Theoretical Computer Science 87 (1991) 97-114
Elsevier

The equivalence in the DCP model*

Reine Fournier

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, C.P. 6128, Succ. A, Montréal, Que., Canada H3C 3J7, and Département de Mathématiques et d'Informatique, Université de Sherbrooke, Sherbrooke, Que., Canada J1K 2R1

Gregor von Bochmann

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, C.P. 6128, Succ. A, Montréal, P.Q., Canada H3C 3J7

Communicated by M. Nivat
Received December 1988
Revised April 1989

Abstract

Fournier, R. and G. von Bochmann, The equivalence in the DCP model, Theoretical Computer Science 87 (1991) 97-114.

The ever increasing complexity of systems stimulates research in the area of processes equivalences. In this paper, processes are considered as black boxes, characterized by their external interactions only, and the equivalences are based on this assumption. The equivalence relation induced from the partial order defined in Johnston's model of Discrete Communicating Processes is studied with the intention of finding its place within the chain of existing equivalences (namely, trace equivalence, testing equivalence, bisimulation and observational equivalence). Unfortunately, this model does not compare easily with the others. However a modification to the original model, consisting in keeping more information within a process identifier, namely all of its immediately performable events, and explicitly writing deadlocks, gives a new equivalence relation $=_{\sigma\text{-}J}$ which is finer than the original one and which has the property of being equivalent to bisimulation.

1. Introduction

Since the beginning of the eighties, several algebraic theories of processes have appeared such as CCS [18], CSP [13], SCCS [19], ACP [1], CIRCAL [17], DCP [6, 23], LOTOS [14], dCCS [15]. Within each theory an equivalence relation is

* This work was partially supported by the Natural Sciences and Engineering Research Council of Canada.

defined to help determine whether two processes are equivalent or not. The theory of equivalences is very useful since it allows us to replace a complex system (or parts of it) by a simpler equivalent one facilitating the analysis of the entire system and the verification of its properties.

This paper deals with the equivalence relation induced from the partial order defined in the DCP model. Johnston's model DCP (Discrete Communicating Processes) [6, 23] permits the formal specification and meaningful analysis of the behaviour of distributed computing systems. Furthermore, it incorporates computational tools to aid analysis and verification [22] which might make it even more appealing. In fact, we shall try to find its place within some of the existing equivalences. Since it proves impossible to realize our goal with the original definition, we shall show how a modification to the original model helps us to fit this new equivalence relation in the chain of existing equivalences. We shall also prove that this new equivalence is finer than the original one.

All equivalences considered in this paper are based on the idea that two systems are equivalent if they cannot be distinguished by (external) observation. However, different forms of observation are considered. We use the term *process* to represent an abstract entity able to perform internal (invisible) actions as well as to communicate with other processes in its environment via communcation events (interactions).

This paper is organized as follows: Section 2 gives a brief overview of the model DCP [6, 23]; Section 3 introduces some equivalences on labelled transitions systems and reminds the reader of the relations between them [7]; Section 4 shows how the equivalence relation induced from the partial order in DCP relates with the above mentioned equivalences; Section 5 shows how the introduction of explicit deadlock in DCP pushes the DCP equivalence into the chain of equivalences described in Section 3; and Section 6 gives a short conclusion.

2. Discrete communication processes

In order to compare the equivalence defined in Johnston's model of Discrete Communication Processes (DCP) [6, 23] with other equivalences (such as observational equivalence [18], bisimulation [21], trace equivalence [12] and testing equivalence [3, 7, 8]), we shall adapt Johnston's equivalence to Labelled Transition Systems (LTS).

Notice that since their introduction by Keller [16], transition systems have been used as an underlying model for many proposed models of parallelism. In fact, transition systems are a relational model based on two primitive notions: state and transition. Since it is possible, for the DCP model, to define the notion of global state and a notion of indivisible action causing a state transition, we can define for each DCP process a corresponding transition system. This correspondence determines an interleaving semantics for the model.

In this paper, we shall consider (following De Nicola [7]) a particular class of nondeterministic transition systems which can be used to model systems controllable through interactions with a surrounding environment, but also capable of performing internal actions τ which cannot be influenced or even seen by any external agent.

Definition 2.1. A *labelled transition system* (*LTS*) is a quadruple $\langle S, A, T, s_0\rangle$ where
(i) S is a countable nonempty set of *states*;
(ii) A is a countable set of *elementary actions*;
(iii) T is a function from $A\cup\{\tau\}$ into $\mathcal{P}(S\times S)$ where $T(\mu)$ is called a *transition relation* and denoted by $\xrightarrow{\mu}$ for each $\mu\in A\cup\{\tau\}$;
(iv) $s_0\in S$ is the *initial state* of the labelled transition system.

In this definition, each binary relation $\xrightarrow{\mu}$ shows the effect of the elementary action μ on the system. In fact, if $q, q'\in S$ and $\langle q, q'\rangle\in\xrightarrow{\mu}$ (denoted $q\xrightarrow{\mu}q'$) this means that if the system is in state q, the execution of μ will bring the system into state q'; $q\xrightarrow{\tau}q'$ indicates that the system while in state q can perform a silent move to state q'.

Such a transition system can obviously be unrolled into a tree whose nodes are the states, the root being the initial state, and whose arcs are labelled with elements of $A\cup\{\tau\}$.

Definition 2.2. Two labelled transition systems with the same set of elementary actions, $\mathrm{LTS}_1=\langle S_1, A, T_1, s_{01}\rangle$ and $\mathrm{LTS}_2=\langle S_2, A, T_2, s_{02}\rangle$ are said to be *isomorphic* if there exists a one-to-one correspondence $f: S_1\rightarrow S_2$ such that
(i) $s_{11}\xrightarrow{a}s_{12}$ if and only if $f(s_{11})\xrightarrow{a}f(s_{12})$ for all $a\in A$ and all $s_{11}, s_{12}\in S_1$;
(ii) $f(s_{01})=s_{02}$.

In this paper, we shall use the following notations:

- A denotes the set of visible actions: $a, b, c, \ldots$ called elementary actions in the above definition;
- A^* denotes the set of strings of elements of A whose elements are $s, s', \ldots$ and ε, the empty chain;
- τ denotes the invisible (internal) action (defined earlier);
- $A_\tau = A\cup\{\tau\}$ whose elements are $\mu_1, \mu_2, \ldots$;
- $p\xrightarrow{\mu_1\mu_2\cdots\mu_n}q$ is the abbreviation of $\exists p_0, \ldots, p_n$ such that
$$p_0=p\xrightarrow{\mu_1}p_1\xrightarrow{\mu_2}\ldots\rightarrow p_{n-1}\xrightarrow{\mu_n}p_n=q;$$
- $p\xrightarrow{\mu_1\mu_2\cdots\mu_n}$ means that there exists a q such that $p\xrightarrow{\mu_1\mu_2\cdots\mu_n}q$;
- $p\overset{\varepsilon}{\Rightarrow}q$ means that there exists an $n\geqslant 0$ such that $p\xrightarrow{\tau^n}q$;
- $p\overset{\mu}{\Rightarrow}q$ means that there exist p_1 and p_2 such that $p\overset{\varepsilon}{\Rightarrow}p_1\xrightarrow{\mu}p_2\overset{\varepsilon}{\Rightarrow}q$;
- $p\xRightarrow{a_1a_2\cdots a_n}q$ means that there exist $p_0, \ldots, p_n$ such that
$$p=p_0\overset{a_1}{\Rightarrow}p_1\overset{a_2}{\Rightarrow}\cdots\xRightarrow{a_{n-1}}p_{n-1}\overset{a_n}{\Rightarrow}p_n=q;$$
- $p\overset{s}{\Rightarrow}$ means that there exists q such that $p\overset{s}{\Rightarrow}q$.

Now let us go back to the DCP model [6, 23]. As in Milner's CCS [18], the DCP model uses the external behaviour to define processes which are described by the interactions that they exchange with their environment, as follows.

Definition 2.3 (Johnston [6]; Rea and Johnston [23]). A *process* $\boldsymbol{p}$ can be defined as a set $\{\langle e_1, \boldsymbol{q}_1\rangle, \ldots, \langle e_n, \boldsymbol{q}_n\rangle\}$ of pairs where each e_i is a communication event and each $\boldsymbol{q}_i$ is a subsequent process or behaviour.

This should be interpreted as follows: the process $\boldsymbol{p}$ offers, for all i, to exchange communication e_i with its environment; if it is accepted then process $\boldsymbol{p}$ will proceed as process $\boldsymbol{q}_i$. This definition is inherently recursive, a process being defined in terms of processes. The behaviour of a discrete communication system is characterized by the pattern (usually infinite) of its exchanges with the environment; it is this behaviour which is called a discrete communication process. These processes can be represented by infinite trees whose branches are labelled with communication events and whose nodes represent the initial process (root) and its successors.

At any given time a process may emit a message or absorb one. The emission of message $\boldsymbol{a}$ will be denoted by $\boldsymbol{a}!$, while the absorption of message $\boldsymbol{a}$ will be denoted by $\boldsymbol{a}?$.

Remark 2.4. A DCP process $\boldsymbol{p}$ can be viewed as the following labelled transition system $\langle S, A, T, \boldsymbol{p}\rangle$, where

(i) S is the set containing $\boldsymbol{p}$ and all its successors;

(ii) A is the set of all communication events used in the definition of $\boldsymbol{p}$ or one of its successors;

(iii) T is a set of transition relations whose elements are binary relations on S denoted by $\xrightarrow{\mu}$ for each $\mu \in A \cup \{\tau\}$, defined as follows: if $\boldsymbol{p}'$ is $\boldsymbol{p}$ or one of its successors and $\langle \mu, \boldsymbol{q}\rangle \in \boldsymbol{p}'$ then we have that $(\boldsymbol{p}', \boldsymbol{q}) \in \xrightarrow{\mu}$ which we write $\boldsymbol{p}' \xrightarrow{\mu} \boldsymbol{q}$.

Example 2.5. If $\boldsymbol{P}_3 = \{\langle b, \mathbf{stop}\rangle\}$ and $\boldsymbol{P}_4 = \{\langle c, \mathbf{stop}\rangle\}$. Then we define $\boldsymbol{P}_1 = \{\langle a, \boldsymbol{P}_3\rangle, \langle a, \boldsymbol{P}_4\rangle\}$ which is represented by the tree in Fig. 1.

Furthermore if $\boldsymbol{P} + \boldsymbol{Q}$ denotes the process that behaves either like $\boldsymbol{P}$ or $\boldsymbol{Q}$ depending on the first offer made by the environment, then $\boldsymbol{P}_2 = \{\langle a, \boldsymbol{P}_3 + \boldsymbol{P}_4\rangle\}$ is represented by the tree in Fig. 2.

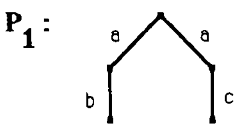


Fig. 1.

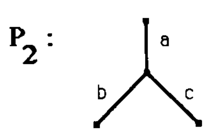


Fig. 2.

Intuitively we would like to say that two processes $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are "related" [6] if, for instance, whatever communication event $\boldsymbol{P}_2$ can offer, $\boldsymbol{P}_1$ can offer it too. Then we shall say that $\boldsymbol{P}_1$ *simulates* $\boldsymbol{P}_2$ [6, 23]. Formally, we have the following definition.

Definition 2.6 (Johnston [6]; Rea, Johnston [23]). Let $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ be two processes. We say that $\boldsymbol{P}_1$ *simulates* $\boldsymbol{P}_2$ (denoted $\boldsymbol{P}_1 \leqslant_{\mathrm{J}} \boldsymbol{P}_2$) if and only if for each $\langle e_2, \boldsymbol{q}_2\rangle \in \boldsymbol{P}_2$ there exists $\langle e_1, \boldsymbol{q}_1\rangle \in \boldsymbol{P}_1$ such that $e_1 = e_2$ and $\boldsymbol{q}_1 \leqslant_{\mathrm{J}} \boldsymbol{q}_2$. Furthermore, we say that $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are ***J-equivalent***, as Johnston-equivalent, (denoted $\boldsymbol{P}_1 =_{\mathrm{J}} \boldsymbol{P}_2$) if and only if $\boldsymbol{P}_1 \leqslant_{\mathrm{J}} \boldsymbol{P}_2$ and $\boldsymbol{P}_2 \leqslant_{\mathrm{J}} \boldsymbol{P}_1$.

Remark 2.7. The processes $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ defined in Example 2.5 are not **J**-equivalent.

Remark 2.8. The partial order $\leqslant_J$ corresponds to the Smyth ordering [25]. Obviously, $=_{\mathrm{J}}$ is an equivalence relation. In fact, two processes are **J**-equivalent if they have the same minimal behaviour. For example,

$$P_1 = \{\langle s, \emptyset\rangle, \langle s, \{\langle y, \emptyset\rangle\}\rangle, \langle d, \{\langle e, \emptyset\rangle\}\rangle\}$$

and

$$P_2 = \{\langle s, \{\langle y, \emptyset\rangle\}\rangle, \langle d, \{\langle e, \emptyset\rangle\}\rangle\}$$

are **J**-equivalent.

We can reformulate the above definition in terms of transition systems. We would get the following definition.

Definition 2.9. Let $\mathbf{LTS}_1 = \langle S_1, A, T_1, s_{01}\rangle$ and $\mathbf{LTS}_2 = \langle S_2, A, T_2, s_{02}\rangle$ be two labelled transition systems with the same set of actions. For $i = 1, 2$, let $\mathbf{LTS}_i(s_i)$ denote the subsystem of $\mathbf{LTS}_i$ which has s_i as its initial state, that is, the subtree of $\mathbf{LTS}_i$ which has s_i as its root. $\mathbf{LTS}_1$ *simulates* $\mathbf{LTS}_2$ (denoted $\mathbf{LTS}_1 \leqslant_{\mathrm{J}} \mathbf{LTS}_2$) if and only if for all $t \in A$ and for all $s_2 \in S_2$ such that $s_{02} \stackrel{t}{\Rightarrow} s_2$, there exists an $s_1 \in S_1$ such that $s_{01} \stackrel{t}{\Rightarrow} s_1$ and $\mathbf{LTS}_1(s_1) \leqslant_{\mathrm{J}} \mathbf{LTS}_2(s_2)$.

3. Overview of other equivalences

In this section, we shall first briefly recall the definitions of some equivalences and show how they are related. The interested reader is referred to De Nicola [7] for a more extensive analysis. In his paper [7], De Nicola considers only processes that can be represented by finitely branching trees. We shall also make this assumption since those DCP processes of practical interest can always be represented by such trees.

3.1. Trace equivalence

A natural approach to system equivalence is considering two systems as equivalent that can perform exactly the same sequences of visible actions (not considering any internal actions) [12].

Definition 3.1.1 (De Nicola [7], Hoare [11]). Let $\mathbf{TS_1} = \langle P, A, T_1, p_0\rangle$ and $\mathbf{TS_2} = \langle Q, A, T_2, q_0\rangle$ be two transition systems. Then we say that $\mathbf{TS_1}$ is *trace equivalent to* $\mathbf{TS_2}$ (denoted $\mathbf{TS_1} \sim_t \mathbf{TS_2}$) if and only if

$$(\forall s \in A^*)\ (p_0 \stackrel{s}{\Rightarrow} \text{if and only if } q_0 \stackrel{s}{\Rightarrow}).$$

Let us define $\mathbf{Traces}(q) = \{s \in A^* \mid q \stackrel{s}{\Rightarrow}\}$ to be set of all possible traces of process q.

Remark 3.1.2. $\mathbf{TS_1} \sim_t \mathbf{TS_2}$ if and only if $\mathbf{Traces}(p_0) = \mathbf{Traces}(q_0)$; furthermore, $\sim_t$ is obviously an equivalence relation.

This equivalence is sometimes called strings equivalence [7].

It can be easily seen that the two processes of Example 2.5 are trace equivalent since $\mathbf{Traces}(\boldsymbol{P_1}) = \{a, ab, ac\} = \mathbf{Traces}(\boldsymbol{P_2})$. However, if $\boldsymbol{P_3}$ is the process shown in Fig. 3, then obviously $\boldsymbol{P_2}$ and $\boldsymbol{P_3}$ are not trace equivalent.

This equivalence is used in automata and language theories; it is also the basis of many semantics proposed for Hoare's CSP [5, 11, 12, 13].

Fig. 3.

3.2. Observational equivalence and bisimulation

Milner defines three equivalences in his CCS model. Two of them are of no interest since they are much too strong to be of any use. Therefore, we shall only consider the observational equivalence which permits the absorption of internal actions.

Definition 3.2.1 (De Nicola [7], Milner [18]). Let $\mathbf{ST} = \langle P, A, T, p_0 \rangle$ be a labelled transition system. Let $p, q \in P$; then

(i) $p \approx_0 q$ is always true,

(ii) $p \approx_{k+1} q$ if and only if, for all $s \in A^*$,

(1) for all p' in P such that $p \xrightarrow{s} p'$, there exists a q' in P such that

$$q \xrightarrow{s} q' \quad \text{and} \quad p' \approx_k q'$$

(2) for all q' in P such that $q \xrightarrow{s} q'$, there exists a p' in P such that

$$p \xrightarrow{s} p' \quad \text{and} \quad p' \approx_k q'.$$

(iii) $p \approx q$ if and only if $p \approx_k q$ for all $k \geqslant 0$; then we say that p is *observationally equivalent to q*.

There is a *natural* extension (given in the next definition) from this definition of observational equivalence between two states of a same labelled transition system to a definition of observational equivalence between two different labelled transition systems [7].

Definition 3.2.2. Let $\mathbf{ST_1} = \langle S_1, A, T_1, p_0 \rangle$ and $\mathbf{ST_2} = \langle S_2, A, T_2, q_0 \rangle$ be two distinct labelled transition systems such that $S_1 \cap S_2 = \emptyset$. If $\mathbf{ST}$, defined as follows.

$$\mathbf{ST} = \langle S_1 \cup S_2 \cup \{s_0\},\ A,\ T_1 \cup T_2,\ s_0 \rangle$$

is the labelled transition system obtained as the result of the union of $\mathbf{ST_1}$ and $\mathbf{ST_2}$, then $\mathbf{ST_1} \approx \mathbf{ST_2}$ if and only if $p_0 \approx q_0$ in $\mathbf{ST}$.

Starting from the notion of weak homomorphism in automata theory, Park [21] proposed in 1981 a new way of defining the observational equivalence (called *bisimulation*). Using this approach, we would say that two states, p and q, are equivalent (denoted $p \approx_{\text{bis}} q$) if and only if there exists a relation $\mathfrak{R}$ (called bisimulation) containing the pair $\langle p, q \rangle$ and guaranteeing that p and q can accomplish the same sequences of visible actions always ending in equivalent states of $\mathfrak{R}$. Formally, we get the following definition.

Definition 3.2.3 (De Nicola [7], Park [21]). Let $\mathbf{ST_1} = \langle S_1, A, T_1, p_0 \rangle$ and $\mathbf{ST_2} = \langle S_2, A, T_2, q_0 \rangle$ be two distinct labelled transition systems such that $S_1 \cap S_2 = \emptyset$. If $\mathfrak{R}$ is a relation between states of two systems, i.e. $\mathfrak{R} \subset S_1 \times S_2$, let us define $\boldsymbol{F}$ by

$$\begin{aligned}\boldsymbol{F}(\mathfrak{R}) = \{\langle p, q \rangle \,|\, \forall s \in A^* \ &\text{(i) if } p \stackrel{s}{\Rightarrow} p' \text{ then } (\exists q')\ (q \stackrel{s}{\Rightarrow} q') \\ &\qquad\qquad\qquad \text{and } \langle p', q' \rangle \in \mathfrak{R} \\ &\text{(ii) if } q \stackrel{s}{\Rightarrow} q' \text{ then } (\exists p')\ (p \stackrel{s}{\Rightarrow} p') \\ &\qquad\qquad\qquad \text{and } \langle p', q' \rangle \in \mathfrak{R}\}.\end{aligned}$$

A relation $\mathfrak{R}$ is a bisimulation if $\mathfrak{R} \subseteq \boldsymbol{F}(\mathfrak{R})$. The relation $\approx_{\mathbf{bis}}$ defined by

$$\approx_{\mathbf{bis}} = \bigcup_{\mathfrak{R} \subseteq \boldsymbol{F}(\mathfrak{R})} \mathfrak{R}$$

is called *observation equivalence.*

Since F is a monotonic function on the lattice of relations ordered by inclusion, the equivalence $\approx_{\mathbf{bis}}$ is obtained by taking the minimal fixed point of F [21].

Definition 3.2.4. Let $\mathbf{ST}_1 = \langle S_1, A, T_1, p_0\rangle$ and $\mathbf{ST}_2 = \langle S_2, A, T_2, q_0\rangle$ be two distinct labelled transition systems. We say that $\mathbf{ST}_1 \approx_{\mathbf{bis}} \mathbf{ST}_2$ if there exists a bisimulation $\mathfrak{R}$ containing the pair $\langle p_0, q_0\rangle$.

Definition 3.2.5. Let R be a binary relation from A to B, we say that R is *image-finite* if and only if for each $a \in A$ the set $R_a = \{y \mid (a, y) \in R\}$ is finite.

The two definitions, Definitions 3.2.2 and 3.2.4, are well studied in [10] and [24]. It is shown that if the relation $\stackrel{s}{\Rightarrow}$ is an image-finite relation, then $\approx$ and $\approx_{\mathbf{bis}}$ coincide; however, if the relation $\stackrel{s}{\Rightarrow}$ is not image-finite, then we can only obtain that $\mathbf{ST}_1 \approx_{\mathbf{bis}} \mathbf{ST}_2$ implies $\mathbf{ST}_1 \approx \mathbf{ST}_2$ [24].

Example 3.2.6. The processes P_1 and P_2 defined in Example 2.5 are not observational equivalent. However, the two processes in Fig. 4 are obviously observational equivalent.

3.3. Testing equivalence

We might take yet another approach to the problem of finding whether or not two processes are equivalent. In fact, the external behaviour of a process can be tested by means of a sequence of tests [20]. When considering nondeterministic processes, not only do we want to know if they pass or not a specific test but also if they will always behave the same way.

In this formulation, we shall consider a set of processes and a set of tests. We shall say that two processes are equivalent (with respect to this set of tests) if they pass exactly the same tests. This equivalence can be split into two preorder relations one of which is formulated in terms of the ability to answer positively to a test, and the other, in terms of the impossibility not to answer positively to a test.

Before defining formally what we mean by testing equivalence, we shall give some useful definitions.

Definition 3.3.1. For any $s \in A^*$, we define

$$p \textbf{ after } s = \{p' \mid p \stackrel{s}{\Rightarrow} p'\}$$

$$P \textbf{ after } s = \bigcup_{p \in P} (p \textbf{ after } s).$$

Fig. 4.

For any finite subset L of A, we define

$$p\ \mathbf{MUST}\ L \text{ if and only if } (p \Rightarrow p' \text{ implies that } \exists a \in L \text{ such that } p' \overset{a}{\Rightarrow})$$

and

$$P\ \mathbf{MUST}\ L \text{ if and only if } p\ \mathbf{MUST}\ L \text{ for all } p \in P.$$

Let B be a closed LOTOS behaviour expression (i.e. without free variables), the labelled transition system associated with B, is Sys $= \langle S, A, T, s_0\rangle$, where S is the set of all behaviour expressions that could be derived starting with B, A denotes the set of all visible actions, T denotes the set of transition relations starting at B or one of its successors and $s_0 = B$.

Definition 3.3.2 (De Nicola [7], ISO [14]). Let $\mathrm{Sys}_1 = \langle S_1, A_1, T_1, s_{01}\rangle$ and $\mathrm{Sys}_2 = \langle S_2, A_2, T_2, s_{02}\rangle$. These systems could be extended to a set of common labels: $A = A_1 \cup A_2$. We define the predicate (Sys_1 **red** Sys_2) by

$$(\mathrm{Sys}_1\ \mathbf{red}\ \mathrm{Sys}_2) \quad \text{if and only if}$$

$$(\forall t \in A^*)(\forall L \subseteq A)\,[(s_{02}\ \mathbf{after}\ t)\ \mathbf{MUST}\ L \text{ implies } (s_{01}\ \mathbf{after}\ t)\ \mathbf{MUST}\ L)].$$

If B_1 and B_2 are two behaviour expressions, we say that B_1 reduces B_2 (denoted B_1 **red** B_2) (see 2, 3) if and only if, for their respective transition system Sys_1, Sys_2, we have Sys_1 **red** Sys_2.

Definition 3.3.3. Two LOTOS behaviour expressions B_1 and B_2 are *testing equivalent* (denoted B_1 **te** B_2) if and only if B_1 **red** B_2 and B_2 **red** B_1.

Using this equivalence, we can identify processes that are not distinguishable by external experiences but would not be observationally equivalent.

Example 3.3.4. The two processes of Fig. 5 are not testing equivalent. However, the two processes of Fig. 6 are testing equivalent.

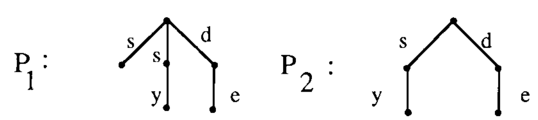


Fig. 5.

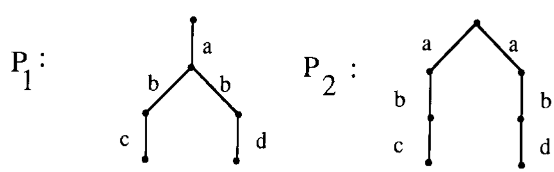


Fig. 6.

3.4. Relationship between different equivalence relations

According to De Nicola [7], the diagram of Fig. 7 shows how these equivalence relations are related.

$$
\begin{array}{ccc}
\approx & \Longleftrightarrow & \approx_{\text{bis}} \\
 & \Downarrow & \\
 & \approx_2 & \\
 & \Downarrow & \\
 & \text{te} & \\
 & \Downarrow & \\
\approx_1 & \Longleftrightarrow & \sim_s
\end{array}
$$

Fig. 7.

4. Johnston's equivalence

We notice that Johnston's equivalence does not appear in the diagram of Fig. 7. The goal of this section is to find a place for this equivalence within this diagram.

4.1. Johnston's equivalence and trace equivalence

First, we shall show that Johnston's equivalence is strictly finer than the trace equivalence.

Proposition 4.1.1. *There exist two processes P_1 and P_2 such that $P_1 \sim_t P_2$ and $\neg(P_1 =_J P_2)$ is true.*

Proof. Take P_1 and P_2 as in Example 2.5. To show that $P_1 \sim_t P_2$ we must show that $\mathbf{Traces}(P_1) = \mathbf{Traces}(P_2)$. But $\mathbf{Traces}(P_1) = \{a, ab, ac\}$ and $\mathbf{Traces}(P_2) = \{a, ab, ac\}$.

Now we shall establish that $\neg(P_1 =_J P_2)$ is true. One can easily see that $P_2 \leqslant_J P_1$. So let us show that $\neg(P_1 \leqslant_J P_2)$ is true. In fact, a is the only communication event of P_2 since $P_2 = \{\langle a, P_3 + P_4\rangle\}$. Consequently, we get $P_2 \xrightarrow{a} P_3 + P_4$ and $P_1 \xrightarrow{a} P_3$ and $P_1 \xrightarrow{a} P_4$. Now $P_3 + P_4 = \{\langle b, \mathbf{stop}\rangle, \langle c, \mathbf{stop}\rangle\}$ and we must show that none of two possibilities $P_3 \leqslant_J P_3 + P_4$ or $P_4 \leqslant_J P_3 + P_4$ is true. But the first one is not true since P_3 has no c communication event. Similarly, the second one is also false. Henceforth, $\neg(P_1 \leqslant_J P_2)$ is true. $\square$

Theorem 4.1.2. *Let P_1 and P_2 be two processes. If $P_1 \leqslant_J P_2$ then* $\mathbf{Traces}(P_2) \subseteq \mathbf{Traces}(P_1)$.

Proof. The proof is done using induction on the length of $s \in \mathbf{Traces}(P_2)$ (denoted $|s|$). Let $s \in \mathbf{Traces}(P_2)$ such that $|s| = 1$. We have that $s \in A$. Since $s \in \mathbf{Traces}(P_2)$, there exists a process Q_2 such that $\langle s, Q_2\rangle \in P_2$. Furthermore, from the definition of $P_1 \leqslant_J P_2$, there exists an $\langle s_1, Q_1\rangle \in P_1$ such that $s = s_1$ and $Q_1 \leqslant_J Q_2$; that is, $s \in \mathbf{Traces}(P_1)$.

Now suppose that the property is satisfied for all pairs of programs $\langle P, Q\rangle$ such that $P \leqslant_J Q$ and for all $s \in \mathbf{Traces}(Q)$ whose length is less than n. Take $s \in \mathbf{Traces}(P_2)$ such that $|s| = n$. We may write s as $a_1 a_2 \ldots a_n$ where each $a_i \in A$. Let $s' = a_2 \ldots a_n$; then $|s'| = n-1$. However, $s \in \mathbf{Traces}(P_2)$ implies that there exist $(n+1)$ processes $P_{20}, P_{21}, \ldots, P_{2n}$ such that $P_{20} = P_2$ and $\langle a_i, P_{2i}\rangle \in P_{2(i-1)}$ for every $i = 1, \ldots, n$. Since $\langle a_1, P_{21}\rangle \in P_{20} = P_2$, we get from the definition of $\leqslant_J$ that there is an $\langle a_1, P_{11}\rangle \in P_1$ such that $P_{11} \leqslant_J P_{21}$. Since $s' \in \mathbf{Traces}(P_{21})$, $|s'| = n-1 < n$ and $P_{11} \leqslant_J P_{21}$ then, by the induction hypothesis, $s' \in \mathbf{Traces}(P_{11})$. Consequently, $s \in \mathbf{Traces}(P_1)$. Hence, $\mathbf{Traces}(P_2) \subseteq \mathbf{Traces}(P_1)$. $\square$

Corollary 4.1.3. *Let P_1 and P_2 be two processes. If $P_1 =_J P_2$ then $P_1 \sim_s P_2$.*

Proof. By Theorem 4.1.2, $P_1 =_J P_2$ implies that $\mathbf{Traces}(P_2) = \mathbf{Traces}(P_1)$ which is the same as $P_1 \sim_t P_2$ by Remark 3.1.2. $\square$

4.2. Johnston's equivalence and testing equivalence

Proposition 4.2.1. *There exist two processes P_1 and P_2 such that P_1 **te** P_2 and $\neg(P_1 =_J P_2)$ is true.*

Proof. Consider the processes given in Fig. 6. By an argument similar to the one given in Proposition 4.1.1, one can easily show that $\neg(P_1 =_J P_2)$ is true. Now we must show that P_1 **te** P_2. This fact is clearly true since both P_1 and P_2 will always accept the sequences, *a* and *ab*, and will sometimes accept the sequences, *abc* or *abd*, sometimes not. $\square$

Proposition 4.2.2. *There exist two processes P_1 and P_2 such that $P_1 =_J P_2$ and $\neg(P_1$ **te** $P_2)$ is true.*

Proof. Consider the processes given in Fig. 5. First we shall prove that $P_1 =_J P_2$. To prove that, we must prove that $P_1 \leqslant_J P_2$ and $P_2 \leqslant_J P_1$.

(a) Let us show that $P_1 \leqslant_J P_2$; that is, for each $\langle e_2, \boldsymbol{q}_2\rangle \in P_2$ there exists $\langle e_1, \boldsymbol{q}_1\rangle \in P_1$ such that $e_1 = e_2$ and $\boldsymbol{q}_1 \leqslant_J \boldsymbol{q}_2$. We have two cases:

Case $e_2 = d$: Then $\boldsymbol{q}_2 = \{\langle e, \emptyset\rangle\}$. Similarly, in P_1, we get that $\boldsymbol{q}_1 = \{\langle e, \emptyset\rangle\}$. Since $\boldsymbol{q}_1 = \boldsymbol{q}_2$ we certainly have $\boldsymbol{q}_1 \leqslant_J \boldsymbol{q}_2$.

Case $e_2 = s$: Then $\boldsymbol{q}_2 = \{\langle y, \emptyset\rangle\}$. Since P_1 is given by $P_1 = \{\langle d, \{\langle e, \emptyset\rangle\}\rangle, \langle s, \emptyset\rangle, \langle s, \{\langle y, \emptyset\rangle\}\rangle\}$, there are two possible successors to P_1 following the interaction s: $\emptyset$ or $\{\langle y, \emptyset\rangle\}$. We may take $\boldsymbol{q}_1$ to be $\{\langle y, \emptyset\rangle\}$. Then $\boldsymbol{q}_1 = \boldsymbol{q}_2$ and we certainly have that $\boldsymbol{q}_1 \leqslant_J \boldsymbol{q}_2$. This finally establishes that $P_1 \leqslant_J P_2$.

(b) Now let us show that $P_2 \leqslant_J P_1$; that is, for each $\langle e_1, \boldsymbol{q}_1\rangle \in P_1$ there exists $\langle e_2, \boldsymbol{q}_2\rangle \in P_2$ such that $e_1 = e_2$ and $\boldsymbol{q}_1 \leqslant_J \boldsymbol{q}_2$. If $\langle e_1, \boldsymbol{q}_1\rangle \in \{\langle s, \{\langle y, \emptyset\rangle\}\rangle, \langle d, \{\langle e, \emptyset\rangle\}\rangle\}$ then we choose $\langle e_2, \boldsymbol{q}_2\rangle = \langle e_1, \boldsymbol{q}_1\rangle$. If $\langle e_1, \boldsymbol{q}_1\rangle = \langle s, \emptyset\rangle$ then we take $\langle e_2, \boldsymbol{q}_2\rangle = \langle s, \{\langle y, \emptyset\rangle\}\rangle$. But we clearly have that $\{\langle y, \emptyset\rangle\} \leqslant_J \emptyset$. Hence we have proven that $P_1 =_J P_2$.

Now we must prove that $\neg(P_1 \textbf{ te } P_2)$ is true. In order to do this it is sufficient to prove that either $\neg(P_1 \textbf{ red } P_2)$ is true or $\neg(P_2 \textbf{ red } P_1)$ is true. We shall prove that $\neg(P_1 \textbf{ red } P_2)$ is true. Let $L = A$ and take $s \in \textbf{Traces}(P_2)$. To prove our claim, it suffices to prove that

$$(\exists Q_1)((P_1 \stackrel{s}{\Rightarrow} Q_1) \wedge (\forall a \in A)\neg(Q_1 \stackrel{a}{\Rightarrow}))$$
$$\wedge \neg((\exists Q_2)((P_2 \stackrel{s}{\Rightarrow} Q_2) \wedge (\forall a \in A)\neg(Q_2 \stackrel{a}{\Rightarrow}))).$$

Let $Q_1 = \emptyset$ then for each $a \in L$ we have that $(\neg(Q_1 \stackrel{a}{\Rightarrow})$ is true. Since

$$P_2 = \{\langle d, \{\langle e, \emptyset\rangle\}\rangle, \langle s, \{\langle y, \emptyset\rangle\}\rangle\},$$

the only possible successor of P_2 after an s interaction is $Q_2 = \{\langle y, \emptyset\rangle\}$. But $Q_2 \stackrel{y}{\Rightarrow}$ and $y \in L$! Hence $\neg(P_1 \textbf{ red } P_2)$ is true. $\square$

Hence, in general, there is no relation between **te** and $=_J$.

4.3. *Johnston's equivalence and observational equivalence*

As a consequence of results illustrated in Fig. 7 and results of Section 4.2, we know that $(P_1 =_J P_2)$ does not imply that $(P_1 \approx_{\text{bis}} P_2)$. Otherwise, since $(P_1 \approx_{\text{bis}} P_2)$ implies that $(P_1 \textbf{ te } P_2)$ (see Fig. 7 and De Nicola [7]), $(P_1 =_J P_2)$ would imply that $(P_1 \textbf{ te } P_2)$ which is contradicted by Proposition 4.2.2. Similarly, we can prove that $(P_1 =_J P_2)$ does not imply that $(P_1 \approx_2 P_2)$.

The example used in Proposition 4.2.2 indicates that, if anything, $\approx_{\text{bis}}$ and $\approx$ are finer than $=_J$.

In his Ph.D. Thesis, Sanderson [24, Chapter 5], gives some results concerning the bisimulation as defined by Park [21]. Within this context, the equivalence is obtained as the maximal fixed point of the relation used to define $\approx_{k+1}$ starting with $\approx_k$ using the partial order induced by the set inclusion. It has been shown by Tarski [26] that such a maximal fixed point always exist under these conditions.

Sanderson shows that bisimulation is stronger than observational equivalence [24, Proposition 5.3]. Furthermore, a simpler relation than the one used to obtain $\approx_{k+1}$ from $\approx_k$ (using only derivations of length at most 1) gives the same maximal fixed point. Consequently, in order to show that $P \approx_{\text{bis}} Q$ it is sufficient to prove the existence of a relation $\mathfrak{R}$ such that $\langle P, Q\rangle \in \mathfrak{R}$ and $\mathfrak{R} \subseteq \boldsymbol{E}(\mathfrak{R})$ where $\boldsymbol{E}$ denotes the simplified relation

$$\begin{aligned}\boldsymbol{E}(\mathfrak{R}) = \{\langle p, q\rangle \mid \forall a \in A \cup \{e\} \ &\text{(i) if } p \stackrel{a}{\Rightarrow} p' \text{ then } (\exists q')(q \stackrel{a}{\Rightarrow} q') \\ &\qquad \text{and } \langle p', q'\rangle \in \mathfrak{R} \\ &\text{(ii) if } q \stackrel{a}{\Rightarrow} q' \text{ then } (\exists p')(p \stackrel{a}{\Rightarrow} p') \\ &\qquad \text{and } \langle p', q'\rangle \in \mathfrak{R}\}\end{aligned}$$

Hence we get [24, Corollary 5.5]

$$\approx_{\mathbf{bis}} = \bigcup_{\Re \subseteq \boldsymbol{E}(\Re)} \Re.$$

Proposition 4.3.1. *Let P' and Q' be two processes. If $P' \approx_{\mathbf{bis}} Q'$ then $P' =_{\mathrm{J}} Q'$.*

To prove this proposition, we shall need the following definition.

Definition 4.3.2. Let A be a process. We define the *length of the process P* (denoted $\lambda(P)$) to be the height of the tree representing P.

Proof of Proposition 4.3.1. The proof is done by induction on the maximum length n of the processes; that is $n = \max\{\lambda(P'), \lambda(Q')\}$. If $n = 0$, the proposition is clearly true.

If $n \geqslant 1$ and $P' \approx_{\mathbf{bis}} Q'$ then there exists a relation $\Re$ such that $\langle P', Q'\rangle \in \Re$ and $\Re \subseteq \boldsymbol{E}(\Re)$. First let us show that $P' \leqslant_{\mathrm{J}} Q'$. Take $\langle s, \boldsymbol{Q}\rangle \in Q'$; we must show that there is a process $\boldsymbol{P}$ such that $\langle s, \boldsymbol{P}\rangle \in P'$ and $\boldsymbol{P} \leqslant_{\mathrm{J}} \boldsymbol{Q}$. Since $\langle P', Q'\rangle \in \Re$ and $\Re \subseteq \boldsymbol{E}(\Re)$, then $\langle P', Q'\rangle \in \boldsymbol{E}(\Re)$. But, by hypothesis, $Q' \overset{s}{\Rightarrow} \boldsymbol{Q}$; consequently, by the definition of $\boldsymbol{E}(\Re)$ there exists $\boldsymbol{P}$ such that $P' \overset{s}{\Rightarrow} \boldsymbol{P}$ and $\langle \boldsymbol{P}, \boldsymbol{Q}\rangle \in \Re$. Now since $\max\{\lambda(\boldsymbol{P}), \lambda(\boldsymbol{Q})\} < n$ and $\langle \boldsymbol{P}, \boldsymbol{Q}\rangle \in \Re$, we get by the induction hypothesis that $\boldsymbol{P} \leqslant_{\mathrm{J}} \boldsymbol{Q}$.

Since $\boldsymbol{E}$ is symmetrical, we also have that $Q' \leqslant_{\mathrm{J}} P'$. $\square$

Now it is possible to insert Johnston's equivalence in the diagram of Fig. 7 as shown in Fig. 8.

5. "Improvements" to Johnston's equivalence

One notices readily the awkward position of Johnston's equivalence in Fig. 8. In

$$\begin{tikzcd}
 & \approx \arrow[r, Leftrightarrow] \arrow[dl, Rightarrow] & \approx_{\mathrm{bis}} \arrow[d, Rightarrow] \\
=_{\mathrm{J}} \arrow[ddr, Rightarrow] & & \approx_2 \arrow[d, Rightarrow] \\
 & & \mathrm{te} \arrow[d, Rightarrow] \\
 & \approx_1 \arrow[r, Leftrightarrow] & \sim_s
\end{tikzcd}$$

Fig. 8.

order to bring back Johnston's equivalence onto the chain of equivalences that we already have, we shall make some slight modifications to the DCP model. In this section, we shall modify the partial order $\leqslant_{\mathbf{J}}$ so that the derived equivalence will fit in the chain of equivalences shown in Fig. 7. At the same time, we shall be able to distinguish between processes given in Proposition 4.2.2 which we do not want to identify since they do not have the same behaviour under all experiments.

If we look at Fig. 5, we see that these processes are **J**-equivalent solely because a deadlock represents the top element in the lattice of processes [6, 23] (that is, any process can *simulate* [6] a deadlock). To denote a deadlock, we introduce a special pair $\langle\sigma, \emptyset\rangle$ where $\sigma \notin A \cup \{\varepsilon\}$. We can now define a new partial order, which we shall denote $\leqslant_{\sigma\mathbf{J}}$.

Definition 5.1. Let P and Q be two processes. Then $P \leqslant_{\sigma\mathbf{J}} Q$ if and only if, after modifying the pairs defining P and Q in the following way: take every pair of P and Q of the form $\langle s, \emptyset\rangle$ $(s \in A)$ and change it into a pair of the form $\langle s, \{\langle\sigma, \emptyset\rangle\}\rangle$, then for each $e_2 \in A' = A \cup \{\varepsilon, \sigma\}$, if $\langle e_2, \boldsymbol{q}_2\rangle \in \boldsymbol{P}_2$ there exists $\langle e_1, \boldsymbol{q}_1\rangle \in \boldsymbol{P}_1$ such that $e_1 = e_2$ and $\boldsymbol{q}_1 \leqslant_{\mathbf{J}} \boldsymbol{q}_2$. That is, we use Definition 2.6 with a new alphabet A'.

Remark 5.2. Clearly this new relation $(\leqslant_{\sigma\mathbf{J}})$ is reflexive and transitive. Hence, $\leqslant_{\sigma\mathbf{J}}$ is a partial order on the set of processes.

Definition 5.3. Let P and Q be two processes. We say that P and Q are $(\sigma\mathbf{J})$-*equivalent* (denoted $P =_{\sigma\mathbf{J}} Q$) if and only if $P \leqslant_{\sigma\mathbf{J}} Q$ and $Q \leqslant_{\sigma\mathbf{J}} P$.

Proposition 5.4. *The equivalence relation* $=_{\sigma\mathbf{J}}$ *is strictly finer than the equivalence relation* $=_{\mathbf{J}}$.

Proof. It is easy to see that $P =_{\sigma\mathbf{J}} Q$ implies that $P =_{\mathbf{J}} Q$ since the only place they differ is in the treatment of deadlocks which are considered to be the top element in the lattice of processes defined by $\leqslant_{\mathbf{J}}$.

Now to show that $=_{\sigma\mathbf{J}}$ is strictly finer than $=_{\mathbf{J}}$, we must provide an example of two processes P_1 and P_2 such that $P_1 =_{\mathbf{J}} P_2$ and $\neg(P_1 =_{\sigma\mathbf{J}} P_2)$ are true. Let us take P_1 and P_2 as in Fig. 5. We know, by Proposition 4.2.2, that $P_1 =_{\mathbf{J}} P_2$ is true.

However, $P_1 =_{\sigma\mathbf{J}} P_2$ is not true; in fact, $P_2 \leqslant_{\sigma\mathbf{J}} P_1$ is not true. Take $\langle s, \{\langle\sigma, \emptyset\rangle\}\rangle \in P_1$. We must find $\langle e_2, q_2\rangle \in P_2$ such that $e_2 = s$ and $q_2 \leqslant_{\sigma\mathbf{J}} \{\langle\sigma, \emptyset\rangle\}$. Since

$$P_2 = \{\langle s, r_2\rangle, \langle d, r_3\rangle \mid r_2 = \{\langle y, \{\langle\sigma, \emptyset\rangle\}\rangle\} \text{ and } r_3 = \{\langle e, \{\langle\sigma, \emptyset\rangle\}\rangle\}\}$$

we must take $q_2 = r_2$. Consequently, we have to prove that $\{\langle y, \{\langle\sigma, \emptyset\rangle\}\rangle\} \leqslant_{\sigma\mathbf{J}} \{\langle\sigma, \emptyset\rangle\}$ is not true which is obviously so. $\square$

Even though this slight modification solves all problems for finite processes, we still are far from a good solution for recursively defined processes as shown by the following example.

Example 5.5. Let P and Q be the processes shown in Fig. 9. Then the reader can easily verify that $P =_{\sigma J} Q$. Clearly, this fact does not correspond to an acceptable situation since P will always be able to perform an *ab*-experiment whilst Q will not always be able to do so. To get rid of this difficulty, we shall rather use the following definition.

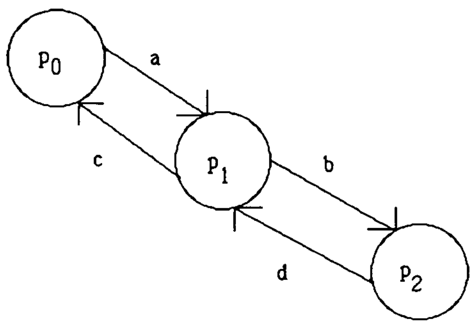
Definition 5.6. Following Definition 2.3, let $p = \{\langle e_1, q_1\rangle, \ldots, \langle e_n, q_n\rangle\}$ be a process. In the remainder of the paper the term *process* will represent an ordered pair $(p, \{e_1, \ldots, e_n\})$ whose first member is the process identifier and the second, the set of all events that can be performed immediately by p including possibly σ. Therefore, we shall write $(p, \{e_1, \ldots, e_n\}) = \{\langle e_1, q_1\rangle, \ldots, e_n, q_n\rangle\}$.

We can now define a new partial order, which we shall denote by $\leqslant_{\sigma\text{-J}}$.

Definition 5.7. Let $P = (p, B_1)$ and $Q = (q, B_2)$ be two processes written according to the preceding definition. Then $P \leqslant_{\sigma\text{-J}} Q$ if and only if $B_1 = B_2$ and $p \leqslant_{\sigma J} q$.

Remark 5.8. Clearly this new relation ($\leqslant_{\sigma\text{-J}}$) is reflexive and transitive. Hence, $\leqslant_{\sigma\text{-J}}$ is a partial order on the set of processes.

P:



Q:

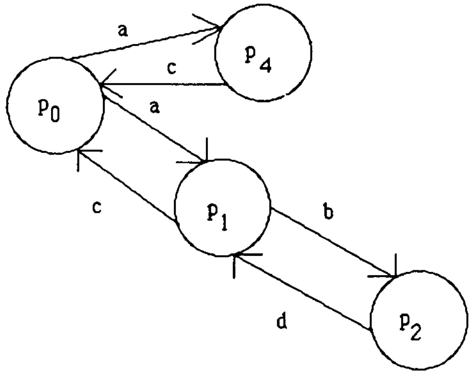


Fig. 9.

Definition 5.9. Let P and Q be two processes. We say that P and Q are (σ-**J**)-*equivalent* (denoted $P =_{\sigma\text{-}\mathbf{J}} Q$) if and only if $P \leqslant_{\sigma\text{-}\mathbf{J}} Q$ and $Q \leqslant_{\sigma\text{-}\mathbf{J}} P$.

Proposition 5.10. *The equivalence relation* $=_{\sigma\text{-}\mathbf{J}}$ *is strictly finer than the equivalence relation* $=_{\mathbf{J}}$.

Proof. This follows easily from Proposition 5.4 and the definition of $=_{\sigma\text{-}\mathbf{J}}$. $\square$

Proposition 5.11. *The example given in Example* 2.5 (*and Proposition* 4.2.1) *shows that there are processes trace equivalent* (*testing equivalent respectively*) *which are not* (σ-**J**)-*equivalent.*

Furthermore, one can easily prove the following result copying the proof used in Theorem 4.1.2 and Corollary 4.1.3. As a corollary of Proposition 5.10, we have the following result.

Theorem 5.12. *Let* P_1 *and* P_2 *be two processes, we have that* $P_1 =_{\sigma\text{-}\mathbf{J}} P_2$ *implies that* $P_1 \sim_{\mathbf{t}} P_2$.

Before proving the following theorem which will ensure $=_{\sigma\text{-}\mathbf{J}}$ a place within the chain of equivalences described in Fig. 7, we need the following lemma.

Lemma 5.13. *Let* $P_1 = (p_1, B_1)$ *and* $P_2 = (p_2, B_2)$ *be two processes such that* $P_1 =_{\sigma\text{-}\mathbf{J}} P_2$. *If* $\langle s, q_1\rangle \in P_1$ *then there is a pair* $\langle s, q_2\rangle \in P_2$ *such that* $q_1 =_{\sigma\text{-}\mathbf{J}} q_2$.

Proof. Suppose on the contrary that there is no such pair. Since $\langle s, q_1\rangle \in P_1$ and $P_1 =_{\sigma\text{-}\mathbf{J}} P_2$ then $B_1 = B_2$ and by definition of $\leqslant_{\sigma\text{-}\mathbf{J}}$ there is a pair $\langle s, q_4\rangle \in P_2$ such that $q_4 \leqslant_{\sigma\text{-}\mathbf{J}} q_1$. But $\langle s, q_4\rangle \in P_2$ and $P_1 =_{\sigma\text{-}\mathbf{J}} P_2$ imply the existence of a pair $\langle s, q_3\rangle \in P_1$ such that $q_3 \leqslant_{\sigma\text{-}\mathbf{J}} q_4$. If $q_3 = q_1$, we are done. Otherwise, we repeat this operation; since the processes that we consider are finitely branching, we shall eventually find two processes q_k and q_{k+2j} such that $q_k = q_{k+2j}$. $\square$

Theorem 5.14. *The equivalence relations* $=_{\sigma\text{-}\mathbf{J}}$ and $\approx_{\mathbf{bis}}$ *are the same relation* (*cf. Fig.* 10).

$$\begin{array}{ccccc} =_{\sigma\text{-}J} & \Longleftrightarrow & \approx & \Longleftrightarrow & \approx_{bis} \\ & & \Downarrow & & \\ & & \approx_2 & & \\ & & \Downarrow & & \\ & & te & & \\ & & \Downarrow & & \\ & \approx_1 & \Longleftrightarrow & \sim_s & \end{array}$$

Fig. 10.

Proof. Let $P_1 = (p_1, B_1)$ and $P_2 = (p_2, B_2)$ be two processes. We must prove that $P_1 =_{\sigma\text{-J}} P_2$ if and only if $P_1 \approx_{\mathbf{bis}} P_2$. By copying with slight modifications the proof of Proposition 4.3.1 we get that $P_1 \approx_{\mathbf{bis}} P_2$ implies that $P_1 =_{\sigma\text{-J}} P_2$.

Now suppose that $P_1 =_{\sigma\text{-J}} P_2$. We must find a relation $\mathfrak{R}$ such that $\langle P_1, P_2\rangle \in \mathfrak{R}$ and $\mathfrak{R} \subseteq \boldsymbol{E}(\mathfrak{R})$. We claim that $\mathfrak{R} = =_{\sigma\text{-J}}$ is such a relation. Take $s \in A \cup \{\varepsilon\}$. First, let $s = \varepsilon$. If $p_1 \overset{s}{\Rightarrow} q_1$, then there is a pair of the form $\langle \tau, q_1\rangle$ in P_1. Since $P_1 =_{\sigma\text{-J}} P_2$, $B_1 = B_2$. Therefore, there exists a pair $\langle \tau, q_2\rangle$ in P_2 such that $q_1 =_{\sigma\text{-J}} q_2$ by Lemma 5.13. Now let $s \in A$. If $P_1 \overset{s}{\Rightarrow} q_1$, we may suppose without loss of generality that we have: $P_1 \overset{s}{\rightarrow} q_1$, that is $\langle s, q_1\rangle \in P_1$. Since $P_1 =_{\sigma\text{-J}} P_2$, there exists a pair $\langle s, q_2\rangle$ in P_2 such that $q_1 =_{\sigma\text{-J}} q_2$ by Lemma 5.13. This concludes the proof. $\square$

6. Conclusion

We have studied the equivalence defined in the Discrete Communicating Processes model. In order to insert it in the chain of existing equivalences, we have slightly modified the definition of the partial order inducing this equivalence and introduce explicit deadlocks. The introduction of "explicit deadlock" was a nice way of smoothing the behaviour of the equivalence relation $=_{\mathrm{J}}$. In doing so we obtained a new equivalence finer than the original one and which turns out to be the bisimulation defined by Park [21].

The modification introduced does not unduly lengthen the automatic verification of processes' equivalence. In fact, it could even help to halt the verification process quicker.

Acknowledgment

The authors wish to thank Professor R. De B. Johnston for enlighting discussions and comments on an earlier draft of this paper. The authors also appreciate the comments of A. Finkel on an earlier draft of this paper.

References

[1] J.A. Bergstra and J.W. Klop, The algebra of communicating processes with abstraction, *Theoret. Comput. Sci.* **37** (1) (1985) 77-121.

[2] E. Brinksma and G. Scollo, Formal notions of implementation and conformance in LOTOS, Memorandum INF-86-13, University of Twente, The Netherlands, November 1986.

[3] E. Brinksma, G. Scollo and C. Steenbergen, LOTOS Specifications, their implementations, and their tests, in: B. Sarikaya, G. v. Bochmann eds., *Protocol Specification, Testing, and Verification VI* (North-Holland, Amsterdam, 1987) 349-360.

[4] S.D. Brookes, On the relationship between CCS and CSP, in: *Proc. ICALP'83*, Lecture Notes in Computer Science, Vol. 154 (Springer, Berlin, 1983) 83-96.

[5] S.D. Brookes, C.A.R. Hoare and A.W. Roscoe, A theory of communicating sequential processes, *J. ACM* **31** (1984) 560–599.
[6] R. De B. Johnston, A mathematical model of discrete communication processes, Tech. Report INRS-Télécommunications No. 85-16, April 1985.
[7] R. De Nicola, Extensional equivalences for transition systems, *Acta Inform.* **24** (1987) 211–237.
[8] R. De Nicola and M. Hennessy, Testing equivalences for processes, *Theoret. Comput. Sci.* **34** (1984) 83–133.
[9] M. Hennessy, Acceptance trees, *J. ACM* **32** (4) (1985) 896–928.
[10] M. Hennessy and R. Milner, Algebraic laws for nondeterminism and concurrency, *J. ACM* **32** (1985) 137–161.
[11] C.A.R. Hoare, Communicating sequential processes, *Comm. ACM* **21** (1978) 666–677.
[12] C.A.R. Hoare, A model for communicating sequential processes, Technical Monograph Prg-22, Computing Laboratory, University of Oxford, 1981.
[13] C.A.R. Hoare, *Processus Séquentiels Communicants* (Masson, Paris, 1987).
[14] ISO, Information Processing Systems-Open systems interconnection-LOTOS, A formal description technique based on the temporal ordering of observational behaviour, DIS 8807, July 1987.
[15] M. Itoh, Concurrency issues in CCS and the realizability by sequential processes, In preparation.
[16] R. Keller, Formal verification of parallel programs, *Comm. ACM* **19** (1976) 561–572.
[17] G.J. Milne, CIRCAL and the representation of communication, concurrency, and time, *ACM Trans. Programming Languages Syst.* **7** (2) 270–298.
[18] R. Milner, *A Calculus of Communicating Systems*, Lecture Notes in Computer Science, Vol. 92 (Springer, Berlin, 1980).
[19] R. Milner, Calculi for synchrony and asynchrony, *Theoret. Comput. Sci.* **25** (1983) 267–310.
[20] E. Moore, Gedanken experiments on sequential machines, in: C.E. Shannon and J. McCarthy eds., *Automata Studies*, (Princeton University Press, Princeton, 1956).
[21] D. Park, Concurrency and automata on infinite sequences, in: *Proc. 5th GI Conference*, Lecture Notes in Computer Science, Vol. 104 (Springer, Berlin, 1981) 167–183.
[22] K. Rea, Automating the analysis and synthesis of discrete communicating processes, Master's Thesis, McGill University, 1984.
[23] K. Rea and R. De B. Johnston, Automated analysis of discrete communication behaviour, *IEEE Trans. Software Engng.*, **13** (10) (1987) 1115–1126.
[24] M.T. Sanderson, Proof techniques for CCS, Ph.D. Thesis, University of Edinburgh, CST-19-82, 1982.
[25] M.B. Smyth, Powerdomains, Theory of computation, Report 12, Department of Computing Science, University of Warwick, 1976.
[26] A. Tarski, A lattice-theoretical fixpoint theorem and its applications, *Pacific J. Math.* **5** (1955) 285–309.