Distributed Bisimulations

ILARIA CASTELLANI

INRIA Sophia-Antipolis, Valbonne, France

AND

MATTHEW HENNESSY

University of Sussex, Falmer, Brighton, United Kingdom

Abstract. A new equivalence between concurrent processes is proposed. It generalizes the well-known bisimulation equivalence to take into account the distributed nature of processes. The result is a noninterleaving semantic theory; concurrent processes are differentiated from processes that are nondeterministic but sequential. The new equivalence, together with its observational version, is investigated for a subset of the language CCS, and various algebraic characterizations are obtained.

Categories and Subject Descriptors: F.3.2 [**Theory of Computation**]: Logics and Meanings of Programs—*semantics of programming languages*

General Terms: Theory, Verification

Additional Key Words and Phrases: Algebraic, concurrent processes, noninterleaving, observational equivalence

1. *Introduction*

Recently there has been a proliferation of algebraic languages for describing concurrent processes. For a representative sample, see [2], [3], [5], [15], and [16]. Essentially each such language consists of a set of combinators—or constructors—for defining new processes in terms of simpler ones, together with some facility for recursive definitions.

A popular, and very successful, method of providing a semantic theory for these languages is via the notion of *bisimulation* [16, 17]. A bisimulation between two processes p and q is a relation R that provides a simulation of the behavior of p by q and simultaneously a simulation of the behavior of q by p. Two processes are said to be *bisimulation-equivalent* if there exists a bisimulation between them. This is now a classical way of formalizing the idea that two processes are indistinguishable when they show the same behavior, and a considerable amount of research effort has been expended in providing axiomatizations for bisimulation equivalence, in a variety of settings.

In this paper we wish to reconsider the notion of bisimulation and suggest a new one that preserves more of the structure of processes. Bisimulations are traditionally based on a description of processes as sequential transition systems, where transi-

Authors' addresses: I. Castellani, INRIA, Ecole des Mines, Sophia-Antipolis, 06565 Valbonne Cédex, France; M. Hennessy, Computer Science, Mathematics and Physics Building, University of Sussex, Falmer, Brighton, Sussex BN1 9QH, England.

tions are labeled by *atomic actions*. In this view a process evolves by successive elementary transitions and a concurrent process is semantically equivalent to a sequential nondeterministic one. Such theory of bisimulations ignores the concurrent structure of processes: it provides a so-called *interleaving semantics* for concurrency.

We would like here to propose a *noninterleaving* semantic theory that retains many of the advantages of the interleaving theories. Unlike [4], where a similar goal is achieved by labeling transitions with composite actions, we shall keep here to systems labeled by atomic actions. We restrict our attention to a rather simple language, the set of finite terms of CCS (Milner's Calculus of Communicating Systems). This provides a perfectly good framework in which to explain our ideas; it should also be adequate for the reader to evaluate them.

Let us briefly sketch the idea underlying our semantics. We interpret finite CCS terms as *distributed labeled transition systems*. In such a system, each transition gives rise to a compound residual $\langle p', p'' \rangle$, made out of a *local* component p' and a *global* component p''. Thus, a typical transition has the form $p \xrightarrow{a} \langle p', p'' \rangle$, where a is an atomic action and p'' includes p' as a component. Intuitively, separating the components allows us to distinguish *causality*, relating the action a to the local residual p', from *concurrency*, relating a to the "rest of" p''.

On the basis of these new transitions, we define on processes a behavioral equivalence called *distributed bisimulation equivalence*, which takes into account both residuals of transitions. We show that this equivalence distinguishes concurrent behaviors from nondeterministic ones, and thus is more discriminating than ordinary bisimulation.

The paper is self-contained even though some knowledge of [11] would be helpful. A good introduction to algebraic behavior languages may be found in [15]. We give now a short summary of the work presented.

In Section 2.1, we reexamine the usual view of processes as labeled transition systems, and propose our alternative description within which structural properties of processes may be reflected. In Section 2.2, we give the definition of distributed bisimulation and contrast it with the standard notion of bisimulation. The new equivalence is given a complete algebraic characterization in Section 2.3. This is achieved by introducing an asymmetric parallel operator $\lfloor\!\!\lceil$, similar to the left-merge operator of [3].

In Section 3 we apply the theory to a language that includes unobservable actions. We obtain a complete axiomatization of the corresponding distributed bisimulation equivalence by adding to the theory of Section 2 a set of so-called τ-laws. These include the τ-laws of [11]. In Section 4, we introduce communication into the language. The CCS approach to communication is followed: Communication is viewed as the simultaneous occurrence of complementary actions. We propose a set of axioms for this extended language, but their completeness remains an open problem. A brief conclusion follows.

2. *Distributed Bisimulations*

2.1 PROCESSES AS LABELED TRANSITION SYSTEMS. A very primitive operational semantics of processes can be given in terms of labeled transition systems.

Definition 2.1. A *labeled transition system* (*lts*) *is a triple* $(P, A, \rightarrow)$, where:

—P is a given set of processes,
—A is a given set of actions,
—$\rightarrow$ is a relation contained in $(P \times A \times P)$, called the *transition relation*.

RULES $\mathcal{R}$

Rule 1. $a{:}\ p \xrightarrow{a} p$

Rule 2. $p \xrightarrow{a} p'$ implies $p + q \xrightarrow{a} p'$
$q + p \xrightarrow{a} p'$

Rule 3. $p \xrightarrow{a} p'$ implies $p \mid q \xrightarrow{a} p' \mid q$
$q \mid p \xrightarrow{a} q \mid p'$

FIG. 1. CCS Transitions

We usually write $p \xrightarrow{a} q$ in place of $(p, a, q) \in \longrightarrow$. Intuitively, this means that p may perform the action a and thereby be transformed into q. In fact, many different interpretations may be placed on these triples and a discussion of these interpretations, for our particular language, will lead to our new semantics.

Let A be a given set of unspecified actions. The language we investigate is parameterized on A. Let Σ_1 be the signature consisting of

—NIL, a constant or nullary operator;
—$a.$ for each $a \in A$ a unary operator, called *prefixing*;
—$+$, a binary infix operator, called *choice*;
—$\mid$, a binary infix operator, called *parallel composition.*

We use L to denote the word algebra generated by Σ_1. When writing words in L, we use the usual conventions of CCS: prefixing has precedence over $\mid$, which in turn has precedence over $+$; prefixed terms such as $a.x$ are abbreviated to ax and NIL is often omitted. For example:

$$a.(c.\text{NIL}) + (b.\text{NIL} \mid (a.\text{NIL} + d.(a.\text{NIL})))$$

is rendered as $ac + b \mid (a + da)$. It represents a process that can either act like:

—the process ac, which performs the action a and then the action c; or
—the process $b \mid (a + da)$, which consists of two subprocesses in parallel, one of which can only perform b while the other can either perform a or d followed by a.

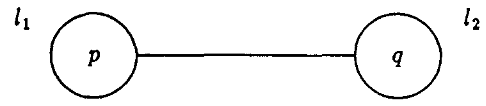
To give L the structure of a labeled transition system we let

—P, the set of processes, be the set of terms in L;
—A be the set of predefined actions or observations;
—$\longrightarrow$, the transition relation, be the least relation satisfying the axioms in Figure 1.

Here $p \xrightarrow{a} q$ is true only if it can be proved using the given three rules. For example, if p denotes $ac + b \mid (a + da)$, then the following are true:

$$p \xrightarrow{a} c,$$
$$p \xrightarrow{b} \text{NIL} \mid (a + da),$$
$$p \xrightarrow{a} b \mid \text{NIL},$$
$$p \xrightarrow{d} b \mid a.$$

We can interpret a transition or observation $p \xrightarrow{a} q$ as being the response of p to some external demand. That is, we can consider the evolution of p as being driven by some external experimenter or *observer*, which at each step asks p for some specific task a. If p is able to satisfy this demand, it performs an action a thereby evolving into a new system q. This is essentially the interpretation used in [11] to motivate the definition of observational equivalence. With respect to this interpretation, Rules 1 and 2 are eminently reasonable. On the other hand, the use of Rule 3 for processes of the form $p \mid q$ implies that the observer in question is ignoring information that should be apparent. At least this is the case if we make some reasonable assumptions about processes and observers. Suppose that $p \mid q$ describes a process consisting of two independent processes p and q, which are physically separated: It represents a distributed system with p in one locality l_1 and q in another locality l_2.



Assuming that the observer is an uncomplicated entity, when he makes an observation, he is *either* at location l_1 *or* at location l_2, but not at both. He then knows that his demand has been satisfied by some subprocess at his locality and that (at least for our simple language) satisfying this demand has not affected the components at other localities.

In proposing a new operational semantics, we do not wish to go as far as to assign names to localities and parameterize observers with respect to these localities. We shall simply assume that each observer is placed at some definite—but unspecified—locality: At each step he can ask for an action from the component at his locality, and observe the *local* result of this action. At the same time, he is informed of the *global* result of his observation. The idea is that the local result represents what causally follows the observed action. In this framework, a primitive observation takes the form:

$$p \xrightarrow{a} \langle p', p'' \rangle.$$

This is interpreted as

—an observer demands an action a of process p,
—satisfying this demand changes the component local to the point of observation into p' and changes the whole process into p''.

To sum up, each observation consists of some (local) action together with its local and global effects. We may add that it is crucial that these two effects should be observed together: Conducting a sequence of purely local observations and the corresponding sequence of global observations independently would not give us as much information. On the basis of our new transitions $p \xrightarrow{a} \langle p', p'' \rangle$, we propose now the following:

Definition 2.2. A *distributed lts* (*d-lts*) is a triple $(P, A, \rightarrow)$, where:

—P is a set of processes,
—A is a set of actions,
—$\rightarrow$ is a relation contained in $(P \times A \times P \times P)$, called the *transition relation*.

We shall write $p \xrightarrow{a} \langle p', p'' \rangle$ instead of $(p, a, p', p'') \in \rightarrow$. According to the interpretation given above, the language L can be viewed as a d-lts by using the transition relation defined in Figure 2. Rule 1′ merely says that an observer of the process ap sees the same local and global effect of the only possible successful demand, that of performing an action a. Rule 2′ is the usual interpretation of the choice operator $+$. Rule 3′, the most interesting one, states that if p can be observed

RULES $\mathscr{R}'$

Rule 1'.	$a: p \xrightarrow{a} \langle p, p\rangle$		
Rule 2'.	$p \xrightarrow{a} \langle p', p''\rangle$	implies	$p + q \xrightarrow{a} \langle p', p''\rangle$
			$q + p \xrightarrow{a} \langle p', p''\rangle$
Rule 3'.	$p \xrightarrow{a} \langle p', p''\rangle$	implies	$p \mid q \xrightarrow{a} \langle p', p'' \mid q\rangle$
			$q \mid p \xrightarrow{a} \langle p', q \mid p''\rangle$

FIG. 2. Distributed Transitions

performing an action a, and this observation changes p locally to p' and globally to p'', then $p \mid q$ can also be observed performing an action a, with a local effect still amounting to p', whereas the global effect is $p'' \mid q$, which includes the further unaffected component q.

For example, if p represents $ab + b(d + e) \mid (e + f)$, the following statements are true:

$$\begin{aligned}
&p \xrightarrow{a} \langle b, b\rangle,\\
&p \xrightarrow{b} \langle d + e, (d + e) \mid (e + f)\rangle,\\
&p \xrightarrow{e} \langle \mathrm{NIL}, b(d + e) \mid \mathrm{NIL}\rangle,\\
&p \xrightarrow{f} \langle \mathrm{NIL}, b(d + e) \mid \mathrm{NIL}\rangle.
\end{aligned}$$

Note that the transition relation given in Figure 1 can be recovered from the distributed one in Figure 2 by ignoring the first component. In fact, as noted already, the second component records the global evolution of the system after a single action, and this is just what an interleaving semantics describes. This also shows very clearly what additional information is used here to acquire more discriminating power over the concurrent aspects of systems.

2.2 BISIMULATIONS. We recall here the definition of bisimulation. Let $(P, A, \rightarrow)$ be an lts. A *bisimulation* is a symmetric relation $R \subseteq P \times P$ that satisfies, for every $(p, q) \in R$, $a \in A$, the following property:

$$p \xrightarrow{a} p' \quad \text{implies} \quad q \xrightarrow{a} q' \qquad \text{for some } q' \quad \text{such that} \quad (p', q') \in R. \qquad (*)$$

If we consider property $(*)$ as a function S on relations, we can rephrase the definition as: R is a bisimulation if it is symmetric and $R \subseteq S(R)$. Then *bisimulation equivalence* is defined by

$$p \sim q \qquad \text{if} \quad (p, q) \in R \quad \text{for some bisimulation } R.$$

This is a slight generalization, due to Park [17], of Milner's notion of *observational equivalence* [11, 15], and we refer the reader to these papers for proper motivation. In short, each transition $p \xrightarrow{a} p'$ can be viewed as an experiment on p, and two processes are equivalent if and only if there is no way of distinguishing them by experiments of this kind. The following results are well known.

LEMMA 2.3

(*a*) *$\sim$ is an equivalence relation*;
(*b*) *$\sim$ is the maximal symmetric fixed-point of the equation $R = S(R)$*;
(*c*) *$\sim$ is preserved by all operators in Σ_1, that is, $\sim$ is a Σ_1-congruence.*

Let us consider the bisimulation equivalence $\sim$ on our simple language L. It is easy to see that $\sim$ satisfies the expected laws:

$$p \mid q = q \mid p, \tag{P1}$$
$$p \mid (q \mid r) = (p \mid q) \mid r, \tag{P2}$$
$$p \mid \mathrm{NIL} = p. \tag{P3}$$

We also have the following identification, expressing the simulation of concurrency by nondeterministic interleaving:

$$a \mid b \sim ab + ba. \tag{L1}$$

We shall now apply exactly the same technique to define a behavioral equivalence in a distributed lts. In this setting, equivalence will mean the inability to formulate a collection of the more complicated experiments to distinguish between processes. More specifically, when testing two processes for equivalence, we shall require that *both* the local and the global results of observations be indistinguishable.

Definition 2.4. Let $(P, A, \rightarrow)$ be a d-lts. A *distributed bisimulation* (*d-bisimulation*) is a symmetric relation $R \subseteq (P \times P)$ satisfying, for every $(p, q) \in R$, $a \in A$, the following property:

$$p \xrightarrow{a} \langle p', p'' \rangle \quad \text{implies} \quad q \xrightarrow{a} \langle q', q'' \rangle \text{ for some } q', q'' \text{ such that } \langle p', q' \rangle \in R \text{ and } \langle p'', q'' \rangle \in R. \tag{**}$$

Again, if we let $D(R)$ denote (**), this amounts to demanding $R \subseteq D(R)$ (for a symmetric R). We then say that p and q are *d-bisimulation equivalent*, noted $p \sim_d q$, if $(p, q) \in R$ for some d-bisimulation R.

LEMMA 2.5

(*a*) *$\sim_d$ is an equivalence relation*;
(*b*) *$\sim_d$ is the maximal symmetric fixed-point of the equation $R = D(R)$*;
(*c*) *$\sim_d$ is a Σ_1-congruence.*

PROOF. The only nontrivial statement is (c). We shall only prove that $\sim_d$ is preserved by the operator $\mid$, since the proof is straightforward for the remaining operators.

Suppose that $p \sim_d q$ and r is an arbitrary process in L. We must construct a d-bisimulation R such that $(p \mid r, q \mid r) \in R$. The required relation is defined

$$R = \{(p_1 \mid r, p_2 \mid r) \mid p_1 \sim_d p_2\} \cup \sim_d. \qquad \square$$

The new behavioral equivalence $\sim_d$ also satisfies the properties P1, P2, and P3 listed above. However, the example of interleaving, L1, fails:

$$a \mid b \not\sim_d ab + ba,$$

because $a \mid b \xrightarrow{a} \langle \mathrm{NIL}, \mathrm{NIL} \mid b \rangle$, whereas the only possible observation of an action a from $(ab + ba)$ gives the result $\langle b, b \rangle$, and obviously $\mathrm{NIL} \not\sim_d b$.

As regards the relationship of our new equivalence with the standard bisimulation equivalence, it is easy to show that the standard equivalence is a *conservative extension* of the new one, in the following precise sense.

PROPOSITION 2.6

(*a*) *If $p \sim_d q$, then $p \sim q$.*
(*b*) *If p, q contain no occurrences of $\mid$, then $p \sim q$ implies $p \sim_d q$.*

PROOF

(a) Let R be a d-bisimulation over L. We show that it is also a bisimulation, that is, $R \subseteq S(R)$. This relies on the following facts:

(i) If $p \xrightarrow{a} r$ in the lts, then there exists some q such that $p \xrightarrow{a} \langle q, r \rangle$ in the d-lts.

(ii) If $p \xrightarrow{a} \langle q, r \rangle$ in the d-lts, then $p \xrightarrow{a} r$ in the lts.
Both these statements are proved by induction on the length of derivations in the respective transition systems.

(b) Let SL denote the subset of L that does not use the operator $|$. Now if $p \in$ SL and $p \xrightarrow{a} r$ in the lts or $p \xrightarrow{a} \langle q, r \rangle$ in the d-lts, it is easy to check that q and r are also in SL. Moreover, for terms of SL, the statements (i) and (ii) above can be strengthened to:

(iii) If $p \in$ SL, then $p \xrightarrow{a} r$ in the lts iff $p \xrightarrow{a} \langle r, r \rangle$ in the d-lts

This is sufficient to prove part (b). For suppose $p \sim q$, where $p, q \in R$. Then we may use property (iii) to show that $R \subseteq D(R)$, that is, R is a d-bisimulation. □

We conclude this section with a simple proposition, which expresses the relation between the local and global residuals of a transition.

Let $\equiv$ be the congruence on L generated by laws P1–P3. It is easy to show that

PROPOSITION 2.7. *If $p \xrightarrow{a} \langle q, r \rangle$, then $\exists s$ such that $q | s \equiv r$.*

This proposition states that the local residual is a parallel component of the global residual. This fact will be used in Section 3 to define distributed transitions in a more complicated setting.

2.3 ALGEBRAIC CHARACTERIZATION. In this section we present a complete axiomatization for each of the two Σ_1-congruences $\sim$ and $\sim_d$ over L. The former is well known and is mentioned only for completeness. The required axioms are given in Figure 3. The summation notation used in (IN), namely:

$$\sum_{i=1}^{n} a_i p_i \quad \text{to stand for} \quad \begin{cases} a_1 p_1 + \cdots + a_n p_n, & \text{if } n > 0, \\ \text{NIL}, & \text{if } n = 0, \end{cases}$$

is justified by the associativity of + as expressed by the law A1. In a summation term $\sum_{i=1}^{n} a_i p_i$ we usually drop the reference to n, rendering the term simply as $\sum a_i p_i$.

Let $\mathscr{A}$ denote the laws A1–A4, IN, as listed in Figure 3. The following result is well known (it is essentially Theorem 4.1 of [13]).

THEOREM 2.8. *The equivalence $\sim$ is the Σ_1-congruence over L generated by the axioms $\mathscr{A}$.*

This theorem is relatively easy to prove because of (IN). With it, all occurrences of $|$ can be eliminated, so that in this framework concurrency is reduced to nondeterminism. The primitive operators are NIL, prefixing, and choice.

Let us now turn our attention to the new equivalence $\sim_d$. We have already seen that (IN) is not valid for $\sim_d$. In the theory generated by $\sim_d$, we expect $|$ to be also a primitive operator.

Axioms $\mathscr{A}$

(A1) $x + (y + z) = (x + y) + z$
(A2) $x + y = y + x$
(A3) $x + \mathrm{NIL} = x$
(A4) $x + x = x$
(IN) If $x = \sum_{i=1}^{n} a_i x_i$, $y = \sum_{j=1}^{m} b_j y_j$, then

$$x \mid y = \sum_{i=1}^{n} a_i(x_i \mid y) + \sum_{j=1}^{m} b_j(x \mid y_j)$$

FIG. 3. Axiomatization of $\sim$.

This does not mean, however, that there will be no interesting dependencies between the operators. We have, for example, the following identifications:

(1) If $p = (r_1 + r_2) \mid s_1 + r_1 \mid (s_1 + s_2)$,
then $p + (r_1 \mid s_1) \sim_d p$.
(2) If $q = (r_1 + r_2) \mid s_1 + (r_1 + r_2) \mid s_2 + r_1 \mid (s_1 + s_2) + r_2 \mid (s_1 + s_2)$,
then $q + (r_1 + r_2) \mid (s_1 + s_2) \sim_d q$

Thus, $\sim_d$ allows more complicated absorptions than the one expressed by the idempotence law A4. Both (1) and (2) are in fact *absorption laws*: for instance, eq. (1) states that the term $(r_1 \mid s_1)$ may be absorbed into the term $r_1 \mid (s_1 + s_2) + (r_1 + r_2) \mid s_1$.

As a matter of fact, we can find arbitrarily complex absorption laws, which are all independent. Before giving more examples, let us formalize what we mean by *absorption* of a term into another.

Definition 2.9. Let $p \subsetapprox_d q \Leftrightarrow (p + q) \sim_d q$.

If $p \subsetapprox_d q$, we say that p is *absorbed* into q. It is clear that $p \sim_d q$ if and only if $p \subsetapprox_d q$ and $q \subsetapprox_d p$.

Thus, in the examples above, we have $(r_1 \mid s_1) \subsetapprox_d p$ and $(r_1 + r_2) \mid (s_1 + s_2) \subsetapprox_d q$. Another example (where we use $\mid$ up to associativity) is

(3) $(r_1 \mid s_1 \mid u_1) \subsetapprox_d (r_1 + r_2) \mid s_1 \mid u_1 + r_1 \mid (s_1 + s_2) \mid u_1 + r_1 \mid s_1 \mid (u_1 + u_2)$,

and the reader should be able to modify (2) above to a related but independent axiom in which $(r_1 + r_2 + r_3) \mid (s_1 + s_2 + s_3)$ is absorbed.

At this point the situation appears problematic. Let us then look back at the operational equivalence $\sim_d$ that we want to axiomatize. It is easy to convince oneself that when $p \xrightarrow{a} \langle q, r \rangle$, this is because p contains an initial subterm aq. Moreover, we know (by Proposition 2.7) that $r \equiv q \mid s$. Here, s represents, intuitively, the term which is concurrent to aq in p: We shall call it the *coterm* of aq in p.

Now, the operational behavior of a term p is exactly determined by the set of its (initial) subterms aq together with their coterms s. Unfortunately, such coterms are not, in general, subterms of p. For example, in

$$p = ((aq \mid s_1) + u) \mid s_2,$$

the term aq has coterm $s = s_1 \mid s_2$, which is not a subterm of p. Intuitively, this is because the action a eliminates all subterms in alternative with aq (the subterm u in the above example).

As a consequence, we are not able to prove the equality of two terms by first comparing their subterms aq and the respective coterms, as would be natural. To get round this difficulty we introduce a new operator $\lfloor\!\lfloor$ in the language, which will be used to express a term p in the *explicit form* $\sum_{i\in I} a_i p_i \lfloor\!\lfloor p_i'$, where for each $i \in I$ p_i' is the coterm of $a_i p_i$. The operational meaning of $\lfloor\!\lfloor$ is, as might be expected, specified by the following rules (respectively in the d-lts and in the lts):

$$p \xrightarrow{a} p'' \quad \text{implies } p \lfloor\!\lfloor q \xrightarrow{a} p'' \mid q \tag{R4}$$
$$p \xrightarrow{a} \langle p', p''\rangle \text{ implies } p \lfloor\!\lfloor q \xrightarrow{a} \langle p', p'' \mid q\rangle. \tag{R4$'$}$$

As these rules suggest, the operator $\lfloor\!\lfloor$ has some similarity with $\mid$. In fact, it is easy to see that $p \lfloor\!\lfloor q$ is absorbed into $p \mid q$, that is: $p \lfloor\!\lfloor q \subsetneq_d p \mid q$.

On the other hand $p \mid q$ is not absorbed in $p \lfloor\!\lfloor q$, whereas it is absorbed in $(p \lfloor\!\lfloor q + q \lfloor\!\lfloor p)$. Indeed, we have the law

$$p \mid q = p \lfloor\!\lfloor q + q \lfloor\!\lfloor p. \tag{LP1}$$

So $\lfloor\!\lfloor$ can be viewed as a sort of *asymmetric parallel operator*: In the term $p \lfloor\!\lfloor q$, the components p and q are concurrent but somehow p has an initial dominance over q. To be sure, the introduction of $\lfloor\!\lfloor$ may seem to bring us back to an interleaving semantics. Fortunately, this is not the case; at least in the d-lts. We have seen already that the behavioral equivalence $\sim_d$ does *not* satisfy the interleaving equation (IN) of Figure 3, and the addition of a new operator could not possibly affect this situation. However, the introduction of $\lfloor\!\lfloor$ brings the distinction between $\sim$ and $\sim_d$ into sharper focus. In the extended calculus, an essential property of $\sim$ is

$$ap \lfloor\!\lfloor q = a(p \mid q). \tag{$\lfloor\!\lfloor$ IN}$$

This law is *not* satisfied by $\sim_d$, as shown by the following counterexample:

$$a\text{NIL} \lfloor\!\lfloor b\text{NIL} \not\sim_d a(\text{NIL} \mid b\text{NIL}).$$

In fact, it may be shown that the interleaving equation (IN) is derivable from the more primitive equations (LP1) and ($\lfloor\!\lfloor$ IN). Indeed these laws have been used by Bergstra and Klop to give a finite axiomatization for the equivalence $\sim$ in the lts (see [3]). We shall return to this point at the end of this section.

Let us now come back to the properties of $\lfloor\!\lfloor$. An important difference between $\lfloor\!\lfloor$ and $\mid$ is that the former satisfies a distributive law

$$(p + q) \lfloor\!\lfloor r = (p \lfloor\!\lfloor r) + (q \lfloor\!\lfloor r), \tag{LP2}$$

whereas it is well known that this is not the case for $\mid$ (at least in the theory of bisimulations). The law (LP2) will help us a great deal in syntactic manipulations: In particular, it will be crucial to reduce a term p to its explicit form $\sum a_i p_i \lfloor\!\lfloor p_i'$. The final nontrivial axiom we require of $\lfloor\!\lfloor$ involves a subtle interaction with $\mid$

$$(p \lfloor\!\lfloor q) \lfloor\!\lfloor r = p \lfloor\!\lfloor (q \mid r). \tag{LP3}$$

The axioms LP1, LP2, and LP3 are very convenient. They can be used to derive the laws of $\mid$

$$p \mid q = q \mid p, \tag{P1}$$
$$p \mid (q \mid r) = (p \mid q) \mid r, \tag{P2}$$

as well as absorption laws such as (1) and (2) above. The other main property of $\mid$

$$p \mid \text{NIL} = p, \tag{P3}$$

follows from (LP1) above and the additional two axioms:

$$p \lceil \mathrm{NIL} = p, \qquad \text{(LP4)}$$
$$\mathrm{NIL} \lceil p = \mathrm{NIL}. \qquad \text{(LP5)}$$

Let $\mathscr{B}$ denote the set of axioms A1–A4, LP1–LP5, which are gathered together in Figure 4. We can now state the main result of this section. Let Σ_2 denote the signature extended with $\lceil$ and EL denote the extended-term language.

THEOREM 2.10 (CHARACTERIZATION). *The equivalence $\sim_d$ is the Σ_2-congruence over EL generated by the axioms $\mathscr{B}$, that is, $p =_{\mathscr{B}} q$ if and only if $p \sim_d q$.*

The remainder of this section is devoted to the proof of this result. One direction is straightforward, and is left to the reader:

LEMMA 2.11 (SOUNDNESS). *If $p, q \in EL$, $p =_{\mathscr{B}} q$ implies $p \sim_d q$.*

The proof of completeness relies, as usual, on a reduction of terms to *normal forms*. We shall adopt as normal forms the explicit forms mentioned earlier on.

Definition 2.12. $\sum_{i=1}^{n} a_i p_i \lceil p_i'$ is a normal form (nf) whenever all p_i, p_i' are nfs.

In particular NIL is an nf (for $n = 0$). We shall often write a normal form (different from NIL) simply as $\sum a_i p_i \lceil p_i'$.

We define now the *depth* $d(p)$ of a term p as follows:

$$\begin{aligned}
d(\mathrm{NIL}) &= 0;\\
d(a.p) &= 1 + d(p);\\
d(q \mid r) &= d(q) + d(r);\\
d(q \lceil r) &= \begin{cases} 0, & \text{if } d(q) = 0;\\ d(q) + d(r), & \text{otherwise;}\end{cases}\\
d(q + r) &= \max\{d(q), d(r)\}.
\end{aligned}$$

We may now prove the following lemma:

LEMMA 2.13 (NORMALIZATION LEMMA). *For any $p \in EL$, there exists a nf n such that $p =_{\mathscr{B}} n$.*

PROOF. By induction on the depth of p. We show at the same time that the depth of a term is preserved by normalization. The only nontrivial cases are when $p = q \mid r$ or $p = q \lceil r$. By induction, q and r have normal forms of equal depth. In both cases, if either of these is NIL, the arguments are trivial, using LP1, LP4, and LP5. Otherwise, we assume that q has the form $\Sigma b_j q_j \lceil q_j'$. We then have

(a) $$\begin{aligned} q \lceil r &= \Sigma(b_j q_j \lceil q_j') \lceil r, && \text{by LP2,}\\ &= \Sigma\, b_j q_j \lceil (q_j' \mid r), && \text{by LP3.}\end{aligned}$$

By induction each $q_j' \mid r$ has an nf (of equal depth), and the result follows, since

$$\begin{aligned} d(q \lceil r) &= d(q) + d(r)\\ &= \max_j\{d(b_j q_j \lceil q_j')\} + d(r)\\ &= \max_j\{d(b_j q_j) + d(q_j') + d(r)\} = d(\Sigma b_j q_j \lceil (q_j' \mid r)).\end{aligned}$$

(b) $q \mid r = q \lceil r + r \lceil q$, by LP1.

We know from part (a) that both $q \lceil r$ and $r \lceil q$ have nfs (of equal depth), and the result follows since the sum of two normal forms is also a normal form.

Axioms $\mathscr{B}$

(A1)	$x + (y + z) = (x + y) + z$
(A2)	$x + y = y + x$
(A3)	$x + \mathrm{NIL} = x$
(A4)	$x + x = x$
(LP1)	$x \mid y = x \lceil y + y \lceil x$
(LP2)	$(x + y) \lceil z = x \lceil z + y \lceil z$
(LP3)	$(x \lceil y) \lceil z = x \lceil (y \mid z)$
(LP4)	$x \lceil \mathrm{NIL} = x$
(LP5)	$\mathrm{NIL} \lceil x = \mathrm{NIL}$

FIG. 4. Axiomatization of $\sim_d$.

Moreover, since we have assumed that q, r have nfs different from NIL, we have indeed

$$d(q \mid r) = d(q) + d(r) = \max\{d(q \lceil r), d(r \lceil q)\} = d(q \lceil r + r \lceil q). \qquad \square$$

The other technical result we need to show completeness is

$$p \mid r \sim_d q \mid r \quad \text{implies} \quad p \sim_d q.$$

The proof is by induction on the size of terms, and to enable the induction to proceed smoothly it is necessary to simultaneously prove an auxiliary statement.

LEMMA 2.14 (SIMPLIFICATION). *For any $p, q, r, r', r'' \in EL$,*

(1) *$r \xrightarrow{a} \langle r', r'' \rangle$ and $p \mid r \sim_d q \mid r''$ imply $q \xrightarrow{a} \langle q', q'' \rangle$ for some q', q'' such that $r' \sim_d q'$ and $r' \sim_d q'$.*
(2) *$p \mid r \sim_d q \mid r$ implies $p \sim_d q$.*

PROOF. We prove the two statements simultaneously, by induction on the sizes of p, q.

(1) Let $r \xrightarrow{a} \langle r', r'' \rangle$. Then, $p \mid r \xrightarrow{a} \langle r', p \mid r'' \rangle$. There are two ways in which this may be matched by a move from $q \mid r''$.

   (i) $q \mid r'' \xrightarrow{a} \langle q', q'' \mid r'' \rangle$, because $q \xrightarrow{a} \langle q', q'' \rangle$.

   In this case $r' \sim_d q'$ and $p \mid r'' \sim_d q'' \mid r''$. Applying induction [case (2)], we obtain $p \sim_d q''$, and thus $q \xrightarrow{a} \langle q', q'' \rangle$ is the required move of $q \mid r''$.

   (ii) $q \mid r'' \xrightarrow{a} \langle s', q \mid s'' \rangle$, because $r \xrightarrow{a} \langle s', s'' \rangle$.

   Here $r' \sim_d s'$ and $p \mid r'' \sim_d q \mid s''$. Applying induction [case (1)], we obtain $q \xrightarrow{a} \langle q', q'' \rangle$, with $q' \sim_d s'$ and $p \sim_d q''$. From $r' \sim_d s'$ we deduce now the required $q' \sim_d r'$.

(2) Suppose $p \xrightarrow{a} \langle p', p'' \rangle$. We must find a matching move $q \xrightarrow{a} \langle q', q'' \rangle$ such that $p' \sim_d q'$ and $p'' \sim_d q''$.

   Consider the move $p \mid r \xrightarrow{a} \langle p', p'' \mid r \rangle$. There are two ways in which this may be matched by a move from $q \mid r$.

   (i) $q \mid r \xrightarrow{a} \langle q', q'' \mid r \rangle$, because $q \xrightarrow{a} \langle q', q'' \rangle$.

   In this case $p' \sim_d q'$ and $p'' \mid r \sim_d q'' \mid r$. Applying induction [case (2)], we obtain $p'' \sim_d q''$.

(ii) $q \mid r \xrightarrow{a} \langle r', q \mid r'' \rangle$, because $r \xrightarrow{a} \langle r', r'' \rangle$.

In this case, $p' \sim_d r'$ and $p'' \mid r \sim_d q \mid r''$. Applying induction [case (1)], we obtain $q \xrightarrow{a} \langle q', q'' \rangle$, with $r' \sim_d q'$ and $p'' \sim_d q''$. From $p' \sim_d r'$ it follows that $p' \sim_d q'$. □

Combining these results, we finally obtain

PROPOSITION 2.15 (COMPLETENESS). *If $p, q \in EL$, $p \sim_d q$ implies $p =_{\mathscr{B}} q$.*

PROOF. By induction on the sum of sizes of p, q. Suppose that $p \sim_d q$. In the light of the normalization lemma, (and given the soundness of $\mathscr{B}$ for $\sim_d$) we may assume p, q to be normal forms:

$$p = \sum a_i p_i \,\Gamma\, p_i', \qquad q = \sum b_j q_j \,\Gamma\, q_j'.$$

We show that $q + p =_{\mathscr{B}} q$. Then, by a symmetric argument, we have also $p + q =_{\mathscr{B}} p$, and by combining these two equalities we obtain the required result: $p =_{\mathscr{B}} p + q =_{\mathscr{B}} q$. To prove $q + p =_{\mathscr{B}} q$ it is enough to show, $\forall i \in I$:

$$q + a_i p_i \,\Gamma\, p_i' =_{\mathscr{B}} q.$$

Since $p \xrightarrow{a_i} \langle p_i, p_i \mid p_i' \rangle$ and $p \sim_d q$, there must exist $j \in J$ such that $q \xrightarrow{b_j} \langle q_j, q_j \mid q_j' \rangle$, with $a_i = b_j$ and $p_i \sim_d q_j$, $p_i \mid p_i' \sim_d q_j \mid q_j'$. Since $\sim_d$ is a congruence, we can replace p_i for q_j in the latter equality, to obtain $p_i \mid p_i' \sim_d p_i \mid q_j'$. Then, by the simplification lemma, we have $p_i' \sim_d q_j'$. Now, from $p_i \sim_d q_j$, $p_i' \sim_d q_j'$, we can infer, using induction, that $p_i =_{\mathscr{B}} q_j$, $p_i' =_{\mathscr{B}} q_j'$. Whence by substitution and A4, we obtain

$$q + a_i p_i \,\Gamma\, p_i' =_{\mathscr{B}} q + b_j q_j \,\Gamma\, q_j' =_{\mathscr{B}} q. \qquad \square$$

We mentioned already that in the theory for $\sim$ on the extended language the interleaving axiom (IN) can be replaced by the law (Γ IN). By adding (Γ IN) to the axioms of Figure 4, we obtain a finite axiomatization for $\sim$ over EL. Indeed, in the theory for $\sim$, our operator Γ has the same meaning as the left-merge operator of Bergstra and Klop [3].

3. *Unobservable Actions*

In this section we develop the theory of distributed bisimulations in the presence of unobservable actions. In a fully-fledged language, such actions come about via internal communications between the parallel components of a process (for more details we refer the reader to [15]). However, we can start to study the effect of unobservable actions on the theory quite independently of how they occur. Moreover, since the theory of ordinary bisimulations is well understood in this framework, at least for simple languages such as L [11], we shall concentrate on distributed bisimulations only.

3.1 WEAK OBSERVATIONS. Let us assume, following Milner [15], that the set A contains a distinguished symbol τ representing an internal unobservable action. That is, A is now of the form $O \cup \{\tau\}$. We shall use a, b, c to represent observable actions from O and μ, ν to range over the entire set A. As in Milner [15], we wish now to replace our observation relations $\xrightarrow{a}$ by the weaker observation relations $\overset{a}{\Longrightarrow}$, in order to abstract to some extent from internal actions.

Informally, the meaning of a weak observation $p \overset{a}{\Longrightarrow} \langle q, r \rangle$ is that p can evolve internally for some time, then perform an action a and thereafter still possibly move internally to reach the state $\langle q, r \rangle$. Thus, a weak observation $\overset{a}{\Longrightarrow}$ involves a transition $\xrightarrow{a}$ as well as a (finite) number, possibly equal to zero, of transitions $\xrightarrow{\tau}$ before and after it. For these unobservable transitions we choose here the simple form $p \xrightarrow{\tau} q$ rather than $p \xrightarrow{\tau} \langle q, r\rangle$. This means that an action τ is regarded as *global*, and can be localized only indirectly, if it affects the observable behavior of the component where it occurs. For example, the locality of the action τ will not be observable in the process $\tau p \mid q$, whereas it will be in the process $(\tau p + q) \mid r$, since here the τ can prevent the local observer of $(\tau p + q)$ from obtaining an action of q. In fact, because of the presence of internal actions, an observation $\overset{a}{\Longrightarrow}$ may have an effect on different components. For example, one would expect

$$ap \mid (q + \tau r) \overset{a}{\Longrightarrow} \langle p, p \mid r\rangle,$$

since the action τ of the component $(q + \tau r)$ may occur while the a-observation is taking place on the component ap.

Such *nonlocal* effects of observations will become more pronounced when we introduce a form of communication into our language, in the next section.

Because of the presence of these nonlocal effects, the formal definition of $\overset{a}{\Longrightarrow}$ is complicated, at least if we retain the present notation. Recall that in $p \overset{a}{\Longrightarrow} \langle q, r\rangle$, the term q represents the local residual of the observation, whereas r represents the global residual. This global residual r must include the local one q, and it is difficult to retain the consistency in $\langle q, r\rangle$ between q and the copy of q in the global process r. Intuitively, r is obtained by placing q in a global context $C[\,]$. So an observation is in fact of the form

$$p \overset{a}{\Longrightarrow} \langle q, C[q]\rangle.$$

In $C[q]$, the occurrence of $[\,]$ represents the locality of the observation and q the local residual. It follows that $C[q]$ contains all the information of the pair $\langle q, C[q]\rangle$. We may therefore formalize an observation simply by an arrow of the form

$$p \overset{a}{\Longrightarrow} C[q],$$

relating a term p to an instantiated context $C[q]$.

In order to avoid confusion, let us formally define what we mean by instantiated context. Let the set of *generalized terms* be defined by the following grammar:

$$t ::= \mathrm{NIL} \mid t + t \mid t \,|\, t \mid t \curlyvee t \mid [\,] \mid [t].$$

Note that all terms in EL are generalized terms. We continue to use p, q, etc. to range over EL, and will refer to elements of EL as terms.

The new syntactic categories we introduce are subsets of the set of generalized terms. A *context* is a generalized term with no occurrence of a subterm of the form $[t]$ in it, and at most one occurrence of $[\,]$. Contexts will usually be denoted by $C[\,]$, $C'[\,]$, $D[\,]$, etc., and $C[t]$ will be used to denote an *instantiated context*, that is, the generalized term obtained by substituting t for the unique occurrence of $[\,]$ in $C[\,]$, if it exists. A *process* is an instantiated context of the form $C[p]$, where p is in EL. In $C[p]$ the occurrence of $[p]$ indicates the locality under scrutiny. We shall use P, Q, etc., to range over processes. Note that all terms in EL are also processes.

In fact there are many processes that will never occur in our operational framework, but technically there is no need to isolate them. Also, it will sometimes be convenient to consider processes as terms in EL simply by ignoring the local information, that is, viewing $[p]$ as p. Formally, the term in EL corresponding to the process $C[p]$ is denoted by $C(p)$; it is obtained by substituting p, rather than $[p]$, for $[\,]$ in $C[\,]$.

The relations $\overset{a}{\Longrightarrow}$ are defined in terms of the two simpler relations $\xrightarrow{\tau}$ and $\xrightarrow{a}$. The relation $\xrightarrow{\tau}$ is defined to be the least relation over processes that satisfies the rules of Figure 5. The type of the relations $\xrightarrow{a}$ is more restricted. They take elements of EL and return processes; they are defined to be the least relations—between EL and the set of processes—that satisfy the rules in Figure 6. Combining the two kinds of arrows, $\xrightarrow{\tau}$ and $\xrightarrow{a}$, we get the relations $\overset{a}{\Longrightarrow}$, from EL to processes, that form the basis of our modified notion of equivalence:

(i) $p \overset{\tau}{\Longrightarrow} q$ if $p \xrightarrow{\tau}{}^{+} q$;
(ii) $p \overset{\varepsilon}{\Longrightarrow} q$ if $p \xrightarrow{\tau}{}^{*} q$;
(iii) $p \overset{a}{\Longrightarrow} C[p']$ if $p \xrightarrow{\tau}{}^{*} q \xrightarrow{a} D[q'] \xrightarrow{\tau}{}^{*} C[p']$.

Here $\xrightarrow{\tau}{}^{+}$ denotes the transitive closure of $\xrightarrow{\tau}$, and $\xrightarrow{\tau}{}^{*}$ its transitive and reflexive closure. Thus, $p \overset{\varepsilon}{\Longrightarrow} q$ means that p may evolve to q by an indeterminate number, possibly zero, of internal moves, while $p \overset{\tau}{\Longrightarrow} q$ means that at least one internal move is performed. Hence, $p \overset{a}{\Longrightarrow} C[p']$ allows internal silent moves to be made during an a-observation.

Examples

$$ap \mid (q + \tau r) \overset{a}{\Longrightarrow} [p] \mid r. \tag{1}$$

Here the a-observation is performed on the component ap and the local residual is p. The global residual is $[p] \mid r$ because the (nonlocal) internal move $(q + \tau r) \xrightarrow{\tau} r$ is performed during the observation.

$$(v + \tau w) \mid a(\tau p + q) \mid x + \tau y \overset{a}{\Longrightarrow} w \mid [p] \mid y. \tag{2}$$

Here the local observation is "$a(\tau p + q)$ performs the action a to become p", whereas the nonlocal effect is to transform the global environment $(v + \tau w) \mid [\,] \mid (x + \tau y)$ into $w \mid [\,] \mid y$.

Because of the simplicity of our language the actual form of the global environment is very simple. Let $\equiv$ denote equality of terms modulo associativity and commutativity of $\mid$, as defined just before Proposition 2.7. It is then easy to prove that

LEMMA 3.1. *If $p \overset{a}{\Longrightarrow} C[p']$, then $\exists s \in EL$ such that $s \mid [\,] \equiv C[\,]$.*

We can also establish a relation with the more primitive observations of the previous section. We leave to the reader the proof of the following proposition:

PROPOSITION 3.2. *If p contains no occurrences of τ, then $p \overset{a}{\Longrightarrow} C[q]$ if and only if $\exists r \equiv C[q]$ such that $p \xrightarrow{a} \langle q, r \rangle$.*

3.2 WEAK DISTRIBUTED BISIMULATIONS. In this section, we revise the definition of distributed bisimulation, using the weak arrows $\overset{a}{\Longrightarrow}$ in place of $\xrightarrow{a}$. The result will be a behavioral equivalence that abstracts from internal actions. In addition to the distributed observation $\overset{a}{\Longrightarrow}$ it is convenient to consider also observations that record the spontaneous evolution of a process after a finite but indefinite amount of time. This amounts to using $\overset{\varepsilon}{\Longrightarrow}$ as well as $\overset{a}{\Longrightarrow}$ in the definition of the weak equivalence. It can be argued that this is a natural form of observation, and its presence is technically useful. Observations $\overset{\varepsilon}{\Longrightarrow}$ are also used in the standard theory of bisimulations [15, 16]. We give next our definition of weak distributed bisimulation.

$$\begin{array}{llll}
\textit{Rule } \tau 1. & \tau\colon P \xrightarrow{\tau} P & & \\
\textit{Rule } \tau 2. & P \xrightarrow{\tau} P' & \text{implies} & P + Q \xrightarrow{\tau} P' \\
 & & & Q + P \xrightarrow{\tau} P' \\
\textit{Rule } \tau 3. & P \xrightarrow{\tau} P' & \text{implies} & P \mid Q \xrightarrow{\tau} P' \mid Q \\
 & & & Q \mid P \xrightarrow{\tau} Q \mid P' \\
\textit{Rule } \tau 4. & P \xrightarrow{\tau} P' & \text{implies} & P \lfloor Q \xrightarrow{\tau} P' \mid Q \\
\textit{Rule } \tau 5. & p \xrightarrow{\tau} p' & \text{implies} & [p] \xrightarrow{\tau} [p']
\end{array}$$

FIG. 5. τ-Rules (for General Processes)

$$\begin{array}{llll}
\textit{Rule } \mathrm{E}1. & a\colon p \xrightarrow{a} [p] & & \\
\textit{Rule } \mathrm{E}2. & p \xrightarrow{a} C[p'] & \text{implies} & p + q \xrightarrow{a} C[p'] \\
 & & & q + p \xrightarrow{a} C[p'] \\
\textit{Rule } \mathrm{E}3. & p \xrightarrow{a} C[p'] & \text{implies} & p \mid q \xrightarrow{a} C[p'] \mid q \\
 & & & q \mid p \xrightarrow{a} q \mid C[p'] \\
 & & & p \lfloor q \xrightarrow{a} C[p'] \mid q
\end{array}$$

FIG. 6. a-Rules (from Terms to Processes)

Definition 3.3. A symmetric relation $R \subseteq (EL \times EL)$ is a *weak d-bisimulation* if it satisfies $R \subseteq WD(R)$, where $WD(R)$ is defined by

$(p, q) \in WD(R)$ if $\forall a \in A$:

(i) $p \overset{\varepsilon}{\Longrightarrow} p'$ implies $q \overset{\varepsilon}{\Longrightarrow} q'$ for some q' such that $(p', q') \in R$,
(ii) $p \overset{a}{\Longrightarrow} C[p']$ implies $q \overset{a}{\Longrightarrow} D[q']$ for some q'
such that $(p', q') \in R$ and $(C[p'], D[q']) \in R$.

We write $p \approx_d q$ if $(p, q) \in R$ for some weak bisimulation. As before, it is straightforward to show that $\approx_d$ is an equivalence relation and is the maximal fixpoint of the mapping WD. Moreover, it is a natural extension of the previous notion of d-bisimulation. The reader can easily check the following proposition:

PROPOSITION 3.4

(i) $p \sim_d q$ implies $p \approx_d q$.
(ii) If p, q contain no occurrences of τ, then $p \approx_d q$ implies $p \sim_d q$.

This new equivalence is also preserved by most of our combinators. However, as expected from the standard theory of bisimulations, it is not preserved by $+$; the usual counterexample works.

Example 3. $\tau a \approx_d a$ but $b + \tau a \not\approx_d b + a$

We react to this inconvenience in the standard way. We take as our reference behavioral equivalence the largest Σ_2-congruence generated by $\approx_d$

$$p \approx_d^c q \qquad \text{if for every context} \quad C[\,]\colon C[p] \approx_d C[q].$$

In the next section, we show that this relation can be characterized equationally by adding new equations to those which characterize the simpler equivalence $\sim_d$. These new equations are concerned with internal actions.

However, to derive this characterization we need to reformulate $\approx_d^c$ in a more usable form. This is quite straightforward as the only combinator that misbehaves is $+$. In fact it is easy to prove that

PROPOSITION 3.5. *$p \approx_d^c q$ iff $a + p \approx_d a + q$ for some a not in p, q.*

As an immediate consequence we have the useful property (recall that $\overset{\tau}{\Rightarrow} = \overset{\tau}{\rightarrow}^{+}$):

PROPERTY 3.6. *$p \approx_d^c q$ and $p \overset{\tau}{\Rightarrow} p'$ implies $q \overset{\tau}{\Rightarrow} q'$ for some q' such that $p' \approx_d q'$.*

The characterization of Proposition 3.5 will be extensively used in the next section. We use it now to establish a close relationship between $\approx_d^c$ and $\approx_d$.

COROLLARY 3.7. *$p \approx_d q$ if and only if one of the following holds*:

(i) $p \approx_d^c q$
(ii) $p \approx_d^c \tau q$
(iii) $\tau p \approx_d^c q$

PROOF. The nontrivial direction is to show that $p \approx_d q$ implies one of the three alternatives (i), (ii), (iii). We proceed by case analysis.

(1) Suppose that $p \overset{\tau}{\Rightarrow} p'$ and the corresponding move of q is $q \overset{\varepsilon}{\Rightarrow} q$, with $p' \approx_d q$. In this case we have $p \approx_d^c \tau q$. In fact it is easy to check that, if $a \notin p, q$, then $p + a \approx_d \tau q + a$: corresponding to the move $p + a \overset{\varepsilon}{\Rightarrow} p'$ of the first term we pick the move $\tau q + a \overset{\varepsilon}{\Rightarrow} q$ of the second term.
(2) Symmetrically, if $q \overset{\tau}{\Rightarrow} q'$ and p replies with $p \overset{\varepsilon}{\Rightarrow} p$, we have $\tau p \approx_d^c q$.
(3) If neither (i) nor (ii) holds, it follows that $p + a \approx_d q + a$, that is $p \approx_d^c q$. □

The next section is devoted to the search of an axiomatization for $\approx_d^c$.

3.3 ALGEBRAIC CHARACTERIZATION. In this section we present a complete set of axioms for the weak behavioral equivalence $\approx_d^c$. First, we find that the three τ-laws of [11] and [15] are valid for $\approx_d^c$:

$$x + \tau x = \tau x, \tag{I1}$$
$$\mu \tau x = \mu x, \tag{I2}$$
$$\mu(x + \tau y) + \mu y = \mu(x + \tau y). \tag{I3}$$

These three properties can be easily proven for $\approx_d^c$ using the characterization in Proposition 3.5. Furthermore, we have a law

$$\tau(x \mid y) = \tau x \mid y, \tag{I4}$$

showing an interesting interaction between τ and $\mid$.

Here again, however, to obtain a complete axiomatization for our behavioral equivalence we need to recourse to the asymmetric operator $\lfloor$. So, for example, the eq. I4 will be derivable from the following two laws of $\lfloor$:

$$\tau x \lfloor y = \tau(x \mid y), \tag{NI1}$$
$$x \lfloor y = x \lfloor \tau y. \tag{NI2}$$

The law NI1 expresses the globality of τ-actions, while NI2 may be viewed as a generalization of I2. One further law is required for $\lfloor$, which is similar in structure to I3:

$$x \lfloor (y + \tau z) + x \lfloor z = x \lfloor (y + \tau z). \tag{NI3}$$

Let now $\mathscr{S}$ denote the set of axioms $\mathscr{B}$ extended with the τ-laws (I1)–(I3), (NI1)–(NI3). All these laws are grouped together in Figure 7. We have the following characterization for $\approx_d^c$.

THEOREM 3.8. *The equivalence $\approx_d^c$ is the Σ_2-congruence generated by the axioms $\mathscr{S}$.*

Axioms $\mathscr{S} = \mathscr{B} \cup \tau$-laws

$$\left.\begin{array}{ll}
\text{(A1)} & x + (y + z) = (x + y) + z \\
\text{(A2)} & x + y = y + x \\
\text{(A3)} & x + \text{NIL} = x \\
\text{(A4)} & x + x = x \\
\text{(LP1)} & x \mid y = x \lfloor\!\lfloor y + y \lfloor\!\lfloor x \\
\text{(LP2)} & (x + y) \lfloor\!\lfloor z = x \lfloor\!\lfloor z + y \lfloor\!\lfloor z \\
\text{(LP3)} & (x \lfloor\!\lfloor y) \lfloor\!\lfloor z = x \lfloor\!\lfloor (y \mid z) \\
\text{(LP4)} & x \lfloor\!\lfloor \text{NIL} = x \\
\text{(LP5)} & \text{NIL} \lfloor\!\lfloor x = \text{NIL}
\end{array}\right]\ \text{Axioms } \mathscr{B}$$

$$\left.\begin{array}{ll}
\text{(I1)} & x + \tau x = \tau x \\
\text{(I2)} & \mu\tau x = \mu x \\
\text{(I3)} & \mu(x + \tau y) + \mu y = \mu(x + \tau y) \\
\text{(NI1)} & \tau x \lfloor\!\lfloor y = \tau(x \mid y) \\
\text{(NI2)} & x \lfloor\!\lfloor \tau y = x \lfloor\!\lfloor y \\
\text{(NI3)} & x \lfloor\!\lfloor (y + \tau z) + x \lfloor\!\lfloor z = x \lfloor\!\lfloor (y + \tau z)
\end{array}\right]\ \tau\text{-laws}$$

FIG. 7. Axiomatization of $\approx_d^c$, in absence of communication

The structure of the proof of this first theorem is similar to that of the corresponding theorem in Section 2.3, but the details are somewhat more complicated. We start by showing the following:

PROPOSITION 3.9 (SOUNDNESS). *$p =_{\mathscr{S}} q$ implies $p \approx_d^c q$*

PROOF. Because of Proposition 3.5, it is sufficient to check that $a + p \approx_d a + q$ for every instantiation $p = q$ of the equations in $\mathscr{S}$; as usual, a should not appear in p, q. Note that it would be difficult to check directly that $\approx_d^c$ satisfies the equations, as we would have to consider all possible contexts. □

The converse proposition, namely the completeness of the axioms, relies on a more complicated kind of normal form, which we define next.

Definition 3.10. $\sum a_i p_i \lfloor\!\lfloor p_i' + \sum \tau p_j$ is a *weak normal form* (*wnf*) whenever all p_i, p_i', p_j are wnfs.

Again, when both sums are empty we obtain the wnf NIL. We have now the following normalization lemma:

LEMMA 3.11 (WEAK NORMALIZATION). *For every $p \in EL$, there exists a wnf n such that $p =_{\mathscr{S}} n$.*

PROOF. By induction on the depth of p. As before, the only difficulty is with terms of the form $q \mid r$ and $q \lfloor\!\lfloor r$. We consider here only the case of $q \lfloor\!\lfloor r$, as the case $q \mid r$ will then follow by LP1.

Suppose $p = q \lfloor\!\lfloor r$. If $m = \text{wnf}(q) = \text{NIL}$, we define $n = \text{NIL}$. Then, using induction and axiom LP4, we get: $p =_{\mathscr{S}} m \lfloor\!\lfloor r = \text{NIL} \lfloor\!\lfloor r =_{\mathscr{S}} \text{NIL} =_{\mathscr{S}} n$.

Otherwise, if $m = \sum a_i q_i \lfloor\!\lfloor q_i' + \sum \tau q_j \neq \text{NIL}$, we have

$$\begin{aligned}
q \lfloor\!\lfloor r &=_{\mathscr{S}} \sum (a_i q_i \lfloor\!\lfloor q_i') \lfloor\!\lfloor r + \sum (\tau q_j \lfloor\!\lfloor r) && \text{by LP2,} \\
&=_{\mathscr{S}} \sum a_i q_i \lfloor\!\lfloor (q_i' \mid r) + \sum \tau(q_j \lfloor\!\lfloor r) && \text{by LP3, NI1.}
\end{aligned}$$

We may now apply induction on each $(q_i' \mid r)$, $(q_j \lfloor\!\lfloor r)$ to obtain a wnf.

We also need a simplification lemma, similar to that used in Section 2.3.

LEMMA 3.12 (SIMPLIFICATION LEMMA). *If* $p, q, r \in \mathscr{P}'$:

(*i*) $r \overset{\tau}{\Longrightarrow} r'$ *and* $p \mid r \approx_d q \mid r'$ *imply* $q \Longrightarrow q'$, *for some* q' *such that* $p \approx_d q'$.
(*ii*) $r \overset{a}{\Longrightarrow} [r'] \mid r''$ *and* $p \mid r \approx_d q \mid r''$ *imply* $q \overset{a}{\Longrightarrow} [q'] \mid q''$, *for some* q', q'' *such that* $r' \approx_d q'$ *and* $p \approx_d q''$.
(*iii*) *Simplification*: $p \mid r \approx_d q \mid r$ *implies* $p \approx_d q$.

PROOF. The three statements are proved simultaneously, by induction on the sum of sizes of p, q, r.

(i) Here $p \mid r \overset{\tau}{\Longrightarrow} p \mid r'$ and therefore $q \mid r'$ must have a matching move $q \mid r' \overset{\varepsilon}{\Longrightarrow} q' \mid r'' \approx_d p \mid r'$, for some q', r'' such that $q \overset{\varepsilon}{\Longrightarrow} q'$ and $r' \overset{\varepsilon}{\Longrightarrow} r''$.
If $r'' = r'$ we may apply induction on part (iii) to get $p \approx_d q'$. Thus, $q \overset{\varepsilon}{\Longrightarrow} q'$ is the required matching move. Otherwise, we apply induction on part (i) to obtain $q' \overset{\varepsilon}{\Longrightarrow} q'' \approx_d p$. Here the matching move is $q \overset{\varepsilon}{\Longrightarrow} q''$.

(ii) Here $p \mid r \overset{a}{\Longrightarrow} p \mid [r'] \mid r''$. Now $q \mid r''$ can match this move in two ways:

(a) $q \mid r'' \overset{a}{\Longrightarrow} [q'] \mid q'' \mid r'''$ because $q \overset{a}{\Longrightarrow} [q'] \mid q''$ and $r'' \overset{\varepsilon}{\Longrightarrow} r'''$, with $r' \approx_d q'$ and $r' \mid p \mid r'' \approx_d q' \mid q'' \mid r'''$. Since $\approx_d$ is preserved by $\mid$, we also have $r' \mid p \mid r'' \approx_d r' \mid q'' \mid r'''$. Whence, by induction on part (iii), we deduce $p \mid r'' \approx_d q'' \mid r'''$.
Now, if $r''' = r''$, we can apply the same induction to obtain $p \approx_d q''$. In this case, $q \overset{a}{\Longrightarrow} [q'] \mid q''$ is the required move of q.
Otherwise, $r'' \overset{\tau}{\Longrightarrow} r'''$, and we can apply induction, part(i), to obtain $q'' \overset{\varepsilon}{\Longrightarrow} q''' \approx_d p$. In this case, the required move of q is $q \overset{a}{\Longrightarrow} [q'] \mid q'''$.

(b) $q \mid r'' \overset{a}{\Longrightarrow} q' \mid [s'] \mid s''$ because $r'' \overset{a}{\Longrightarrow} [s'] \mid s''$ and $q \overset{\varepsilon}{\Longrightarrow} q'$. Then, $r' \approx_d s'$ and $r' \mid p \mid r'' \approx_d s' \mid q' \mid s''$. Again, we have $r' \mid p \mid r'' \approx_d r' \mid q' \mid s''$ and we may apply induction, part (iii), to obtain $p \mid r'' \approx_d q' \mid s''$. We can now finally apply induction on part (ii) to get $q' \overset{a}{\Longrightarrow} [q''] \mid q'''$, and therefore $q \overset{a}{\Longrightarrow} [q''] \mid q'''$, with $r'' \approx_d q''$ and $p \approx_d q'''$.

(iii) To prove $p \approx_d q$, we show that for any move of p there exists a corresponding move of q. The converse will then follow by symmetry.

(a) $p \overset{\tau}{\Longrightarrow} p'$. Then, $p \mid r \overset{\tau}{\Longrightarrow} p' \mid r$ and $q \mid r$ must have a matching move $q \mid r \overset{\varepsilon}{\Longrightarrow} q' \mid r' \approx_d p' \mid r$, where $q \overset{\varepsilon}{\Longrightarrow} q'$ and $r \overset{\varepsilon}{\Longrightarrow} r'$.
If $r' = r$, we directly apply induction on part (iii) to get $p' \approx_d q'$.
Otherwise, $r \overset{\tau}{\Longrightarrow} r'$, and we may apply induction on part (i) to obtain $q' \overset{\varepsilon}{\Longrightarrow} q'' \approx_d p'$.

(b) $p \overset{a}{\Longrightarrow} [p'] \mid p''$. Then $p \mid r \overset{a}{\Longrightarrow} [p'] \mid p'' \mid r$. Again, $q \mid r$ has two ways to match this move:

(I) $q \mid r \overset{a}{\Longrightarrow} [q'] \mid q'' \mid r'$ because $q \overset{a}{\Longrightarrow} [q'] \mid q''$ and $r \overset{\varepsilon}{\Longrightarrow} r'$. We then have $p' \approx_d q'$ and $p' \mid p'' \mid r \approx_d q' \mid q'' \mid r'$.
If $r' = r$, we may apply induction, part (iii), to get $p' \mid p'' \approx_d q' \mid q''$. In this case, $q \overset{a}{\Longrightarrow} [q'] \mid q''$ is the required move of q.
Otherwise, $r \overset{\tau}{\Longrightarrow} r'$ and we may apply induction on part (i) to obtain $q' \mid q'' \overset{\varepsilon}{\Longrightarrow} s' \mid s'' \approx_d p' \mid p''$, for some s', s'' such that $q' \overset{\varepsilon}{\Longrightarrow} s'$ and $q'' \overset{\varepsilon}{\Longrightarrow} s''$. Once more we have two cases. If $s' = q'$, the required move of q is $q \overset{a}{\Longrightarrow} [q'] \mid s''$. If $q' \overset{\tau}{\Longrightarrow} s'$, we may once more apply induction, part (i), to the equation $s' \mid s'' \approx_d p' \mid p''$, and obtain $s'' \overset{\varepsilon}{\Longrightarrow} s''' \approx_d p''$. In this case the matching move is $q \overset{a}{\Longrightarrow} [q'] \mid s'''$.

(II) $q \mid r \overset{a}{\Longrightarrow} q' \mid [r'] \mid r''$ because $r \overset{a}{\Longrightarrow} [r'] \mid r''$ and $q \overset{\varepsilon}{\Longrightarrow} q'$. Then $p' \approx_d r'$ and $p' \mid p'' \mid r \approx_d r' \mid q' \mid r''$. Whence we can deduce

$r' \mid p'' \mid r \approx_d r' \mid q' \mid r''$. From the latter equation and induction, part (iii), we get $p'' \mid r \approx_d q' \mid r''$. Applying induction on part (ii), we finally obtain $q' \overset{a}{\Longrightarrow} [q''] \mid q'''$, with $r' \approx_d q''$ and $p'' \approx_d q'''$. Since $p' \approx_d r'$, this is the required move of q. □

The proof of our completeness result is more complicated than for $\sim_d$. Besides a normalization lemma, it also requires two *absorption lemmas*, which show how residuals may be absorbed into terms.

LEMMA 3.13 (τ-ABSORPTION LEMMA). *If p is a wnf and $p \overset{\tau}{\Longrightarrow} p'$, then $p + \tau p' =_{\mathscr{Y}} p$.*

PROOF. The case $p \xrightarrow{\tau} q$ is trivial. The general case is proved by induction on the number of τ-actions in p. The inductive step uses axiom I1. □

The second absorption lemma, which is somewhat more involved, concerns the observations $\overset{a}{\Longrightarrow}$.

LEMMA 3.14 (GENERALIZED ABSORPTION LEMMA). *If p is a wnf and $p \overset{a}{\Longrightarrow} C[p'] \equiv [p'] \mid r$, then $p + ap' \curlyvee r =_{\mathscr{Y}} p$.*

PROOF. By induction on the length of the derivation $p \overset{a}{\Longrightarrow} C[p']$. We work modulo the relation $\equiv$, so $C[p']$ will be rendered as $[p'] \mid r$. Let $p = \sum a_i p_i \curlyvee p_i' + \sum \tau p_j$. We examine three cases:

(i) $p \xrightarrow{a} [p'] \mid r$. Thus, there exists $i \in I$ such that $a = a_i$, $p' = p_i$, and $r = p_i'$. We then have our result by simple absorption A4.

(ii) $p \overset{a}{\Longrightarrow} [p'] \mid r$ because for some $j \in J$: $\tau p_j \xrightarrow{\tau} p_j \overset{a}{\Longrightarrow} [p'] \mid r$. By induction $p_j + ap' \curlyvee r =_{\mathscr{Y}} p_j$. Whence we deduce, using I1:

$$\begin{aligned} p &=_{\mathscr{Y}} p + \tau p_j, \\ &=_{\mathscr{Y}} p + \tau p_j + p_j, \\ &=_{\mathscr{Y}} p + \tau p_j + p_j + ap' \curlyvee r, \\ &=_{\mathscr{Y}} p + ap' \curlyvee r. \end{aligned}$$

(iii) $p \overset{a}{\Longrightarrow} [p'] \mid r$ because $p \overset{a}{\Longrightarrow} [q] \mid s \xrightarrow{\tau} [p'] \mid r$. By induction, we have $p =_{\mathscr{Y}} p + aq \curlyvee s$. There are now two subcases, according to how the internal move $[q] \mid s \xrightarrow{\tau} [p'] \mid r$ is made:

(a) $q \xrightarrow{\tau} p'$ and $s = r$. Then $q =_{\mathscr{Y}} q + \tau p'$ by the τ-absorption lemma. We then have, using axiom I3:

$$aq =_{\mathscr{Y}} a(q + \tau p') =_{\mathscr{Y}} a(q + \tau p') + ap' =_{\mathscr{Y}} aq + ap'.$$

Whence we deduce, using axiom LP1:

$$\begin{aligned} p &=_{\mathscr{Y}} p + aq \curlyvee r, \\ &=_{\mathscr{Y}} p + (aq + ap') \curlyvee r, \\ &=_{\mathscr{Y}} p + aq \curlyvee r + ap' \curlyvee r, \\ &=_{\mathscr{Y}} p + ap' \curlyvee r. \end{aligned}$$

(b) $s \xrightarrow{\tau} r$ and $q = p'$. Again we may use the τ-absorption lemma to get $s =_{\mathscr{Y}} s + \tau r$. We may now deduce, using axiom NI3:

$$\begin{aligned} p &=_{\mathscr{Y}} p + ap' \curlyvee (s + \tau r), \\ &=_{\mathscr{Y}} p + ap' \curlyvee (s + \tau r) + ap' \curlyvee r, \\ &=_{\mathscr{Y}} p + ap' \curlyvee r. \end{aligned}$$ □

We are now ready to prove the completeness of the equations.

PROPOSITION 3.15 (COMPLETENESS). *If $p, q \in EL$, $p \approx_d q$ implies $p =_{\mathscr{E}} q$.*

PROOF. The proof is by induction on the sum of sizes of p, q. Suppose then that $p \approx_d^c q$. We may assume p, q to be normal forms:

$$p = \sum a_i p_i \;\Upsilon\; p_i' + \sum \tau p_j, \qquad q = \sum b_n q_n \;\Upsilon\; q_n' + \sum \tau q_m.$$

We proceed with our usual method. We show that $q + p =_{\mathscr{E}} q$ and the result will follow by symmetry. We prove separately

(i) $q + a_i p_i \;\Upsilon\; p_i' =_{\mathscr{E}} q \qquad \forall i \in I$,
(ii) $q + \tau p_j =_{\mathscr{E}} q \qquad \forall j \in J$.

PROOF OF (i). We have here $p \overset{a_i}{\Longrightarrow} [p_i] \mid p_i'$. Since $p \approx_d^c q$, there must exist q', r such that $q \overset{a_i}{\Longrightarrow} [q'] \mid r$, with $p_i \approx_d q'$ and $p_i \mid p_i' \approx_d q' \mid r$. Since $\approx_d$ is preserved by $\mid$, we have also $q' \mid p_i' \approx_d q' \mid r$, and applying the simplification lemma, we get $p_i' \approx_d r$. Now Corollary 3.7 gives three possible cases for $p_i \approx_d q'$ and $p_i' \approx_d r$, in each of which we can apply induction to obtain $p_i =_{\mathscr{E}} q'$ or $\tau p_i =_{\mathscr{E}} q'$ or $p_i =_{\mathscr{E}} \tau q'$, and $p_i' =_{\mathscr{E}} r$ or $\tau p_i' =_{\mathscr{E}} r$ or $p_i' =_{\mathscr{E}} \tau r$. We then have

$$\begin{aligned} a_i p_i \;\Upsilon\; p_i' &=_{\mathscr{E}} a_i q' \;\Upsilon\; p_i' && \text{possibly using I2,} \\ &=_{\mathscr{E}} a_i q' \;\Upsilon\; r && \text{possibly using NI2,} \end{aligned}$$

whence, by the generalized absorption lemma, we finally deduce

$$q =_{\mathscr{E}} q + a_i q' \;\Upsilon\; r =_{\mathscr{E}} q + a_i p_i \;\Upsilon\; p_i'.$$

PROOF OF (ii). Here $p \approx_d^c q$ and $p \xrightarrow{\tau} p_j$ imply $q \overset{\tau}{\Longrightarrow} r$ for some r such that $p_j \approx_d r$. Using Corollary 3.7 and induction as before, we obtain $p_j =_{\mathscr{E}} r$ or $\tau p_j =_{\mathscr{E}} r$ or $p_j =_{\mathscr{E}} \tau r$. Then, prefixing both terms by τ and possibly using I2, we have $\tau p_j =_{\mathscr{E}} \tau r$. We may now use the τ-absorption lemma to get

$$q =_{\mathscr{E}} q + \tau r =_{\mathscr{E}} q + \tau p_j. \qquad \square$$

4. *Communication*

In this section we reinterpret the parallel operator $\mid$ in order to allow communication between its components. This is a very straightforward extension, but we still lack an equational characterization for the corresponding distributed bisimulation.

4.1 COMMUNICATION AS MUTUAL OBSERVATION. We adopt Milner's model of communication [15], which is one of the standard methods for introducing communication into process algebras. We shall recall just the essential features here, referring the reader to [15] for more details. We assume our set of observable actions O to be of the form $\Lambda \cup \overline{\Lambda}$, where Λ is a given set of observation or action names and Λ is the set of their formal complements, $\overline{\Lambda} = \{\bar{a} \mid a \in \Lambda\}$. For convenience, we also say that a is the complement of $\bar{a}$, that is, $\bar{\bar{a}} = a$, for any $a \in O$.

In this setting, communication is defined to be the simultaneous occurrence of two complementary actions. So a communication occurs in the process $p \mid q$ if p performs some action a and q simultaneously performs its complement $\bar{a}$. In other words, communication is mutual observation between two parallel processes. Any communication is treated as an *internal* action and is therefore denoted by τ.

Rule F1.	μ: $P \overset{\mu}{\longmapsto} P$		
Rule F2.	$P \overset{\mu}{\longmapsto} P'$	implies	$P + Q \overset{\mu}{\longmapsto} P'$
			$Q + P \overset{\mu}{\longmapsto} P'$
Rule F3.	$P \overset{\mu}{\longmapsto} P'$	implies	$P \mid Q \overset{\mu}{\longmapsto} P' \mid Q$
			$Q \mid P \overset{\mu}{\longmapsto} Q \mid P'$
			$P \lceil Q \overset{\mu}{\longmapsto} P' \mid Q$
Rule F4.	$p \overset{\mu}{\longmapsto} p'$	implies	$[p] \overset{\mu}{\longmapsto} [p']$
Rule F5.	$P \overset{a}{\longmapsto} P'$, $Q \overset{\bar{a}}{\longmapsto} Q'$	imply	$P \mid Q \overset{\tau}{\longmapsto} P' \mid Q'$

FIG. 8. Global transitions (for general processes)

Let us explain how we introduce this kind of communication in our language. We want to define weak transitions $\overset{a}{\Longrightarrow}$ where communications are absorbed both before and after a transition $\xrightarrow{a}$; since the result of a transition $\xrightarrow{a}$ is a process of the form $C[p]$, we need a communication rule for processes as well as for terms in EL. In order to obtain this, we first extend the transitions $\xrightarrow{a}$ to general processes.

Moreover, since τ-actions are taken to be global, a communication transition will be inferred from "global" transitions $\xrightarrow{a}$ (i.e., transitions that do not introduce a locality in the residual). We shall denote these global transitions for processes by $\overset{a}{\longmapsto}$, and the resulting τ-transitions by $\overset{\tau}{\longmapsto}$. The relations $\overset{\mu}{\longmapsto}$, relating processes to processes, are defined in Figure 8. For terms in EL, the transitions $\overset{a}{\longmapsto}$ correspond precisely to the distributed observations $\xrightarrow{a}$ if we ignore the locality in the residual. For processes of the form $C[p]$, the global transitions $\overset{\mu}{\longmapsto}$ preserve the existing locality, without ever introducing a new one.

Based on the arrows $\overset{a}{\longmapsto}$, we may now finally define our communication rule for processes (and terms).

Rule F5. $P \overset{a}{\longmapsto} P'$, $Q \overset{\bar{a}}{\longmapsto} Q'$ imply $P \mid Q \overset{\tau}{\longmapsto} P' \mid Q'$.

The global transitions $\overset{a}{\longmapsto}$ are only used to derive communications. The resulting transitions $\overset{\tau}{\longmapsto}$ are then used, together with the distributed transitions $\xrightarrow{a}$, to define the weak relations $\overset{a}{\Longrightarrow}$ (and $\overset{\varepsilon}{\Longrightarrow}$):

(i) $p \overset{\tau}{\Longrightarrow} q$	if	$p \overset{\tau}{\longmapsto}{}^{+} q$,
(ii) $p \overset{\varepsilon}{\Longrightarrow} q$	if	$p \overset{\tau}{\longmapsto}{}^{*} q$,
(iii) $p \overset{a}{\Longrightarrow} C[p']$	if	$p \overset{\tau}{\longmapsto}{}^{*} q \xrightarrow{a} D[q'] \overset{\tau}{\longmapsto}{}^{*} C[p']$.

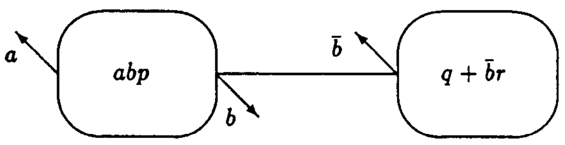
Example 4

$abp \mid (q + \bar{b}r) \overset{a}{\Longrightarrow} [p] \mid r$ $(*)$

because $abp \mid (q + \bar{b}r) \xrightarrow{a} [bp] \mid (q + \bar{b}r)$ $(**)$

and $[bp] \overset{b}{\longmapsto} [p]$, $(q + \bar{b}r) \overset{\bar{b}}{\longmapsto} r$, whence $[bp] \mid (q + \bar{b}r) \overset{\tau}{\longmapsto} [p] \mid r$.

The latter transition, combined with $(**)$, gives $(*)$. Graphically, we have the following situation:



Here, the locality of the a-observation is linked to another component of the system, and the act of observing the action a via a weak observation $\overset{a}{\Longrightarrow}$ effects a modification of this remote component. One may easily design examples in which a local observation effects a long chain of modifications in a collection of increasingly remote components.

We can now use our new observations to define a weak distributed bisimulation, exactly as we did in the previous section. We still use $\approx_d$ to denote the resulting equivalence. A typical identification under $\approx_d$ will be

Example 5. Let $p = (aq + r) \mid (\bar{a}q' + r')$. Then $p \approx_d p + \tau(q \mid q')$.

This example merely emphasizes the fact that τ is synonymous with communication.

As in the previous section, the equivalence $\approx_d$ is not preserved by the operator $+$ and, as usual, we take as our behavioral relation its closure with respect to contexts $\approx_d^c$. In the next section, we discuss a possible equational characterization of $\approx_d^c$ in this new setting.

4.2 Towards an Algebraic Characterization. As noted already, the relation $\approx_d^c$ is difficult to manipulate; we proceed here to define a more amenable alternative, as we did in the previous section. We state without proof

Proposition 4.1. *$p \approx_d^c q$ iff $a + p \approx_d a + q$ for some a not in p, q.*

One virtue of this reformulation of $\approx_d^c$ is that we can check fairly easily if it satisfies equations. One interesting new equation that it does satisfy is

$$(ax + y) \mid (\bar{a}x' + y') = (ax + y) \mid (\bar{a}x' + y') + \tau(x \mid x'). \qquad \text{(C1)}$$

The phenomenon underlying this equation has already been discussed in Example 5. A similar absorption equation is

$$abx \mid \bar{b}y = abx \mid \bar{b}y + ax \upharpoonright y. \qquad \text{(C2)}$$

Although C2 involves the operator $\upharpoonright$, there are instances of this phenomenon in the original CCS language:

$$abx \mid \bar{b} = abx \mid \bar{b} + ax.$$

Unfortunately, these equations cannot be used as the basis of a complete axiomatization. As usual, the key for the axiomatization lies with $\upharpoonright$.

Recall that in Section 4.1 we did not introduce an observation rule corresponding to Rule F5 for $\upharpoonright$: According to our definition, processes may not communicate across $\upharpoonright$. For this reason, the law LP1

$$x \mid y = x \upharpoonright y + y \upharpoonright x$$

is no longer valid. One way to solve this problem would be to include $\upharpoonright$ in Rule F5. However, this would render some of our axioms unsound.

Instead we outline a simple approach; we introduce a new operator $|_c$ (analogue to that used in [3]), which enforces communication between its components. The operational semantics of $|_c$ is specified by the unique rule:

Rule $\tau 7$. $P \overset{a}{\mapsto} P'$, $Q \overset{\bar{a}}{\mapsto} Q'$ imply $P|_c Q \overset{\tau}{\mapsto} P' \mid Q'$.

So $P|_c\, Q$ can only perform communications, that is, τ-moves. In this extended language, LP1 is replaced by

$$x \mid y = x \upharpoonright y + y \upharpoonright x + x|_c y. \qquad \text{(LP1')}$$

(LP1′) $x \mid y = x \upharpoonright y + y \upharpoonright x + x \mid_c y$
(CP1) $x \mid_c (y + z) = (x \mid_c y) + (x \mid_c z)$
(CP2) $x \mid_c y = y \mid_c x$
(CP3) $x \mid_c \mathrm{NIL} = \mathrm{NIL}$
(CP4) $(\mu x \upharpoonright x') \mid_c (\nu y \upharpoonright y') = \begin{cases} \tau(x \mid y) \upharpoonright (x' \mid y'), & \text{if } \mu = \bar{\nu} \\ \mathrm{NIL}, & \text{otherwise} \end{cases}$
(CP5) $a(x \mid x') \upharpoonright (y \mid y') \sqsubseteq a(cx \upharpoonright x' + v) \upharpoonright (\bar{c}y \upharpoonright y' + w)$

FIG. 9. Communication axioms for $\approx_d^c$.

Axiomatizing the properties of $\mid_c$ is straightforward because of its simplicity. The operator distributes over +

$$x \mid_c (y + z) = x \mid_c y + x \mid_c z, \tag{CP1}$$

and it only allows communications between its arguments

$$\mu x \mid_c \nu y = \begin{cases} \tau(x \mid y), & \text{if } \mu = \bar{\nu}, \\ \mathrm{NIL}, & \text{otherwise.} \end{cases}$$

A more general version of this identity, involving $\upharpoonright$, is the law CP4 given in Figure 9; CP1–CP3 are the other properties of interest of $\mid_c$. The law CP5, where $\sqsubseteq$ is defined by $p \sqsubseteq q$ if $(p + q) = q$, is a general version of the absorption equation C2.

As can be seen from these equations, there is a close analogy between our combinator $\mid_c$ and the "communication merge" of [3]. Note also that the law C1 above is now a derived equation.

We would like to show that the new set of equations is complete, but the proof eludes us (precisely, we have not been able to generalize the simplification lemma); this remains an open problem. However, a complete axiomatization for a slightly different formulation of our semantics may be found in [6].

5. *Conclusion*

We have provided a new "noninterleaving" semantics for simple CCS-like languages. This semantics is based on a minor extension of the well-known idea of bisimulation; the extension takes into account some information on the distributed nature of processes. The result is a semantic theory that takes concurrency into account and that can be completely axiomatized, at least for simple languages. The major omission in this respect is a treatment of hiding or restriction of channels. In an interleaving semantics, such as the standard bisimulation theory, this presents no problem although the introduction of hiding into the language increases its expressiveness considerably. It enables one to abstract from internal details of a process. However, considerable thought is required before we can extend our ideas of local and global observations to distributed systems in which the behavior of individual components can be influenced by remote components that are linked by invisible channels. The basic problem is to decide how to continue the isolated observations of such local components and how to formalize this decision as a modification of our notion of distributed bisimulation. At least, we hope that the present paper convinces the reader that this line of research is worth pursuing. When the theory is extended to a more expressive language, such as the whole "pure CCS", the new operator $\upharpoonright$ will be seen to be more reasonable. One would expect then $ax \upharpoonright y$ to be equivalent to the CCS expression $(\alpha a x \mid \bar{\alpha} y)\backslash\alpha$.

There have been other attempts at providing noninterleaving semantics for CCS. For example, in [9] CCS terms are interpreted as Petri nets. However, Petri nets are a rather concrete operational model of computation, whose algebraic nature is not very well understood. Moreover, their semantics makes distinctions that are difficult to justify intuitively. For example, the terms aNIL and (aNIL + aNIL) are treated differently. Similar remarks apply to translations of CCS into event structures. Such translations are useful for comparing disparate models of computation, but these structures are too concrete to provide an abstract behavioral view of processes. It may be, however, that behavioral equivalences can be defined directly on event structures, which could then be inherited directly by CCS. See [4], [6], and [21] for work in this direction.

Another proposal for a noninterleaving semantics of CCS may be found in [7] and other papers by the same authors. Although the starting idea of their semantics is very similar to ours (keep to atomic transitions, but identify the component that moves at each step), they do not use it to directly define an equivalence relation, as we do, but rather to build partially ordered computations for processes.

More recently, labeled partial orders have been used as semantic domains for algebraic concurrent languages, see [4], [18], and [19] and in this setting there have been a number of generalizations of bisimulation equivalence. The essential idea is to generalize the simple experiment $p \xrightarrow{a} q$ to $p \xrightarrow{o} q$, where o is a labeled partial order giving considerable detail of the computation from p to q. This form of experiment is used in [4], [7], and [21] to define new variants of observational equivalence. The only complete axiomatization may be found in [4], and it is known that their equivalence is coarser than distributed bisimulation. For example, the terms

$$p = a(b + c) + (a \mid b) + (a \mid c),$$
$$q = p + a \mid (b + c),$$

are identified by pomset bisimulation and distinguished by distributed bisimulation. A proof that distributed bisimulation equivalence implies pomset bisimulation equivalence, for the language without communication considered in [4], may be found in [6].

A different generalization of observational equivalence is given in [10]. Here actions are assumed to be nonatomic, having a distinct beginning and end. The resulting equivalence on finite CCS terms with communication and τ-actions is axiomatized. This equivalence is different from distributed bisimulation because it does not satisfy the law

$$\mu(x + \tau y) + \mu y = \mu(x + \tau y). \tag{13}$$

Finally, we should mention some attempts at generalizing equivalences other than bisimulation to take concurrency into consideration. In [1], the testing equivalence of [8] is extended to event structures, and in [20], an extension of failure semantics for CSP is presented.

REFERENCES

1. ACETO, L., DE NICOLA, R., AND FANTECHI, A. Testing equivalences for event structures. Nota interna B4-63. Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Richerche, Pisa, Italy, 1986.
2. AUSTRY, D., AND BOUDOL, G. Algèbre de processus et synchronisation. *J. Theoret. Comput. Sci. 30* (1984), 91–131.
3. BERGSTRA, J., AND KLOP, J. Algebra of communicating processes with abstraction. *J. Theoret. Comput. Sci. 37*, 1 (1985), 77–121.

4. Boudol, G., and Castellani, I. Concurrency and atomicity. *Theoret. Comput. Sci. 59* (1988), 25–84.
5. Brookes, S., Hoare, C., and Roscoe, A. A theory of communicating sequential processes. *J. ACM 31*, 3 (July 1984), 560–599.
6. Castellani, I. Bisimulations for concurrency. PhD dissertation, Univ. Edinburgh, Edinburgh, Great Britain, 1988.
7. Degano, P., De Nicola, R., and Montanari, U. Partial ordering derivations for CCS. In *Proceedings of Fundamentals of Computer Science 85*. Lecture Notes in Computer Science, vol. 199, Springer-Verlag, New York, 1985, pp. 520–533.
8. De Nicola, R., and Hennessy, M. Testing equivalences for processes. *J. Theoret. Comput. Sci. 34* (1984), 83–133.
9. Goltz, U., and Mycroft, A. On the relationship of CCS and Petri nets. In *Proceedings of the International Conference on Automata, Languages and Programming* (*ICALP 84*). Lecture Notes in Computer Science, vol. 172. Springer-Verlag, New York, 1984, pp. 196–208.
10. Hennessy, M. Axiomatising finite concurrent processes. Tech. Rep. 4, Computer Science, Univ. Sussex, Sussex, Great Britain, 1987.
11. Hennessy, M., and Milner, R. Algebraic laws for nondeterminism and concurrency. *J. ACM 32*, 1 (Jan. 1985), 137–161.
12. Hennessy, M., and Plotkin, G. A term model for CCS. In Lecture Notes in Computer Science, vol. 88. Springer-Verlag, New York, 1980.
13. Hoare, C. *Communicating Sequential Processes*. Prentice-Hall, New York, 1985.
14. Milner, R. Flowgraphs and flow algebras. *J. ACM 26*, 4 (Oct. 1979), 794–818.
15. Milner, R. A calculus of communicating systems. In Lecture Notes in Computer Science, vol. 92. Springer-Verlag, New York, 1980.
16. Milner, R. Calculi for synchrony and asynchrony. *J. Theoret. Comput. Sci. 25* (1983).
17. Park, D. Concurrency and automata on infinite sequences. In Lecture Notes in Computer Science, vol. 104. Springer-Verlag, New York, 1981.
18. Pratt, V. R. Modelling concurrency with partial orders. *Int. J. Prog. 15* (1986).
19. Shields, M. Concurrent machines. *The Comput. J. 28* (1985).
20. Taubner, D., and Vogler, W. The step failure semantics. In Lecture Notes in Computer Science, vol. 247. Springer-Verlag, New York, 1987.
21. van Glabbeek, R., and Vaandrager, F. Petri net models for algebraic theories of concurrency. In *Proceedings of the PARLE Conference* (Eindhoven). Lecture Notes in Computer Science, vol. 254. Springer-Verlag, New York, 1987, pp. 224–242.
22. Winskel, G. Event structure semantics for CCS and related languages. In *Proceedings of ICALP 82*. Lecture Notes in Computer Science, vol. 140. Springer-Verlag, New York, 1982.

RECEIVED DECEMBER 1987; REVISED DECEMBER 1988; ACCEPTED MARCH 1989