INTRODUCTION

In this paper, I will describe and critically evaluate the paper *Distributed Bisimulations*, by Ilaria Castellani and Matthew Hennessy [5]. The aim of their paper is to generalise bisimulation equivalence for a subset of CCS to take into account the distributed nature of processes and, thus give a noninterleaving semantic theory which has many of the advantages of the interleaving theory. This theory allows concurrent processes to be distinguished from nondeterministic sequential processes and preserves more of the structure of processes. The article deals predominantly with the axiomatisation of the new equivalences.

Milner's CCS (Calculus of Communicating Systems) [11], assumes that the observer and the processes occur in one locality. The authors assume that processes occur in independent localities and therefore the observer as 'an uncomplicated entity' can be at either location but not at both. The observer asks for a specific action and will see the results of this local action. Also the observer will be informed of the global results of the action. A primitive observation takes the form $p \stackrel{a}{\rightarrow} \langle p', p'' \rangle$. This can be interpreted as an observer requesting an action of a of process p and performing a causes the local component to evolve to p', and the whole process changes to p''.

The structure of Castellani and Hennessy's paper is as follows - they present the notion of a distributed labelled transition system, and from this define the concepts of strong distributed bisimulation and weak distributed bisimulation. There is no communication between processes. The two distributed bisimulations are axiomatised in a similar fashion to the approach taken in Chapter 7.4 in Milner [11]. In the final section a calculus with communication is introduced and suggestions for axiomatisation are made.

My aim in this paper is to explore the concepts presented in Distributed Bisimulations by way of examples and explanations with reference to Milner's work [11]. I will not discuss all the details of the paper and I will only give an outline of the axiomatisation, as I feel that the proofs are fully dealt with in the paper. I will describe the concepts and evaluate the paper as I proceed. As there are some notational differences that are significant, I will first introduce basic concepts as they are presented in the paper and then proceed to new concepts. In the last section, I would like to pose a few questions without attempting a solution, that relate to Castellani and Hennessy's argument for a distributed bisimulation.

The sections are as follows:

NOTE : In general, rules are described using an =, for example $p \mid q = q \mid p$. The authors then state that a particular equivalence relation, for example ~, satisfies the rule. This is contrast to Milner's notation [11]. I will use the authors' notation as it allows for rules to numbered for easy reference. The references to CCS all refer to Milner's book [11] unless otherwise stated.

PROCESSES AS LABELLED TRANSITION SYSTEMS

This section defines a labelled transition system and Σ_1, a signature to generate the processes in our labelled transition system. The labelled transition system will be a subset of CCS. After this, the concept of a distributed labelled transition system will be defined

Definition

A labelled transition system (lts) is a triple $(P, A, \rightarrow)$ where

i) P is a given set of processes

ii) A is a given set of actions

iii) $\rightarrow$ is a relation contained in $(P \times A \times P)$ called a transition relation. ∎

$(p, a, q) \in \rightarrow$ is usually written $p \xrightarrow{a} q$ for $a \in A$ and $p, q \in P$. Also let x, y represent process variables.

Definition

Let Σ_1 be the signature consisting of

i) NIL, a constant or nullary operator;

ii) $a.$ for each $a \in A$ a unary operator called prefixing

iii) $+$, a binary infix operator called choice

iv) $|$, a binary infix operator called parallel composition.

The authors suggest abbreviations that I will not use for reasons of clarity. The precedence of the operators is Prefix, |, +. $\blacksquare$

Use L to denote the word algebra generated by Σ_1

To give L the structure of lts, let

i) P the set of processes, be the set of terms in L

ii) A is the predefined set of actions

iii) $\rightarrow$ the transition relation, be the least relation satisfying the following axioms, $\mathcal{R}$.

(In the paper, the first rule was written '*a:* $p \xrightarrow{a} p$'. This notation was not explained and I have transcribed it as '$\forall\, a \in A,\ a.p \xrightarrow{a} p$'. This point applies to similar rules throughout the paper*)*

Rules $\mathcal{R}$

(R1) Rule 1. $\forall\, a \in A,\ a.p \xrightarrow{a} p$

(R2) Rule 2. $p \xrightarrow{a} p'$ implies $p + q \xrightarrow{a} p'$ and $q + p \xrightarrow{a} p'$

(R3) Rule 3. $p \xrightarrow{a} p'$ implies $p \mid q \xrightarrow{a} p' \mid q$ and $q \mid p \xrightarrow{a} q \mid p'$

The elements of P will be referred to as terms and processes. However later on in the paper, *processes* takes on a very specific meaning.

In this lts, the terms involve prefixing, choice, and parallel composition. There is no restriction, relabelling or constant definition which therefore does not allow recursive definition of processes. The rules given for the lts above are the same as Act (R1), Sum_2 (R2) and Com_1 (R3) and Com_2 (R3) as given on page 46 of Milner[11]. However there is no rule to match Com_3 - this means that there is no communication between processes. As would be expected, $p \xrightarrow{a} q$ is true only if it can be proved by using the three rules (R1) - (R3).

In Milner's CCS, $p \xrightarrow{a} q$ can be interpreted as the response of p to an external demand from an observer for the action a in which it changes to the state q. The authors consider the rule (R3) $p \xrightarrow{a} p'$ implies $p \mid q \xrightarrow{a} p' \mid q$ and $q \mid p \xrightarrow{a} q \mid p'$ and state that this rule implies that the observer is ignoring facts that should be obvious i.e. the facts that different processes can be at different localities and that the local action is *causing* a local result.

The authors assume that $p \mid q$ represents a process consisting of two independent processes which are physically separated - p is at location l_1 and q is at location l_2. The assumption is that the observer is an 'uncomplicated entity' and therefore can only be at one location. The observer knows that the request has satisfied by a subprocess at the observer's location and (in this simple case where there is no communication) this has not affected subprocesses at other localities. The authors have chosen not to name localities or parameterise observers as being at particular localities. *Each* observer is placed at a definite locality. Each observer can request an action from the subprocess at its position, and observe the local result of the action, and simultaneously the observer is informed of the global result of the action. Local and global results must be observed together to be meaningful. Intuitively, this concept captures causality - the local action causes the local result.

A primitive observation takes the form $p \xrightarrow{a} \langle p', p'' \rangle$. This can be interpreted as (i) an observer requests an action of a of process p, (ii) satisfying the demand changes the subprocess local to the point of observation into p' and the whole process evolves to p''.

I interpret this as meaning that a process (consisting of a number of subprocesses) can be viewed as spread over a number of localities with a different 'observer' at each location. When proving two processes are bisimilar, it is required that from each observer's point of view (at the equivalent locations in the different processes), the processes are indistinguishable. I will demonstrate this concept after the definition of distributed bisimulation has been presented.

Definition

A distributed labelled transition system (d-lts) is a triple $(P, A, \rightarrow)$ where

i) P is a given set of processes

ii) A is a given set of actions

iii) $\rightarrow$ is a relation contained in $(P \times A \times P \times P)$ called a transition relation.

$(p,a,p',p'') \in \rightarrow$ is usually written $p \xrightarrow{a} \langle p', p'' \rangle$ for $a \in A$ and $p, p', p'' \in P$ ∎

As given above, L can be viewed as a d-lts by using the transition relation defined by the following rules, $\mathcal{R}'$.

Rules $\mathcal{R}'$

(R1′) Rule 1′. $\forall\, a \in A,\ a.p \xrightarrow{a} \langle p, p \rangle$

(R2′) Rule 2′. $p \xrightarrow{a} \langle p', p'' \rangle$ implies $p + q \xrightarrow{a} \langle p', p'' \rangle$
and $q + p \xrightarrow{a} \langle p', p'' \rangle$

(R3′) Rule 3′. $p \xrightarrow{a} \langle p', p'' \rangle$ implies $p \mid q \xrightarrow{a} \langle p', p'' \mid q \rangle$
and $q \mid p \xrightarrow{a} \langle p', q \mid p'' \rangle$

The rules (R1′) - (R3′) are similar to (R1) - (R3) and the first set of rules can be recovered by ignoring the first component of the residuals. Note that there is no communication implied by these rules.

Examples

These are all distributed transitions that can be performed by the process

$$a.b.\mathrm{NIL} + (b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid (e.\mathrm{NIL} + f.\mathrm{NIL}))$$

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

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

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

$a.b.\mathrm{NIL} + (b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid (e.\mathrm{NIL} + f.\mathrm{NIL})) \xrightarrow{f}$
$\langle \mathrm{NIL}, b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid \mathrm{NIL} \rangle$ ∎

BISIMULATIONS AND DISTRIBUTED BISIMULATIONS

Definition

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:

$p \xrightarrow{a} p'$ implies $q \xrightarrow{a} q'$ for some q' such that $(p',q') \in R$

Bisimulation equivalence is defined by

$p \sim q$ if $(p,q) \in R$ for some bisimulation R ∎

This definition of bisimulation is the same as Milner's strong bisimulation, although it is phrased slightly differently. The second part of the definition can be reformulated as: $\sim\, = \cup \{ R : R$ is a bisimulation $\}$.

As would be expected, the following results hold.

Lemma

(a) ~ is an equivalence relation

(b) ~ is preserved by all operators in Σ_1 - ~ is a Σ_1-congruence. If $p \sim q$ then

$$a.p \sim a.q, \quad p + r \sim q + r, \quad \text{and } p \mid r \sim q \mid r \qquad \blacksquare$$

~ satisfies the following laws:

(P1) $p \mid q = q \mid p$

(P2) $p \mid (q \mid r) = (p \mid q) \mid r$

(P3) $p \mid \text{NIL} = p$

(L1) $a.\text{NIL} \mid b.\text{NIL} \sim a.b.\text{NIL} + b.a.\text{NIL}$

(L1) is a limited form of Milner's expansion law and demonstrates interleaving and the fact that concurrent processes can be expressed as a nondeterministic, sequential process.

Definition

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

$p \xrightarrow{a} \langle p', p'' \rangle$ implies $q \xrightarrow{a} \langle q', q'' \rangle$ for some q', q''

such that $(p',q') \in R$ and $(p'',q'') \in R$

d-bisimulation equivalence is defined by

$$p \sim_d q \quad \text{if } (p,q) \in R \quad \text{for some d-bisimulation } R$$

This can also be expressed as: $\sim_d = \cup \{ R : R \text{ is a d-bisimulation} \}$. ∎

Lemma

(a) $\sim_d$ is an equivalence relation

(b) $\sim_d$ is a Σ_1-congruence. If $p \sim_d q$ then

$$a.p \sim_d a.q, \quad p + r \sim_d q + r, \quad \text{and } p \mid r \sim_d q \mid r \qquad \blacksquare$$

$\sim_d$ satisfies the rules (P1)-(P3) as given above. However it does not satisfy (L1) $a.\text{NIL} \mid b.\text{NIL} \sim a.b.\text{NIL} + b.a.\text{NIL}$, which means this is not an interleaved semantics.

Example

In this example, I show that

$$a.\mathrm{NIL} \mid b.\mathrm{NIL} \not\sim_{\mathrm{d}} a.b.\mathrm{NIL} + b.a.\mathrm{NIL}$$

Each process can perform an a action

$a.\mathrm{NIL} \mid b.\mathrm{NIL} \xrightarrow{a} \langle \mathrm{NIL} , \mathrm{NIL} \mid b.\mathrm{NIL} \rangle$ whereas

$a.b.\mathrm{NIL} + b.a.\mathrm{NIL} \xrightarrow{a} \langle b.\mathrm{NIL} , b.\mathrm{NIL} \rangle$ and $\mathrm{NIL} \not\sim_{\mathrm{d}} b.\mathrm{NIL}$

This means that the two agents cannot be bisimilar ∎

The following proposition relates d-bisimulation to bisimulation. Intuitively, in the part (a), the local residual is ignored.

Proposition

(a) If $p \sim_{\mathrm{d}} q$ then $p \sim q$

(b) If p, q contain no occurrences of |, then $p \sim q$ implies $p \sim_{\mathrm{d}} q$ ∎

Let $\equiv$ be the congruence on L generated by laws (P1) - (P3) namely equality of terms modulo associativity and commutativity of |.

Proposition If $p \xrightarrow{a} \langle q, r \rangle$, then $\exists\, s$ such that $q \mid s \equiv r$ ∎

Example

I modify the semaphore example given in Chapter 1.4 and Exercise 4.1 of Milner to give a very simple example of distributed bisimulation. The modification is required as there is no recursive definition in the d-lts presented. Consider an counter that has the following form $C = add1.\mathrm{NIL}$. To count up to n, copies of the process can be combined as follows. (In this example, '=' means syntactically equivalent)

$C_n = C \mid C \mid ... \mid C$, so that there are n processes in the composition.

A process D_n could also be defined

$D_n = add1.add1. ... .add1.\mathrm{NIL}$, where there are n prefixed actions.

Consider a system as follows:

The aim is to count to 4, by two processes in separate localities counting to 2.

$S1 = C_2 \mid C_2$ and $S2 = D_2 \mid D_2$

where $C_2 = C \mid C$, $D_2 = add1.add1.\mathrm{NIL}$ from the definitions above

A third system could be defined to do centralised counting to 4

$S3 = D_4 = add1.add1.add1.add1.\mathrm{NIL}$

I will now illustrate (without full proof) that $S1 \sim_{\mathrm{d}} S2$ and $S1 \sim S3$, $S2 \sim S3$, but $S1 \not\sim_{\mathrm{d}} S3$, $S2 \not\sim_{\mathrm{d}} S3$

Consider the set

$R1 = \{\ (C_2 \mid C_2, D_2 \mid D_2),\ (\text{NIL} \mid C, add1.\text{NIL}),\ (\text{NIL}, \text{NIL}),$
$\quad (\text{NIL} \mid C \mid C_2, add1.\text{NIL} \mid D_2),\ (\text{NIL} \mid C_2, \text{NIL} \mid D_2),$
$\quad (\text{NIL} \mid C \mid \text{NIL} \mid C, add1.\text{NIL} \mid add1.\text{NIL})\ \}$

$R1$ is a bisimulation and $S1 \sim_d S2$ since $(S1,S2) \in R$

(The bisimulation has been reduced in size by the use of associativity and commutativity.)

When $S1 = C_2 \mid C_2 \xrightarrow{add1} \langle \text{NIL} \mid C, \text{NIL} \mid C \mid C_2 \rangle$ then
$S2 = D_2 \mid D_2 \xrightarrow{add1} \langle add1.\text{NIL}, add1.\text{NIL} \mid D_2 \rangle$ and
$(\text{NIL} \mid C, add1.\text{NIL}) \in R$ and $(\text{NIL} \mid C \mid C_2, add1.\text{NIL} \mid D_2) \in R$

When $S2 = D_2 \mid D_2 \xrightarrow{add1} \langle add1.\text{NIL}, add1.\text{NIL} \mid D_2 \rangle$ then
$S1 = C_2 \mid C_2 \xrightarrow{add1} \langle \text{NIL} \mid C, \text{NIL} \mid C \mid C_2 \rangle$ and
$(\text{NIL} \mid C, add1.\text{NIL}) \in R$ and $(\text{NIL} \mid C \mid C_2, add1.\text{NIL} \mid D_2) \in R$

This argument could be repeated for each pair in R.

The bisimulation to prove $S1 \sim S3$ is as follows:

$R2 = \{\ (C_2 \mid C_2, D_4),\ (\text{NIL} \mid C \mid C_2, add1.add1.add1.\text{NIL}),$
$\quad (\text{NIL} \mid C_2, add1.add1.\text{NIL}),\ (\text{NIL} \mid C \mid \text{NIL} \mid C, add1.add1.\text{NIL}),$
$\quad (\text{NIL} \mid C, add1.\text{NIL})\ \}$

A similar bisimulation can be created to prove $S2 \sim S3$

To show that $S1 \not\sim_d S3$, consider
$\text{NIL} \mid C \mid C_2 \xrightarrow{add1} \langle \text{NIL}, \text{NIL} \mid C_2 \rangle$
$add1.add1.add1.\text{NIL} \xrightarrow{add1} \langle add1.add1.\text{NIL}, add1.add1.\text{NIL} \rangle$
but $add1.add1.\text{NIL} \not\sim_d \text{NIL}$. ∎

ALGEBRAIC CHARACTERISATION OF DISTRIBUTED BISIMULATION

In this section, the axiomatisation of $\sim$ and $\sim_d$ are presented. The axioms for $\sim$, $\mathcal{A}$ are given below.

Axiomatisation of $\sim$: Axioms $\mathcal{A}$

(A1) $x + (y + z) = (x + y) + z$

(A2) $x + y = y + x$

(A3) $x + \text{NIL} = x$

(A4) $x + x = x$

(IN) If $x = \Sigma_{i=1}^{n} a_i.\, x_i$ and $y = \Sigma_{j=1}^{m} b_j.y_j$ then
$$x \mid y = \Sigma_{i=1}^{n} a_i.\,(x_i \mid y) + \Sigma_{j=1}^{m} b_j.\,(x \mid y_j)$$

Note that $\Sigma_{i=1}^{n}\ a_i.\ x_i$ will usually be written $\Sigma\ a_i.\ x_i$

$$\Sigma_{i=1}^{n}\ a_i.\ p_i \ = \ a_1.\ p_1 + \ ...\ + \ a_n.\ p_n \qquad \text{if } n > 0$$
$$= \ \text{NIL} \qquad \text{if } n = 0$$

The axiom (IN) is a rephrasing of the expansion law excluding communication. It illustrates the interleaved semantics of ~.

Example

Considering the actions that x can perform

$\Sigma\ a_i.\ x_i \ \xrightarrow{a_i}\ x_i$ for each i

and y can perform

$\Sigma\ b_j.y_j \ \xrightarrow{a_i}\ y_j$ for each j

By expansion

$x \mid y = \ \Sigma\ \{\ a_i.(\ x_i \mid y\ ) : x \ \xrightarrow{a_i}\ x_i\ \} + \ \Sigma\ \{\ b_j.(\ x \mid y_j\ \ ) : y \ \xrightarrow{a_i}\ y_j\ \ \}$ ∎

Theorem The equivalence ~ is the Σ_1 congruence over L generated by the axioms $\mathcal{A}$
This means that $p =_{\mathcal{A}} q$ if and only if $p \sim q$ ∎

It is necessary for this type of proof to convert all processes into normal form. In this case, the normal form will be $\Sigma_{i=1}^{n}\ a_i.\ p_i$

All occurrences of | can be removed using (IN), so that concurrency is reduced to nondeterminism. The primitive operators are prefixing, choice and NIL. In Chapter 7.4 of Milner, an axiomatisation is presented for finite serial agents, namely those agents using the operators finite Summation (choice) and Prefix. Any agent containing finite Summation, Composition, Restriction or Relabelling, (but no Constants or Recursions) can be equated by expansion to a finite serial agent. Since the parallel operator is therefore excluded, the axioms (A1) - (A4) are used for Milner's axiomatisation.

Earlier it was shown that (LP1) is not valid for $\sim_d$ and hence (IN) is not valid. Therefore | cannot be removed from the axiomatisation of $\sim_d$, as was the case with ~.

Example

This example shows that (IN) is not valid for $\sim_d$

Consider $x = \Sigma\ a_i.\ \text{NIL}$ and $y = \Sigma\ b_j.\text{NIL}$ then

then

$x \mid y \ \xrightarrow{a_i}\ \langle\ \text{NIL}, \text{NIL} \mid y\ \ \rangle$

$\Sigma_{i=1}^{n}\ a_i.\ (\ \text{NIL} \mid y\ ) + \Sigma_{j=1}^{m}\ b_j.\ (\ x \mid \text{NIL})\ \ \xrightarrow{a_i} \langle\ \text{NIL} \mid y, \text{NIL} \mid y\ \ \rangle$

and NIL $\not\sim_d$ NIL $\mid y$ ∎

The next issue is the axiomatisation of $\sim_d$. Looking at this intuitively, when $p \xrightarrow{a} \langle\ q, r\ \rangle$, this is because p contains an initial subterm $a.q$. In an earlier proposition it was shown that $\exists\, s$ such that $q \mid s \equiv r$. s is a term concurrent to $a.q$ in p called a coterm of $a.q$ in p. The behaviour of p as seen from a specific locality is determined by the set of its initial subterms $a.q$ and their coterms, s. Generally coterms are not subterms. For example

$$p = ((\, a.q \mid r\,) + u\,) \mid t$$

The coterm of $a.q$ in p is $r \mid t$ since $p \xrightarrow{a} q \mid r \mid t$. Therefore the equality of two terms cannot be proved by comparing their subterms and respective coterms, because the coterms are not subterms.

To get around this problem, a new operator $\lceil$, is introduced which can be viewed as an asymmetric parallel operator. $p \lceil q$ are concurrent, but p has initial dominance. p can then be expressed in the explicit form

$$\Sigma_{i \in I}\, a_i.\, p_i \lceil p_i' \text{ where for each } i \in I,\ p_i' \text{ is the coterm of } a_i.\, p_i$$

Intuitively, the $a_i.\, p_i$ represent the localities where an action can be requested.

Example

As an example, the process $(a.q \mid b.(c.p + d.r)) + e.s$ will be given in its explicit form and all its transitions will be listed.

$(a.q \mid b.(c.p + d.r)) + e.s \;=\; a.q \lceil b.(c.p + d.r) \;+\; b.(c.p + d.r) \lceil a.q +\; e.s \lceil \mathrm{NIL}$

$(a.q \mid b.(c.p + d.r)) + e.s \xrightarrow{a} q \mid b.(c.p + d.r)$

$(a.q \mid b.(c.p + d.r)) + e.s \xrightarrow{b} a.q \mid (c.p + d.r)$

$(a.q \mid b.(c.p + d.r)) + e.s \xrightarrow{e} s$ ∎

The new transition rules for $\lceil$ are as follows:

(R4)	Rule 4. $p \xrightarrow{a} p''$	implies	$p \lceil q \xrightarrow{a} p'' \mid q$
(R4′)	Rule 4’. $p \xrightarrow{a} \langle p', p'' \rangle$	implies	$p \lceil q \xrightarrow{a} \langle p', p'' \mid q \rangle$

Example

Taking the explicit form of p given above, the following transitions can occur.

$\Sigma\, a_i.\, p_i \xrightarrow{a_i} \langle\, p_i\,,\; p_i\, \rangle$

$\Sigma\, a_i.\, p_i \lceil p_i' \xrightarrow{a_i} \langle\, p_i\,,\; p_i \mid p_i' \,\rangle$

The initial subterms of the process are the $a_i.\, p_i$ and the coterms are p_i' and the global state is $p_i \mid p_i'$ after an a_i transition.

From the example above, taking the explicit form of $(a.q \mid b.(c.p + d.r)) + e.s$, it can be seen that it has the same transitions.

$a.q \lceil b.(c.p + d.r) \;+\; b.(c.p + d.r) \lceil a.q \;+\; e.s \lceil \mathrm{NIL} \xrightarrow{a} q \mid b.(c.p + d.r)$

$a.q \lceil b.(c.p + d.r) \;+\; b.(c.p + d.r) \lceil a.q \;+\; e.s \lceil \mathrm{NIL} \xrightarrow{b} a.q \mid (c.p + d.r)$

$a.q \lceil b.(c.p + d.r) \;+\; b.(c.p + d.r) \lceil a.q \;+\; e.s \lceil \mathrm{NIL} \xrightarrow{e} s$ $\blacksquare$

There are equations involving the new operator. The law (LP1) relates $\mid$ to $\lceil$.

(LP1) $\quad p \mid q \;=\; p \lceil q \;+\; q \lceil p$

This is straightforward as the two terms on the right hand side can simulate all that that the left hand side can do and *vice versa*, considering the rules for $\mid$ and $\lceil$.

(R3′) $\quad p \xrightarrow{a} \langle p', p'' \rangle$ implies $p \mid q \xrightarrow{a} \langle p', p'' \mid q \rangle$ and $q \mid p \xrightarrow{a} \langle p', q \mid p'' \rangle$

(R4′) $\quad p \xrightarrow{a} \langle p', p'' \rangle$ implies $p \lceil q \xrightarrow{a} \langle p', p'' \mid q \rangle$

This new operator does not give an interleaved semantics, since $\sim_{\mathrm{d}}$ does not satisfy (IN) as shown earlier. There is a new interleaving rule for $\lceil$, ($\lceil$IN) which is satisfied by $\sim$ but not by $\sim_{\mathrm{d}}$.

($\lceil$IN) $\quad a.p \lceil q = a.(p \mid q)$

Example

The first part of the example, shows how ($\lceil$IN) holds for $\sim$.

When $a.p \lceil q \xrightarrow{a} p \mid q$ then $a.(p \mid q) \xrightarrow{a} p \mid q$ and $p \mid q \sim p \mid q$. This is the only action that can be performed by $a.p \lceil q$.

When $a.(p \mid q) \xrightarrow{a} p \mid q$ then $a.p \lceil q \xrightarrow{a} p \mid q$ and $p \mid q \sim p \mid q$. This is the only action that can be performed by $a.(p \mid q)$.

Therefore $a.p \lceil q \sim a.(p \mid q)$

It is shown that ($\lceil$IN) does not hold for $\sim_{\mathrm{d}}$

$a.p \lceil q \xrightarrow{a} \langle p,\; p \mid q \rangle$

$a.(p \mid q) \xrightarrow{a} \langle p \mid q,\; p \mid q \rangle$ and $p \not\sim_{\mathrm{d}} p \mid q$. $\blacksquare$

$\lceil$ satisfies a distributive law as follows which allows syntactic manipulations.

(LP2) $\quad (p + q) \lceil r = (p \lceil r) + (q \lceil r)$

(LP3) $\quad (p \lceil q) \lceil r = p \lceil (q \mid r)$

(LP4) $\quad p \lceil \mathrm{NIL} = p$

(LP5) $\quad \mathrm{NIL} \lceil p = \mathrm{NIL}$

Examples

This example illustrates (LP2)

$(a.\mathrm{NIL} + b.\mathrm{NIL}) \lceil c.\mathrm{NIL} \xrightarrow{a} \langle \mathrm{NIL},\ \mathrm{NIL} \mid c.\mathrm{NIL} \rangle$

$(a.\mathrm{NIL} + b.\mathrm{NIL}) \lceil c.\mathrm{NIL} \xrightarrow{b} \langle \mathrm{NIL},\ \mathrm{NIL} \mid c.\mathrm{NIL} \rangle$

$(a.\mathrm{NIL} \lceil c.\mathrm{NIL}) + (b.\mathrm{NIL} \lceil c.\mathrm{NIL}) \xrightarrow{a} \langle \mathrm{NIL},\ \mathrm{NIL} \mid c.\mathrm{NIL} \rangle$

$(a.\mathrm{NIL} \lceil c.\mathrm{NIL}) + (b.\mathrm{NIL} \lceil c.\mathrm{NIL}) \xrightarrow{b} \langle \mathrm{NIL},\ \mathrm{NIL} \mid c.\mathrm{NIL} \rangle$

This example illustrates (LP3)

$(a.\mathrm{NIL} \lceil b.\mathrm{NIL}) \lceil c.\mathrm{NIL} \xrightarrow{a} \langle \mathrm{NIL},\ \mathrm{NIL} \mid b.\mathrm{NIL} \mid c.\mathrm{NIL} \rangle$

$a.\mathrm{NIL} \lceil (b.\mathrm{NIL} \mid c.\mathrm{NIL}) \xrightarrow{a} \langle \mathrm{NIL},\ \mathrm{NIL} \mid b.\mathrm{NIL} \mid c.\mathrm{NIL} \rangle$ ∎

Let Σ_2 be the signature with $\lceil$ included, and *EL* the extended language generated by this.

Axiomatisation of $\sim_{\mathrm{d}}$: Axioms $\mathcal{B}$

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

Theorem (Characterisation) The equivalence $\sim_{\mathrm{d}}$ is the Σ_2 congruence over *EL* generated by the axioms $\mathcal{B}$, i.e. $p =_{\mathcal{B}} q$ if and only if $p \sim_{\mathrm{d}} q$

Proof outline

(⇒) (Soundness)

If $=_{\mathcal{B}}$ is replaced with $\sim_{\mathrm{d}}$ in each axiom, they are still hold.

(⇐) (Completeness)

Definition $\Sigma\, a_i.\, p_i \lceil p_i'$ is a normal form (nf) whenever all p_i, p_i' are nfs. For n=0, NIL is a normal form. The depth $d(p)$ of p is also defined.

Normalization Lemma For any $p \in EL$, there exists a nf n such that $p =_{\mathcal{B}} n$. Also

$d(p) = d(n)$

Simplification Lemma For any $p,q,r \in EL$, $p \mid r \sim_{\mathrm{d}} q \mid r$ implies $p \sim_{\mathrm{d}} q$

The proof for completeness is by induction on the sum of the sizes of p and q. Suppose $p \sim_{\mathrm{d}} q$. p and q can be written as nfs. Then it is shown that $p + q =_{\mathcal{B}} q$. To show this it is sufficient to show that $q + a_i.\, p_i \lceil p_i' =_{\mathcal{B}} q$ for each i . This requires the use of the simplification lemma. By symmetry $p + q =_{\mathcal{B}} p$ from which the result follows. ∎

The concept of an internal unobservable action is introduced into the calculus. At this point it is only a distinguished symbol and does not represent communication between processes.

Assume A contains a distinguished symbol τ representing an internal unobservable action. Then $A = O \cup \tau$. The symbols a,b,c will range over O and μ,ν will range over A. The observation relation $\xrightarrow{a}$ will be replaced with $\stackrel{a}{\Rightarrow}$, where $\stackrel{a}{\Rightarrow}$ can be viewed as a number of internal actions (possibly none) followed by an a action followed by more internal actions (again possibly none). τ actions are regarded as global actions that can be localised only indirectly, if it affects the observable behaviour of the component where it occurs.

Example

In $\tau.p \mid q$, the τ action can be regarded as global as it does not affect how each subcomponent acts. However in $(\tau.p + q) \mid r$, the τ action can pre-empt the actions of q in the first component and thus can affect the local actions of this component, by preventing some actions from occurring. ∎

The definition of $\stackrel{a}{\Rightarrow}$ becomes complicated using the current notation. When $p \stackrel{a}{\Rightarrow} \langle q, r \rangle$, q is the local residual of the observation, whereas r is the global residual which includes q. It becomes problematic to maintain the consistency in $\langle q, r \rangle$ between q and the copy of q in r. By placing q in a global context C[], this can be rewritten $p \stackrel{a}{\Rightarrow} \langle q, C[\mathrm{q}] \rangle$, which can be further reduced to $p \stackrel{a}{\Rightarrow} C[\mathrm{q}]$. In C[], [] gives the locality of the observation and q the local residual, therefore C[q] represents the full details of $\langle q, C[\mathrm{q}] \rangle$.

These concepts are defined formally.

Definition The set of generalised terms are defined by the following grammar

$$t ::= \mathrm{NIL} \mid t + t \mid t \,|\, t \mid t \upharpoonleft t \mid [\,] \mid [t]$$ ∎

The authors state that all elements of EL are generalised terms. The elements in EL will be called terms, with $p,q \in EL$. This is the language defined Σ_2, defined earlier, where Σ_2 consists of NIL, $a.$ for each $a \in A$, $+$, $\mid$ and $\upharpoonleft$.

I find that this leads to a conflict in definitions as *EL* allows terms with prefixing, for example a.NIL, whereas this is not defined for generalized terms. In fact, generalised terms consist only of the nullary operator NIL combined in different ways. I suggest that this is a mistake and that generalised terms should be defined by the grammar.

$$t ::= \mathrm{NIL} \;\big|\; t + t \;\big|\; t \,|\, t \;\big|\; t \lceil t \;\big|\; [\,] \;\big|\; [t] \;\big|\; \mu.t\,, \quad \mu \in A$$

The grammar for *EL* could be written in a similar fashion as:

$$e ::= \mathrm{NIL} \;\big|\; e + e \;\big|\; e \,|\, e \;\big|\; e \lceil e \;\big|\; \mu.e\,, \quad \mu \in A$$

Examples

$a.(b.\mathrm{NIL} + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ is a term and also a generalised term.

$a.([b.\mathrm{NIL}] + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ is a generalised term, but not a term. $\blacksquare$

Definition

A context is a generalised term with no occurrence of a subterm of the form $[t]$ in it and at most one occurrence of $[\,]$ (there could be none). Contexts will usually be denoted $C[\,]$ $\blacksquare$

Examples

$a.([\,] + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ and $a.(b.\mathrm{NIL} + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ are contexts. $\blacksquare$

Definition

An instantiated context, $C[t]$, is a generalised term obtained by substituting $[t]$ for the unique occurrence of $[\,]$, if it exists. $\blacksquare$

Examples

The following give examples of instantiated contexts.

If $C[\,] = a.([\,] + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ then $C[f.\mathrm{NIL}] = a.([f.\mathrm{NIL}] + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$

If $C[\,] = a.(b.\mathrm{NIL} + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ then $C[f.\mathrm{NIL}] = a.(b.\mathrm{NIL} + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ $\blacksquare$

Definition

A process is an instantiated context of the form $C[p]$, where $p \in EL$. I will refer to the set of processes as *PR*. $\blacksquare$

Examples

If $C[\,] = a.([\,] + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ then

$C[f.\mathrm{NIL}] = a.([f.\mathrm{NIL}] + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ is a process since $f.\mathrm{NIL} \in EL$

and

$C[f.\mathrm{NIL}+[\mathrm{a.NIL}]\,] = a.([f.\mathrm{NIL}+[\mathrm{a.NIL}]\,] + c.\mathrm{NIL}) \,|\, d.\mathrm{NIL}$ is not a process since $f.\mathrm{NIL}+[\mathrm{a.NIL}] \notin EL$ $\blacksquare$

P, Q will range over processes. All terms in EL are processes, but the reverse is not true. In $C[p]$, the occurrence of $[p]$ indicates the locality under observation. Processes can be viewed as terms in EL by ignoring local details, replacing $[p]$ by p.

Definition

$C(p) \in EL$ is the term corresponding to the process $C[p]$ and is obtained by substituting p rather than $[p]$, for $[]$ in $C[]$

Example

If $C[] = a.([] + c.\mathrm{NIL}) \mid d.\mathrm{NIL}$ then $C(f.\mathrm{NIL}) = a.(f.\mathrm{NIL} + c.\mathrm{NIL}) \mid d.\mathrm{NIL}$ ▮

The authors define weak observation, $\overset{a}{\Rightarrow}$, in terms of $\overset{\tau}{\rightarrow}$ and $\overset{a}{\rightarrow}$.

$\overset{\tau}{\rightarrow}$ is taken to be the least relation that satisfies the τ-Rules.

τ-Rules (for General Processes).

(Rτ1) Rule τ1. $\tau.P \overset{\tau}{\rightarrow} P$

(Rτ2) Rule τ2. $P \overset{\tau}{\rightarrow} P'$ implies $P + Q \overset{\tau}{\rightarrow} P'$ and $Q + P \overset{\tau}{\rightarrow} P'$

(Rτ3) Rule τ3. $P \overset{\tau}{\rightarrow} P'$ implies $P \mid Q \overset{\tau}{\rightarrow} P' \mid Q$ and $Q \mid P \overset{\tau}{\rightarrow} Q \mid P'$

(Rτ4) Rule τ4. $P \overset{\tau}{\rightarrow} P'$ implies $P \lceil Q \overset{\tau}{\rightarrow} P' \mid Q$

(Rτ5) Rule τ5. $p \overset{\tau}{\rightarrow} p'$ implies $[p] \overset{\tau}{\rightarrow} [p']$

Rule τ5 allows the τ action to be localised when necessary. The reason for using processes in these rules is that τ actions are global actions.

Examples

These are examples of global τ transitions.

$[a.\mathrm{NIL}] + \tau.p \overset{\tau}{\rightarrow} p$

$[q] \mid \tau.p \overset{\tau}{\rightarrow} [q] \mid \mathrm{p}$ ▮

$\overset{a}{\rightarrow}$ takes elements of EL and returns processes. They are defined to be least relations that satisfy the a-Rules. These rules deal with observable actions between terms in EL and processes.

***a*-Rules (from Terms to Processes)**

(E1) Rule 1. $\forall\, a \in O,\ a.p \overset{a}{\rightarrow} [p]$

(E2) Rule 2. $p \overset{a}{\rightarrow} C[p']$ implies $p + q \overset{a}{\rightarrow} C[p']$ and $q + p \overset{a}{\rightarrow} C[p']$

(E3) Rule 3. $p \overset{a}{\rightarrow} C[p']$ implies $p \mid q \overset{a}{\rightarrow} C[p'] \mid q$ and $q \mid p \overset{a}{\rightarrow} q \mid C[p']$ and $p \lceil q \overset{a}{\rightarrow} C[p'] \mid q$

Examples

These are examples of observable actions by the agent

$$a.b.\mathrm{NIL} + (b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid (e.\mathrm{NIL} + f.\mathrm{NIL}))$$

$a.b.\mathrm{NIL} + (b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid (e.\mathrm{NIL} + f.\mathrm{NIL})) \xrightarrow{a} [\, b.\mathrm{NIL}\,]$

$a.b.\mathrm{NIL} + (b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid (e.\mathrm{NIL} + f.\mathrm{NIL})) \xrightarrow{b}$
$[\,(d.\mathrm{NIL} + e.\mathrm{NIL})\,] \mid (e.\mathrm{NIL} + f.\mathrm{NIL})$

$a.b.\mathrm{NIL} + (b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid (e.\mathrm{NIL} + f.\mathrm{NIL})) \xrightarrow{e}$
$b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid [\mathrm{NIL}]$

$a.b.\mathrm{NIL} + (b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid (e.\mathrm{NIL} + f.\mathrm{NIL})) \xrightarrow{f}$
$b.(d.\mathrm{NIL} + e.\mathrm{NIL}) \mid [\mathrm{NIL}]$ ∎

In combining $\xrightarrow{a}$ and $\xrightarrow{\tau}$, the relation $\xRightarrow{a}$ from *EL* to *PR* can be defined.

i) $p \xRightarrow{\tau} q$ if $p\ (\xrightarrow{\tau})^{+}\ q$

ii) $p \xRightarrow{\epsilon} q$ if $p\ (\xrightarrow{\tau})^{*}\ q$

iii) $p \xRightarrow{a} C[p']$ if $p\ (\xrightarrow{\tau})^{*}\ q \xrightarrow{a} D[q']\ (\xrightarrow{\tau})^{*}\ C[p']$

where $(\xrightarrow{\tau})^{+}$ denotes the transitive closure of $\xrightarrow{\tau}$ (this means that at least one internal action must be performed), and $(\xrightarrow{\tau})^{*}$ its transitive and reflexive closure (this means that zero or more internal actions can be performed).

Examples

These are examples of $\xRightarrow{a}$ observations. $\xrightarrow{\epsilon}$ is used to denote no τ transitions.

$a.p \mid (q + \tau.r) \xRightarrow{a} [p] \mid r$

since $a.p \mid (q + \tau.r) \xrightarrow{\tau} a.p \mid r \xrightarrow{a} [p] \mid r \xrightarrow{\epsilon} [p] \mid r$

$a.p \mid (q + \tau.r) \xRightarrow{a} [p] \mid (q + \tau.r)$

since $a.p \mid (q + \tau.r) \xrightarrow{\epsilon} a.p \mid (q + \tau.r) \xrightarrow{a} [p] \mid (q + \tau.r) \xrightarrow{\epsilon} [p] \mid (q + \tau.r)$ ∎

Let $\equiv$ be the congruence on L generated by laws (P1) - (P3) as previously, namely equality of terms module associativity and commutativity.

Lemma If $p \xrightarrow{a} C[p']$ then $\exists\, s$ such that $[\,] \mid s \equiv C[\,]$ ∎

This proposition links the two definitions of distributed transitions.

Proposition If p contains no occurrences of τ, then $p \xRightarrow{a} C[q]$ if and only if $\exists\, r \equiv C[q]$ such that $p \xrightarrow{a} \langle q, r \rangle$ ∎

WEAK DISTRIBUTED BISIMULATION

A behavioural equivalence that abstracts from internal actions is now defined. Both $\stackrel{a}{\Rightarrow}$ and $\stackrel{\epsilon}{\Rightarrow}$ must be considered, since the authors are dealing with the evolution of a process after a finite amount of time, therefore both cases are possible.

Definition 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) $p \stackrel{\epsilon}{\Rightarrow} p'$ implies $q \stackrel{\epsilon}{\Rightarrow} q'$ for some q' such that $(p', q') \in R$,

ii) $p \stackrel{a}{\Rightarrow} C[p']$ implies $q \stackrel{a}{\Rightarrow} D[q']$ for some q' such that $(p', q') \in R$

and $(\boldsymbol{C[p']}, \boldsymbol{D[q']}) \in R$

This can be expressed as $p \approx_d q$ if $(p,q) \in R$, for some weak d-bisimulation R ▮

I find that there is a major problem with this definition. The instantiated contexts $C[p']$ and $D[q']$ possibly contain subterms of the form $[p']$ and $[q']$ respectively. These subterms denote the locality of the action a. This means that the ordered pair

$$(C[p'], D[q']) \notin (EL \times EL)$$

since EL is the extended language generated by Σ_2 which contains the prefix operator, the choice operator, the parallel operator and the asymmetric parallel operator; and does not contain [] and $[t]$.

There are two possible changes that could be made to make this definition more meaningful. I will state them both, but I will use the second one as it is more complete and simpler.

If the definition is changed so that $R \subseteq (PR \times PR)$ then rules are required to map a actions from Processes to Processes. Rules iv) to vi) (on the following page) would be required in addition to i) - iii). iv) and v) come straight from the τ-Rules given above. vi) is problematic since a-Rules map from Terms to Processes, and mappings from Processes to Processes are required. Rules to map observable actions from processes to processes are introduced later in the paper, but only for cases where a new locality is not introduced into the new process.

i) $p \overset{\tau}{\Rightarrow} q$ if $p\,(\overset{\tau}{\rightarrow})^{+}\,q$

ii) $p \overset{\epsilon}{\Rightarrow} q$ if $p\,(\overset{\tau}{\rightarrow})^{*}\,q$

iii) $p \overset{a}{\Rightarrow} C[p']$ if $p\,(\overset{\tau}{\rightarrow})^{*}\,q \overset{a}{\rightarrow} D[q']\,(\overset{\tau}{\rightarrow})^{*}\,C[p']$

iv) $C[p'] \overset{\tau}{\Rightarrow} D[q']$ if $C[p']\,(\overset{\tau}{\rightarrow})^{+}\,D[q']$

v) $C[p'] \overset{\epsilon}{\Rightarrow} D[q']$ if $C[p']\,(\overset{\tau}{\rightarrow})^{*}\,D[q']$

vi) $C[p'] \overset{a}{\Rightarrow}$ **?** if $C[p']\,(\overset{\tau}{\rightarrow})^{*}\,D[q'] \overset{a}{\rightarrow}$ **?** $(\overset{\tau}{\rightarrow})^{*}$ **?**

The definition of weak d-bisimulation should also be revised to cater for these new transitions.

The second option is that the text should read $(C(p'), D(q')) \in R$, since by definition $C(p')$ and $D(q')$ are in EL. This solution is simpler, and I will use this one.

Lemma

(a) $\approx_d$ is an equivalence relation ∎

The following proposition relates d-bisimulation to weak d-bisimulation.

Proposition

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 $p \sim_d q$ ∎

$\approx_d$ is not preserved by +, therefore it is not a congruence. $\approx_d$ is preserved by all other operators.

Example

In this example, I will demonstrate that $p + r \approx_d q + r$, does not necessarily follow from $p \approx_d q$.

Consider b.NIL + $\tau.a$.NIL and b.NIL + a.NIL

$\tau.a.\mathrm{NIL} \overset{a}{\Rightarrow} [\mathrm{NIL}]$, $\tau.a.\mathrm{NIL} \overset{\epsilon}{\Rightarrow} a.\mathrm{NIL}$

$a.\mathrm{NIL} \overset{a}{\Rightarrow} [\mathrm{NIL}]$, $a.\mathrm{NIL} \overset{\epsilon}{\Rightarrow} a.\mathrm{NIL}$

and NIL $\approx_d$ NIL, a.NIL $\approx_d$ a.NIL

therefore $\tau.a.\mathrm{NIL} \approx_d a.\mathrm{NIL}$

However

$b.\mathrm{NIL} + \tau.a.\mathrm{NIL} \overset{\epsilon}{\Rightarrow} a.\mathrm{NIL}$

$b.\mathrm{NIL} + a.\mathrm{NIL} \overset{\epsilon}{\Rightarrow} b.\mathrm{NIL} + a.\mathrm{NIL}$

and $a.\mathrm{NIL} \not\approx_d b.\mathrm{NIL} + a.\mathrm{NIL}$

therefore $b.\mathrm{NIL} + \tau.a.\mathrm{NIL} \not\approx_d b.\mathrm{NIL} + a.\mathrm{NIL}$ ∎

Definition

$p \approx_d^c q$ if for every context $C[]$: $C[p] \approx_d C[q]$

This defines the equivalence to be the largest Σ_2 congruence generated by $\approx_d$. ∎

This definition has similar problem as discussed earlier. $C[p]$, $C[q] \notin EL$, and I suggest that this be rewritten as $C(p) \approx_d C(q)$. This means that all subterms of the form $[t]$ will be replaced with t.

The crucial part of the definition is that it excludes agents p and q that have the property that $p \approx_d q$, but $p + r \not\approx_d q + r$ for some r. It does this as follows:
$p \approx_d^c q$ implies $p \approx_d q$ since $C[] = []$ is a context. Also for every $r \in EL$,
$C[] = [] + r$ is a context, so $p + r \approx_d q + r$, therefore the problem case cannot arise.
In general these problematic terms are the one that are equated by the law $\tau.p \approx_d p$.

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

Property $p \approx_d^c q$ and $p \overset{\tau}{\Rightarrow} p'$ implies $q \overset{\tau}{\Rightarrow} q'$ for some q' such that $p' \approx_d q'$ ∎

Corollary $p \approx_d q$ if and only if one of the following holds:

i) $p \approx_d^c q$

ii) $p \approx_d^c \tau q$

iii) $\tau p \approx_d^c q$ ∎

ALGEBRAIC CHARACTERISATION OF WEAK DISTRIBUTED BISIMULATION

The following axiomatization is presented for $\approx_d^c$.

Axiomatisation of $\approx_d^c$ **(in the absence of communication) :**	**Axioms** $\mathcal{C} = \mathcal{B} \cup \tau$-laws
(A1) $x + (y + z) = (x + y) + z$	(LP1) $x \mid y = x \lceil y + y \lceil x$
(A2) $x + y = y + x$	(LP2) $(x + y) \lceil z = (x \lceil y) + (y \lceil z)$
(A3) $x + \text{NIL} = x$	(LP3) $(x \lceil y) \lceil z = x \lceil (y \mid z)$
(A4) $x + x = x$	(LP4) $x \lceil \text{NIL} = x$
(I1) $x + \tau.x = \tau$	(LP5) $\text{NIL} \lceil x = \text{NIL}$
(I2) $\mu.\tau.x = \mu.x$	
(I3) $\mu.(x + \tau.y) + \mu.y = \mu.(x + \tau.y)$	
(NI1) $\tau.x \lceil y = \tau.(x \lceil y)$	
(NI2) $x \lceil \tau.y = x \lceil y$	

(NI3) $x \upharpoonleft (y + \tau.z) + x \upharpoonleft z = x \upharpoonleft (y + \tau.z)$

Theorem (Characterisation) The equivalence $\approx_d^c$ is the Σ_2 congruence over EL generated by the axioms $\mathcal{C}$, i.e. $p =_{\mathcal{C}} q$ if and only if $p \approx_d^c q$

Proof outline

(⇒) (Soundness)

It is sufficient to check that $a+p \approx_d a+q$ for each instantiation of $p=q$ in $\mathcal{C}$ where a is not in p or q.

(⇐) (Completeness)

Definition $\Sigma\, a_i.\, p_i \upharpoonleft p_i' + \Sigma\, \tau.p_j$ is a weak normal form (wnf) whenever all p_i, p_i', p_j are wnfs.

The second summation is required in the definition of wnf to cater for any global τ actions that might occur.

Normalisation Lemma For any $p \in EL$, there exists a wnf n such that $p =_{\mathcal{C}} n$. Also $d(p) = d(n)$

Simplification Lemma For any $p,q,r \in \mathrm{P}'$, $p \mid r \approx_d q \mid r$ implies $p \approx_d q$

τ-Absorption Lemma If p is a wnf and $p \stackrel{\tau}{\Rightarrow} p'$, then $p + \tau.p' =_{\mathcal{C}} p$

Generalised Absorption Lemma If p is a wnf and $p \stackrel{a}{\Rightarrow} C[p'] \equiv [p'] \mid r$, then $p + a.p' \upharpoonleft r =_{\mathcal{C}} p$

The proof for completeness is by induction on sum of the sizes of p and q. Suppose $p \approx_d^c q$. p and q can be written as wnfs. Then it is shown that $p + q =_{\mathcal{C}} q$. To show this it is sufficient to show that $q + a_i.\, p_i \upharpoonleft p_i' =_{\mathcal{C}} q$ for each i which requires the simplification lemma and the general absorption lemma, and that $q + \tau.p_j =_{\mathcal{C}} q$ for each j which requires the fact that $\mu.\tau.x = \mu.x$ and the τ-absorption lemma. By symmetry $p + q =_{\mathcal{C}} p$ and the result follows. ∎

COMMUNICATION AS MUTUAL OBSERVATION

The authors adopt Milner's model of communication, which is a standard method. Let O, the observable actions be $\Lambda \cup \bar{\Lambda}$, where Λ is a given set of action names and $\bar{\Lambda}$ is the set of their formal complements, $\bar{\Lambda} = \{\, \bar{a} \mid a \in \Lambda \}$ and $\bar{\bar{a}} = a$, $\forall\, a \in O$.

Communication is defined as simultaneous occurrence of two complementary actions. Any communication is treated as an internal action and denoted τ. These are global actions and therefore global actions that do not produce local residuals, are required.

The transitions $\xrightarrow{a}$ are extended to general processes. This relationship currently maps from terms in *EL* to processes. 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 into the residual. These global transitions will be denoted by $\stackrel{a}{\mapsto}$, and the resulting τ transitions by $\stackrel{\tau}{\mapsto}$

The following rules hold for global actions $\stackrel{\mu}{\mapsto}$

Global transitions (for General Processes)	
(F1)	Rule F1. $\forall\, \mu \in A,\ \mu.P \stackrel{\mu}{\mapsto} P$
(F2)	Rule F2. $P \stackrel{\mu}{\mapsto} P'$ implies $P + Q \stackrel{\mu}{\mapsto} P'$ and $Q + P \stackrel{\mu}{\mapsto} P'$
(F3)	Rule F3. $P \stackrel{\mu}{\mapsto} P'$ implies $P \mid Q \stackrel{\mu}{\mapsto} P' \mid Q$ and $Q \mid P \stackrel{\mu}{\mapsto} Q \mid P'$ and $P \restriction Q \stackrel{\mu}{\mapsto} P' \mid Q$
(F4)	Rule F4. $p \stackrel{\mu}{\mapsto} p'$ implies $[p] \stackrel{\mu}{\mapsto} [p']$
(F5)	Rule F5. $P \stackrel{a}{\mapsto} P'$ $Q \stackrel{\bar{a}}{\mapsto} Q'$ implies $P \mid Q \stackrel{\tau}{\mapsto} P' \mid Q'$ (Communication rule)

These are similar to the τ-rules (Rτ1) - (Rτ5) given earlier with an additional rule for communication between processes. For a process of the form $C[p]$, $\stackrel{\mu}{\mapsto}$ maintains the existing locality in the process without introducing a new one.

The global transitions $\stackrel{a}{\mapsto}$ are used for communications. The resulting transitions $\stackrel{\tau}{\mapsto}$ are used with the distributed transitions $\xrightarrow{a}$ to obtain the weak relations $\stackrel{a}{\Rightarrow}$ and $\stackrel{\tau}{\Rightarrow}$

i) $p \stackrel{\tau}{\Rightarrow} q$ if $p\ (\stackrel{\tau}{\mapsto})^{+}\ q$

ii) $p \stackrel{\epsilon}{\Rightarrow} q$ if $p\ (\stackrel{\tau}{\mapsto})^{*}\ q$

iii) $p \stackrel{a}{\Rightarrow} C[p']$ if $p\ (\stackrel{\tau}{\mapsto})^{*}\ q \xrightarrow{a} D[q']\ (\stackrel{\tau}{\mapsto})^{*}\ C[p']$

Example

The actions that can be performed by $a.b.\mathrm{NIL} \mid (c.\mathrm{NIL} + \bar{a}.\mathrm{NIL})$ are the following

$a.b.\mathrm{NIL} \mid (c.\mathrm{NIL} + \bar{a}.\mathrm{NIL}) \xrightarrow{\bar{a}} a.b.\mathrm{NIL} \mid \mathrm{NIL}$

$a.b.\mathrm{NIL} \mid (c.\mathrm{NIL} + \bar{a}.\mathrm{NIL}) \xrightarrow{a} [b.p] \mid (c.\mathrm{NIL} + \bar{a}.\mathrm{NIL})$

These two actions $\bar{a}$, a can be performed because there is no restriction.

$a.b.\mathrm{NIL} \mid (c.\mathrm{NIL} + \bar{a}.\mathrm{NIL}) \stackrel{b}{\Rightarrow} [p] \mid \mathrm{NIL}$

since $a.b.\mathrm{NIL} \stackrel{a}{\mapsto} b.\mathrm{NIL}$ and $(c.\mathrm{NIL} + \bar{a}.\mathrm{NIL}) \stackrel{\bar{a}}{\mapsto} \mathrm{NIL}$

therefore $a.b.\mathrm{NIL} \mid (c.\mathrm{NIL} + \bar{a}.\mathrm{NIL}) \stackrel{\tau}{\mapsto} b.\mathrm{NIL} \mid \mathrm{NIL}$

and also $b.\mathrm{NIL} \mid \mathrm{NIL} \xrightarrow{b} [p] \mid \mathrm{NIL}$ ∎

When an action b is requested at a locality and the component satisfies that request, it could cause a modification of a different component, which in turn could set off a chain of changes of other components.

This means now there are concepts of $\stackrel{a}{\Rightarrow}$ and $\stackrel{\epsilon}{\Rightarrow}$ which include communication and therefore weak d-bisimulation can be redefined (using the second option to correct the definition). The same notation is used as previously.

Definition 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) $p \stackrel{\epsilon}{\Rightarrow} p'$ implies $q \stackrel{\epsilon}{\Rightarrow} q'$ for some q' such that $(p', q') \in R$,

ii) $p \stackrel{a}{\Rightarrow} C[p']$ implies $q \stackrel{a}{\Rightarrow} D[q']$ for some q' such that $(p', q') \in R$ and $(\mathrm{C}(p'), \mathrm{D}(q')) \in R$

This can be expressed as $p \approx_{\mathrm{d}} q$ if $(p,q) \in R$, for some weak d-bisimulation R ▮

As previously $\approx_{\mathrm{d}}$ is not preserved by +, and the behavioural equivalence is taken as the closure of $\approx_{\mathrm{d}}$ with respects to contexts, $\approx_{\mathrm{d}}^{\mathrm{c}}$

Definition

$p \approx_{\mathrm{d}}^{\mathrm{c}} q$ if for every context $C[\,]$: $C(p) \approx_{\mathrm{d}} C(q)$

This defines the equivalence to be the largest Σ_2 congruence generated by $\approx_{\mathrm{d}}$. ▮

$\approx_{\mathrm{d}}^{\mathrm{c}}$ is difficult to manipulate,and a similar proposition is required.

Proposition $p \approx_{\mathrm{d}}^{\mathrm{c}} q$ iff $a + p \approx_{\mathrm{d}} a + q$ for some a not in p,q. ▮

TOWARDS AN ALGEBRAIC CHARACTERISATION OF WEAK DISTRIBUTED BISIMULATION WITH COMMUNICATION

Processes cannot communicate across $\upharpoonleft$ and therefore (LP1) no longer holds.

(LP1) $p \mid q \;=\; p \upharpoonleft q \;+\; q \upharpoonleft p$

A new operator $|_{\mathrm{c}}$ is introduced which enforces communication between components. The operational semantics of $|_{\mathrm{c}}$ are as follows. $P \;|_{\mathrm{c}}\, Q$ can only perform τ moves.

(F6) Rule F6. $P \stackrel{a}{\mapsto} P' \quad Q \stackrel{\bar{a}}{\mapsto} Q'$ implies $P \;|_{\mathrm{c}}\, Q \stackrel{\tau}{\mapsto} P' \mid Q'$

(LP1) is replaced by:

(LP1') $x \mid y = x \lceil y + x \lceil y + x \mid_c y$

$\mid_c$ can be axiomatised as follows.

Communication Axioms for $\approx_d^c$

(LP1') $x \mid y = x \lceil y + x \lceil y + x \mid_c y$

(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) $(\mu.x \lceil x') \mid_c (\nu.y \lceil y') = \tau.(x \mid y) \lceil (x' \mid y')$ if $\mu = \bar{\nu}$

$= \mathrm{NIL}$ otherwise

(CP5) $a.(x \mid x') \lceil (y \mid y') + a.(c.x \lceil x' + v) \lceil (\bar{c}.y \lceil y' + w) =$
$a.(c.x \lceil x' + v) \lceil (\bar{c}.y \lceil y' + w)$

There is no proof presented for completeness, as the simplification lemma has not been generalised. However the authors refer to a complete axiomatization for a slightly different formulation of the semantics in [4].

CASTELLANI AND HENNESSY'S CONCLUSIONS

They have shown a new 'noninterleaving' semantics for simple CCS-like languages excluding communication which can be completely axiomatised. The authors note that major omission is treatment of hiding or restriction of channels. and that much work must be done before this extension be realised. They intend to pursue this line of research.

The authors also list other work that has been done in obtaining noninterleaving semantics for CCS.

i) Petri nets [9]

ii) Event structures [1] [4] [13]

The authors comment that the problems with first two approaches is that they give very concrete models of concurrent computation.

iii) Degano, De Nicola and Montanari [6] follow a similar approach to the authors, but use it build partially ordered computations for processes. In a later work [9] published after Castellani and Hennessy's paper they describe the calculus in terms of partial orderings, where CCS agents are decomposed into sets of sequential subagents.

iv) Labelled partial orders [1] have been used for concurrent languages. A new equivalence, pomset (partially ordered multiset) bisimulation has been presented with an axiomatisation. Pomset bisimulation is coarser then distributed bisimulation.

WHY DISTRIBUTED BISIMULATIONS?

In this section, I would like to investigate some questions that relate to Castellani and Hennessy's argument for distributed bisimulation. I will not try to answer this question as it is beyond the scope of this paper.

In their introduction, the authors state they wish to preserve more of the structure of processes, as compared to more traditional approaches to bisimulation that allow interleaving semantics which ignores the concurrent structure of systems. The d-lts allows causality to be distinguished by retaining the local information of a process when a transition occurs and this d-lts can be used to define a distributed bisimulation equivalence that is more discriminating than an ordinary bisimulation.

The questions that these statements give rise to are as follows:

Why does causality need to be captured?
What structure of processes is to be retained?
Why is an interleaving semantics not sufficient?
Do distributed bisimulations capture the desired structure?

After surveying the literature, it seems that this is ill-defined area. There are two general approaches to describing concurrent systems, interleaving or partial ordering which is often referred to as 'true concurrency'. First I will discuss points relating to interleaving, secondly those relating to partial orders and other noninterleaving approaches, and thirdly general points about specifying concurrent systems.

Degano et al [8] claim that the adherents of interleaving use interleaving to express concurrency by allowing events to occur in any order. They then go on to state that this imposes 'a total ordering among spatially separated and causally independent events'; this approach assumes a global clock and global state. They also state that the reasons that this approach is advocated is because of the simplicity of the underlying mathematics.

Degano et al [7] in a different paper, note that temporal/causal dependency plays a crucial role in defining certain properties and it may be awkward to define starvation/fairness in interleaving languages. This is also noted in Carchiolo et al [3], where limitations of LOTOS (a specification language based on CCS) are discussed. One of these is the fact that CCS is not suited to the treatment of fairness. Brookes et al

[2] state that ‘it seems impossible to define a notion of fairness such that a ‘fair’ process can be distinguished from an ‘unfair’ one by any finite observation. This point is also mentioned by Lamport and Lynch [10] when describing CCS.

Lamport and Lynch [10] describe the Alternating Bit Protocol as an example of an distributed algorithm. This algorithm been has specified using pure (non-distributed) CCS by Milner [11].

Pratt [12] gives a concrete situation as an argument against interleaving. He considers two ships, one in the Pacific and one in the Indian Ocean. Events on the buses of the computers on the ships take place in nanoseconds whereas communication between the ships via satellite takes a second or more. He goes on to state that ‘the idea that the totality of the two events can have a well defined linear ordering can have no practical status beyond that of a convenient mathematical fiction. Our position is that it is neither convenient nor mathematically useful. It is just as convenient, and more useful to work with partial orders.’ Some areas where Pratt claims interleaving gives problems are where there are nonatomic events or events that are a set of moments.

Degano et al [8] claim that partial orderings express concurrency as absence of ordering and no global clock is imposed. The behaviour of a system is expressed in terms of the causal relationships between subparts. In [7] Degano et al state that partial orderings explicitly specify the temporal and causal relations between events.

In terms of general comments about concurrent systems, Lamport and Lynch [10] note that in sequential computing the theory rests on concepts that are independent of any model and ‘if there is any such fundamental formal concepts underlying distributed computing, they have yet to be discovered’. They also point out that often it is argued that if there is no global state in a system, then one should not use a global state in reasoning about the model. However Lamport and Lynch argue that one can reason using an arbitrary chosen global state even though there is no unique definition of global state. Degano et al [7] point out that the issue of interleaved or partial orders cannot be resolved until there is a precise definition of ‘what and how to observe’ of the computations of a model.

This above discussion reveals that there is much conflict over what is required when modeling concurrent systems. A major problem with CCS (apart from the fact that it models concurrency with nondeterministic sequentiality) is that it cannot model fairness properties. This however is not solved by the introduction of distributed transitions and bisimulations as the observations are still finite. A problem with most of the literature on this topic is that no examples are given of concrete problems that require solution by retaining the temporal/causal dependencies, although theoretically this is an interesting problem.

CONCLUSION

The authors have presented an interesting extension to CCS to take into account the distributed nature of processes. As yet, the axiomatisation is only developed for simple CCS-like languages without communication. A simple language with communication has been developed, but no axiomatisation has been discovered for it. I found a few problems with the paper that I attempted to resolve. These occurred in the definition of weak d-bisimulation. I was also unable to give an example of where it would be preferable to use this concept instead of ‘pure’ CCS. However since the semantics were only defined for simple languages, this cannot be a major criticism.

One aspect of this that I find problematic, is that the representation of communication has become less clear than presented in original CCS. The authors have now created three different types of transition i) observable local action, ii) global τ action (that could be localised if the τ action was pre-emptive) and iii) global communication actions a and $\bar{a}$ which combine to form a global τ action. The fact that restriction and relabelling are missing from the description perhaps hinders understanding of how these different types of transitions interact.

If this theory does become further developed, there will need to be trade-off between its power to describe distributed systems and the fact that certain issues have become more complex.

As described in the last section it is difficult to define what is required for a language for specifying concurrent systems and this is an area that requires better understanding, before a concept such as distributed bisimulations can be fully evaluated.

REFERENCES

1. BOUDOL, G. AND CASTELLANI, I. Concurrency and atomicity. *Theoretical Computer Science* 59 (1988), 25-64.
2. BROOKES, S.D., HOARE, C.A.R. AND ROSCOE, A.W. A theory of communicating sequential processes. *Journal of the ACM*, 31(3), July 1984, 560-599.
3. CARCHIOLO, V., FARO, A., MIRABELLA, O., PAPPALARDO, G. AND SCOLLO, G. A LOTOS Specification of the PROWAY Highway Service. *IEEE Transactions on Computer* C-35(11) November 1986, 949-966.
4. CASTELLANI, I. Bisimulations for concurrency. PhD Dissertation, University of Edinburgh, Edinburgh, Great Britain, 1988.
5. CASTELLANI, I. AND HENNESSY, M. Distributed bisimulations. *Journal of the ACM* 36(4) October 1989, 887-911.
6. 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, 520-533.
7. DEGANO, P. AND MONTANARI, U. Concurrent histories: A basis for observing distributed systems, *Journal of Computer and System Sciences* 34, 1987, 422-461.
8. DEGANO, P., DE NICOLA, R. AND MONTANARI, U. A partial ordering semantics for CCS. *Theoretical Computer Science* 75 1990, 223-262.
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, 196-208.
10. LAMPORT, L. AND LYNCH, N., Distributed computing: models and methods. In *Handbook of Theoretical Computer Science Vol II*, Ed J. van Leeuwen, Elsevier, 1990.
11. MILNER, R. *Communication and Concurrency*. Prentice Hall, United Kingdom, 1989.
12. PRATT, V. Modelling concurrency with partial orders, *International Journal of Parallel Programming* 15(1) 1986, 33-71.
13. 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. 104, Springer-Verlag, New York, 1987, 224-242.