W_{\Sigma_2}$, $p \equiv q$ *implies* $p \approx_c q$.

PROOF. It is sufficient to show that if $a_1 = a_2$ is an instance of any axiom then $a_1 \approx_c a_2$. Let $a_1 = a_2$ be an instance of any axiom other than A6.2. In this case it is easily seen that for $\mu \in \Lambda \cup \{\tau\}$, $\mathrm{Der}_\mu(a_1) = \mathrm{Der}_\mu(a_2)$. It follows that $a_1 \approx' a_2$ and therefore by the lemma, $a_1 \approx_c a_2$. In case of an instance of A6.2, a simple proof by induction on Σ_2 contexts $\mathscr{C}[\,]$ will show that $\mathscr{C}[a_1] \approx \mathscr{C}[a_2]$. □

C2. COMPLETENESS OF THE AXIOMATIZATION OF $\approx_c$ OVER SIGNATURE Σ_2. In this section we show that $p \approx_c q$ implies $p \equiv q$.

Definition. p is a *sumform* if it is of the form $\sum \mu_i p_i$, where each p_i is a sumform. (Note that NIL is a sumform.)

Definition. $p \equiv_s q$ (*sumcongruence*) if $p \equiv q$ may be proved using (A1) and (A2) alone. That is, $\equiv_s$ is the congruence induced by (A1) and (A2).

ABSORPTION LEMMA. *If* $p \in W_{\Sigma_1}$ *and* $q \equiv p'$ *for some* p' *that is a* μ*-derivative of* p, *then* $\mu q + p \equiv p$.

PROOF. By induction on the structure of p. We may assume that p has the form $\sum_{1 \le i \le n} \mu_i p_i$.

Case (i). $\mu = \mu_i$ and $q \equiv p_i$. Then $p + \mu q \equiv p + \mu_i p_i \equiv p$ using (A1)–(A3).

Case (ii). $\mu = \mu_i$ and $q \equiv d$, $d \in \mathrm{Der}_\tau(p_i)$. By induction $p_i + \tau q \equiv p_i$. Therefore

$$\begin{aligned} p + \mu q &\equiv p + \mu_i p_i + \mu_i q \quad \text{using (A1)–(A3)}, \\ &\equiv p + \mu_i(p_i + \tau q) + \mu_i q, \\ &\equiv p + \mu_i(p_i + \tau q) \quad \text{from (A6.1)}, \\ &\equiv p. \end{aligned}$$

Case (iii). $\mu_i = \tau$ and $q \equiv d$, $d \in \mathrm{Der}_\mu(p_i)$. By induction, $p_i + \mu q \equiv p_i$. So

$$\begin{aligned} p + \mu q &\equiv p + \tau p_i + \mu q \quad \text{using (A1)–(A3)}, \\ &\equiv p + \tau(p_i + \mu q) + \mu q, \\ &\equiv p + \tau(p_i + \mu q) + p_i + \mu q + \mu q, && \text{from (A5)}, \\ &\equiv p + \tau(p_i + \mu q) + p_i + \mu q, && \text{from (A3)}, \\ &\equiv p + \tau(p_i + \mu q) && \text{from (A5)}, \\ &\equiv p. \end{aligned}$$

Definition. A sumform $p = \sum \mu_i p_i$ is a *proper normal form* if

(i) it is not of the form $\tau p'$;
(ii) each p_i is a proper normal form;
(iii) for $i \neq j$, p_i is not sumcongruent ($\equiv_s$) to any μ_i-derivative of $\mu_j p_j$.

An *improper normal form* is τp, where p is a proper normal form.
A *normal form* is either a proper or an improper normal form.

NORMAL FORM LEMMA. *Every sumform p is congruent to a normal form.*

PROOF. By induction on the structure of p. Let p be $\sum_{1 \leq i \leq n} \mu_i p_i$. By induction and (A6.2) we may assume that each p_i is in proper normal form. Suppose that, for some $k \neq j$, there exists $d \in \mathrm{Der}_{\mu_k}(\mu_j p_j)$ such that $p_k \equiv d$. From the Absorption Lemma $\mu_k p_k + \mu_j p_j \equiv \mu_j p_j$. So $p \equiv \sum_{i \neq k} \mu_i p_i$ and the result now follows by induction on the number of occurrences of duplicate derivatives. □

DERIVATIVE LEMMA. *The following are equivalent for normal forms p and q:*

(1) $p \equiv_s q$.
(2) *Each μ-derivative of p is a sumcongruent to a μ-derivative of q, and vice versa.*

PROOF

(1) implies (2): Immediate.
(2) implies (1): Let $p = \sum \lambda_i d_i + \sum \tau e_j$ and $p' = \sum \lambda'_i d'_i + \sum \tau e'_j$ be normal forms with sumcongruent derivatives, where $\lambda_i, \lambda'_i \in \Lambda$.

(A) We first show that each e_j is a sumcongruent to some e'_j and vice versa. Take e_1. Since $e_1 \in \mathrm{Der}_\tau(p)$, it is sumcongruent to some τ-derivative of p', say e'_1 or one of its τ-derivatives. In the former case, we are done; assume the latter.

But e'_1 is, by assumption, sumcongruent to e_j or one of its τ-derivatives for some $j, j \neq 1$ since e_1 is a proper subexpression of e'_1 up to sumcongruence. In either case e_1 is sumcongruent to a τ-derivative of τe_j, a contradiction since p is a normal form.

(B) Next, we show each d_i sumcongruent to some d'_i, $\lambda'_i = \lambda_i$, and vice versa. Take d_1. Since $d_1 \in \mathrm{Der}_{\lambda_1}(p)$, it is sumcongruent to some λ_1-derivative of p'. This cannot be a λ_1-derivative of some $\tau e'_j$—hence of some τe_j—since p is normal.

Hence, *either* $d_1 \equiv_s d_1'$ say, with $\lambda_1' = \lambda_1$, and we are done, *or* $d_1 \in \mathrm{Der}_\tau(d_1')$ up to congruence.

In the latter case d_1', a λ_1-derivative of p', must be sumcongruent to a λ_1-derivative of some summand of p, not $\lambda_1 d_1$ itself since d_1 is a proper subexpression of d_1' up to sumcongruence. Hence, d_1 is sumcongruent to a λ_1-derivative of the same summand, a contradiction since p is normal.

Combining (A) and (B), $p \equiv_s q$ follows. □

THEOREM 4.2, PART (ii) (COMPLETENESS). *For $p, q \in W_{\Sigma_2}$, $p \approx_c q$ implies $p \equiv q$.*

PROOF. Since every $p \in W_{\Sigma_2}$ is congruent under $\equiv$ to a sumform (by the axioms, especially (A7) to eliminate "|") and thence to a normal form (by the Normal Form Lemma), by the Soundness Theorem (Theorem 4.1, Part (i)) it is enough to consider normal forms p, q.

(A) p, q are proper normal forms. We prove by induction on the structure of p and q that, for λ_0 not in p or q,

$$p \not\equiv_s q \text{ implies } \forall^{ae}\, k.\ p \mid \lambda_0^k \not\approx q \mid \lambda_0^k,$$

where "$\forall^{ae}\, k$." means "for almost all k", and λ_0^k stands for λ_0 prefixed k times to NIL.

Case 1. q is sumcongruent to a τ-derivative of p. Since p is a proper normal form, $p = \mu d + \tau e + \cdots$ with $q \in \mathrm{Der}_\tau(\tau e)$ up to sumcongruence.

(i) $\mu = \tau$. Then for arbitrary k

$$p \mid \lambda_0^{k+1} \overset{\lambda_0}{\Longrightarrow} d \mid \lambda_0^k,$$

whereas $q \mid \lambda_0^{k+1} \overset{\lambda_0}{\Longrightarrow} r$ implies $r = q' \mid \lambda_0^k$, where $q' \in \mathrm{Der}_\tau(q)$ or $q' = q$, and so $q' \in \mathrm{Der}_\tau(\tau e)$ up to sumcongruence. Since p is normal, $d \not\equiv_s q'$, whence by induction

$$\forall^{ae}\, k.\ d \mid \lambda_0^k \not\approx q' \mid \lambda_0^k.$$

Since the number of possible q' is finite, we also have

$$\forall^{ae}\, k.\ p \mid \lambda_0^k \not\approx q \mid \lambda_0^k.$$

(ii) $\mu \neq \tau$. Then for arbitrary k

$$p \mid \lambda_0^k \overset{\mu}{\Longrightarrow} d \mid \lambda_0^k,$$

whereas $q \mid \lambda_0^k \overset{\mu}{\Longrightarrow} r$ implies $r = q' \mid \lambda_0^k$, where $q' \in \mathrm{Der}_\mu(q)$ and so $q' \in \mathrm{Der}_\mu(\tau e)$ up to sumcongruence. As before, $d \not\equiv_s q'$, and we previously proceed as in (i).

Case 2. Neither p nor q is sumcongruent to a τ-derivative of the other, and $p \not\equiv_s q$. Then by the Derivative Lemma, without loss of generality, for some μ and p', $p' \in \mathrm{Der}_\mu(p)$ but p' is sumcongruent to no μ-derivative of q.

(i) $\mu = \tau$. Then for arbitrary k

$$p \mid \lambda_0^{k+1} \overset{\lambda_0}{\Longrightarrow} p' \mid \lambda_0^k$$

whereas $q \mid \lambda_0^{k+1} \overset{\lambda_0}{\Longrightarrow} r$ implies $r = q' \mid \lambda_0^k$, where $q' = q$ or $q' \in \mathrm{Der}_\tau(q)$. In either case $p' \not\equiv_s q'$, and we proceed as in Case 1.

(ii) $\mu \neq \tau$. Then for arbitrary k

$$p \mid \lambda_0^k \overset{\mu}{\Longrightarrow} p' \mid \lambda_0^k$$

whereas $q \mid \lambda_0^k \overset{\mu}{\Longrightarrow} r$ implies $r = q' \mid \lambda_0^k$, where $q' \in \mathrm{Der}_\mu(q)$. Then, $p' \not\equiv_s q'$ and we proceed as in Case 1.

(B) p, q are arbitrary normal forms. If $p \approx_c q$, then also $p + \lambda_0 \approx_c q + \lambda_0$, where λ_0 does not occur in p or q, and both are proper normal forms. Hence by (A)

$$p + \lambda_0 \equiv_s q + \lambda_0,$$

from which $p \equiv_s q$ follows.

C3. PROOFS OF THEOREMS 3.2 AND 5.2

C3.1. *Outline of Proof of Theorem* 5.2. The proof just given for Theorem 4.2 can be adapted in a trivial manner to obtain Theorem 5.2. The axioms (A8)–(A10) hold for $\approx_c$ and also normal forms exist for the extended language since these axioms allow us to eliminate all instances of the operator [S].

C3.2. *Outline of Proof of Theorem* 3.2. Let $\equiv$ denote the congruence over W_{Σ_1} generated by (A1)–(A6). To prove soundness, that is, $p \equiv q \Rightarrow p \approx_c q$, it is more convenient to have a simpler representation of $\approx_c$.

Define $p \lessapprox q$ if for all $\mu \in \Lambda \cup \{\tau\}$

(i) $p' \in \mathrm{Der}_\mu(p)$ implies $p' \approx q'$, for some $q' \in \mathrm{Der}_\mu(q)$.
(ii) $q' \in \mathrm{Der}_\mu(q)$ implies $p' \approx q'$, for some $p' \in \mathrm{Der}_\mu(p)$.

It is easy to see that $\lessapprox$ is a congruence contained in $\approx$ and with a little work it can be shown to coincide with $\approx_c$. With this characterization, it is easy to prove that every instance of the axioms (A1)–(A6) satisfies $\approx_c$ and soundness follows.

To prove completeness, we use the same approach as in Section C2. This time we require a different notion of normal form, in which τ may only appear at top level:

(i) NIL is a *tight normal form.*
(ii) $\Sigma \mu_i N_i$ is a *normal form* if

(a) each N_i is a tight normal form,
(b) if $\mu_i = \mu_j$ then $N_i \not\equiv_s N_j$,
(c) if $\mu_i = \tau$ and $\mu_j = \lambda$ then $\mu_j N_j$ is not sumcongruent to any summand of N_i.

(iii) $\Sigma_i \mu_i N_i$ is a *tight normal form* if

(a) it is a normal form,
(b) $\mu_i \neq \tau$, $1 \leq i \leq n$.

Using (A5) and (A6), every term in W_{Σ_1} can be reduced to a normal form. By structural induction, we can then prove that $N \approx N'$ implies $N \equiv_s N'$ for tight normal forms. By extending this result to arbitrary normal forms, the result is established. □

REFERENCES

1. GORDON, M. J. *The Denotational Description of Programming Languages.* Springer-Verlag, New York, 1979.
2. HENNESSY, M., AND MILNER, R. On observing nondeterminism and concurrency. In *Proceedings of the 7th Colloquium on Automata, Languages and Programming.* Lecture Notes in Computer Science, vol. 85. Springer-Verlag, New York, Berlin, Heidelberg, 1980, pp. 299–309.
3. HENNESSY, M., AND PLOTKIN, G. D. Full abstraction for a simple parallel programming language. In *Proceedings of the 8th MFCS Conference* (Olomouc, Czechoslovakia). Lecture Notes in Computer Science, vol. 74. Springer-Verlag, New York, 1979, pp. 108–121.
4. MILNE, G., AND MILNER, R. Concurrent processes and their syntax. *J. ACM 26,* (1979), 302–321.
5. MILNER, R. Synthesis of communicating behaviour. In *Proceedings of the 7th MFCS Conference* (Zacopane, Poland). Lecture Notes in Computer Science, vol. 64. Springer-Verlag, New York, 1978, pp. 71–83.
6. MILNER, R. *A Calculus of Communicating Systems.* Lecture Notes in Computer Science, vol. 92. Springer-Verlag, New York, 1980.
7. PLOTKIN, G. D. A powerdomain construction. *SIAM J. Comput. 5,* 3 (1976), 452–487.
8. PNUELI, A. The temporal logic of programs. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science* (Providence, R.I.). IEEE, New York, 1977, pp. 46–57.
9. PRATT, V. R. Semantical considerations of Floyd-Hoare logic. In *Proceedings of the 17th Annual Symposium on Foundations of Computer Science.* IEEE, New York, 1976, pp. 109–121.
10. SMYTH, M. Powerdomains. *J. Comput. Syst. Sci. 15,* 1 (1978), 23–36.
11. STOY, J. E. *Denotational Semantics: The Scott Strachey Approach to Programming Language Theory.* MIT Press, Cambridge, Mass., 1977.

RECEIVED DECEMBER 1980; REVISED MAY 1984; ACCEPTED JULY 1984