Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J.C.M. Baeten, J.A. Bergstra, J.W. Klop

Ready trace semantics for concrete process algebra with priority operator

Department of Computer Science Report CS-R8517 August

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

READY TRACE SEMANTICS FOR CONCRETE PROCESS ALGEBRA WITH PRIORITY OPERATOR

J.C.M. BAETEN

Department of Computer Science, University of Amsterdam

J.A. BERGSTRA

Department of Philosophy, University of Utrecht; Department of Computer Science, University of Amsterdam

J.W. KLOP

Centre for Mathematics and Computer Science, Amsterdam

ABSTRACT

We consider a process semantics intermediate between bisimulation semantics and readiness semantics, called here ready trace semantics. The advantage of this semantics is that, while retaining the simplicity of readiness semantics, it is still possible to augment this process model with the mechanism of atomic actions with priority (the θ operator). It is shown that in readiness semantics and a fortiori in failure semantics such an extension with θ is impossible. Ready trace semantics is considered here in the simple setting of concrete process algebra, that is: without abstraction (no silent moves), moreover for finite processes only. For such finite processes without silent moves a complete axiomatisation of ready trace semantics is given via the method of process graph transformations.

1980 Mathematics Subject Classification: 68B10, 68C01, 68D25, 68F20
1982 CR Categories: F.1.1, F.1.2, F.3.2, F.4.3 69F11, 69F12, 69F32, 69F43
KEY WORDS & PHRASES: process algebra, concurrency, readiness semantics, failure semantics, ready trace semantics, priority operator
NOTE: This research is partially supported by ESPRIT contract 432, METEOR.
This report will be submitted for publication elsewhere.

Report CS-R8517
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

INTRODUCTION

In BAETEN, BERGSTRA & KLOP [1] an operator θ is introduced, in the setting of bisimulation semantics, which introduces priorities on the atomic steps in a process. This priority operator is useful in process specification; e.g. one can model interrupt mechanisms with it (see [1]). In BERGSTRA, KLOP & OLDEROG [2] a complete axiomatisation for finite processes with communication but without silent moves, is given for readiness semantics and failure semantics.

Now the starting point for this paper is the question whether θ can consistently be added to readiness and failure semantics as expounded in [2]. This is not obvious, since readiness and failure semantics equate many more processes than bisimulation semantics does. Indeed it turns out that θ and readiness or failure semantics are inconsistent (section 1).

The question next considered is whether there is a process semantics "close to" readiness and failure semantics to which θ can be consistently added. It turns out that there is a very natural semantics with this property: RTS, ready trace semantics, which is interesting for its own sake. In PNUELI [8], RTS is called 'barbed semantics'. We give a complete axiomatisation for finite τ-less processes under this semantics, as well as a complete axiomatisation (RTS_θ) which takes moreover θ into account. The method of proof (to obtain the completeness results) is via process graph transformations which enjoy the termination and confluency property, as in [2].

This paper can be read independently, but it is useful to have seen [1,2]. Some general references are: for bisimulation semantics, MILNER [6]; for readiness semantics, OLDEROG & HOARE [7], for failure semantics, BROOKES, HOARE & ROSCOE [5], and for a connection between bisimulation and failure semantics, BROOKES [4]. A more complete list of references can be found in [1,2].

CONTENTS

1. THE INCOMPATIBILITY OF READINESS AND FAILURE SEMANTICS WITH THE PRIORITY OPERATOR

We start with the simple demonstration of the fact that it is impossible to extend readiness and (a fortiori) failure semantics with the priority operator θ, as this observation is one of the primary motivations for the introduction of the ready trace semantics below. First we will make precise the different concepts involved in this observation. We recall the following facts from BERGSTRA, KLOP & OLDEROG [2].

1.1. THEOREM. *The axiom systems* BPA_δ, BPA_δ + R1,2, BPA_δ + R1,2 + S *in Table 1 are complete axiomatisations of respectively bisimulation semantics, readiness semantics and failure semantics on finite processes (over alphabet* A *without* τ *).*

BPA_δ	$x + y = y + x$	A1
	$(x + y) + z = x + (y + z)$	A2
	$x + x = x$	A3
	$(x + y)z = xz + yz$	A4
	$(xy)z = x(yz)$	A5
	$\delta + x = x$	A6
	$\delta x = \delta$	A7
BPA_δ + R1,2	$a(bx + u) + a(by + v) = a(bx + by + u) + a(bx + by + v)$	R1
	$a(b + u) + a(by + v) = a(b + by + u) + a(b + by + v)$	R2
BPA_δ + R1,2 + S	$ax + a(y + z) = ax + a(x + y) + a(y + z)$	S

TABLE 1

(A is the alphabet of actions; δ is the deadlock constant and $A_\delta = A \cup \{\delta\}$. The variables a,b range over A_δ and x,y,z,u,v range over all processes.)

The notions of bisimulation semantics, readiness semantics and failure semantics are (essentially) standard and well-known; for details of their definitions referring to the present situation where two termination possibilities exist (viz. a trace may end in δ, unsuccessfully, or in a proper action $\in$ A, successfully) we refer to [2].

1.1.1. REMARK. In [2] a generalisation of Theorem 1.1 is proved where also parallel operators and communication are present. For the purpose of this section we do not need these. In section 2 and later we will consider communication as well (i.e. BPA_δ is replaced by ACP).

1.2. In BAETEN, BERGSTRA & KLOP [1] the priority operator θ was introduced which enables one to express that some actions have priority in a choice (a sum) over others. 'Priorities' are given by a partial order $>$ on A_δ where δ is always the least element. So if the ordering $a > b$ is adopted, then

$$\theta(ax + by + z) = \theta(ax + z).$$

Another property of θ is that

$$\theta(ax) = a\theta(x), \text{ and } \theta(ax + ay) = a\,\theta(x) + a\,\theta(y), \text{ etc.}$$

Below we will give a complete (and even finite) axiomatisation of θ, but for the time being these properties of θ suffice. As shown in [1], θ is vital for modeling in process algebra features such as interrupt mechanisms. In [1] the operator θ is introduced in the setting of bisimulation semantics. It is hence an obvious question whether such a natural operator can "consistently" be added to readiness and failure semantics. Here we should explain what is meant by "consistently": as an ultimate criterion for consistency of a process axiomatisation T we require that T does not derive an equation $t_1 = t_2$ between finite closed process expressions such that $trace(t_1) \neq trace(t_2)$. Here $trace(t)$ is defined in such a way that termination δ's are visible, e.g. $trace(a + b\delta) = \{a, b\delta\}$. However, $trace(a + \delta) = \{a\}$, since $a + \delta = a$ (cf. axiom A6 in Table 1). For a precise definition (which moreover involves τ-steps) see [3].

The definite answer to the question just raised is negative, as the following counterexamples show.

1.3. PROPOSITION. *Failure semantics with priority operator is inconsistent. In particular,* $BPA_\delta + R1,2 + S + \theta$ *is inconsistent.*

PROOF. Consider process expressions

t_1: $ab + a(c + d)$

t_2: $ab + a(c + d) + a(b + c)$.

According to Table 1 (axiom S), $BPA_\delta + R1,2 + S + \theta' \vdash t_1 = t_2$. Hence in this system $BPA_\delta + R1,2 + S + \theta$ we have $\theta(t_1) = \theta(t_2)$. We adopt the following priority of atoms: $b < c < d$. Now w.r.t. this ordering:

$$\theta(t_1) = a\,\theta(b) + a\,\theta(c+d) = ab + ad$$

$$\theta(t_2) = a\,\theta(b) + a\,\theta(c+d) + a\,\theta(b+c) = ab + ad + ac.$$

So in the axiom system under consideration we derive the equality of two expressions with different trace set. □

In fact, the previous proposition is strengthened by the following, which shows that the inconsistency is already obtainable in readiness semantics:

1.4. PROPOSITION. *Readiness semantics with priority operator is inconsistent. In particular,* $BPA_\delta + R1,2 + \theta$ *is inconsistent.*

PROOF. Let

$$t_1 \equiv a(bc + d) + a(be + f)$$

$$t_2 \equiv a(be + d) + a(bc + f).$$

Adopt the priorities: $d > b > f$. Now $BPA_\delta + R1,2 + \theta \vdash t_1 = t_2$ (using R1), but

$$\theta\,(t_1) = ad + abe$$

$$\theta\,(t_2) = ad + abc. \qquad \square$$

2. READY TRACE SEMANTICS: A MODEL OF FINITE PROCESSES

We will now introduce ready trace equivalence on processes, which will be in this paper finite and τ-less.

2.1. DEFINITION. Let $\mathbb{H}$ be the domain of finite acyclic process graphs with edges labeled by elements from A_δ. The graphs $g \in \mathbb{H}$ will be supposed to be in δ-normal form, i.e. δ-steps may only occur at the end of branches of g and may have no alternatives.

On $\mathbb{H}$ we define operations $+, \cdot, \|, \lfloor\!\lfloor, |, \partial_H$ as defined in BERGSTRA, KLOP & OLDEROG [2]. These definitions are supposed known in this paper.

For the sake of completeness we include here the definition of the ready set $\mathcal{R}[\![g]\!]$ of $g \in \mathbb{H}$:

2.2. DEFINITION. Let σ vary over A^*, the set of words over the action alphabet A (not A_δ). The empty word is λ . Let $g \in \mathbb{H}$. Then the ready set of g, notation: $\mathcal{R}[\![g]\!]$, is the least set satisfying the following clause:

for all $\sigma \in A^*$, $g \xrightarrow{\sigma} h$ implies $(\sigma, I(h)) \in \mathcal{R}[\![g]\!]$.

Here $g \xrightarrow{\sigma} h$ (h is a derivation of g via σ) if there is a path, determining the word σ, from the root of g to the root of the subgraph h.

Further, I(h) is the set of initial steps of h; if h is a single δ-step then $I(h) = \emptyset$ and if h is $\mathbb{0}$, the zero graph $\overset{\downarrow}{o}$ without edges, $I(h) = I(\mathbb{0}) = \{\varepsilon\}$. ($\varepsilon$ is a formal symbol denoting succesful termination.)

2.3. EXAMPLE. If g is the process graph in Figure 1:

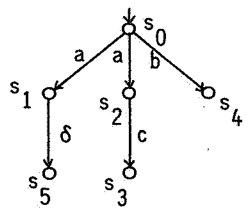


Figure 1.

then $\mathcal{R}[\![g]\!] = \{(\lambda, \{a,b\}), (a,\emptyset), (a,\{c\}), (ac, \{\varepsilon\}), (b, \{\varepsilon\})\}$.

(The displayed contributions in $\mathcal{R}[\![g]\!]$ are yielded by nodes, respectively, s_0, s_1, s_2, s_3, s_4. Note that s_5 gives no contribution.)

2.4. DEFINITION. $g,h \in \mathbb{H}$ are ready equivalent if $\mathcal{R}[\![g]\!] = \mathcal{R}[\![h]\!]$; notation $g \equiv_{\mathcal{R}} h$.

It is proved in BERGSTRA, KLOP & OLDEROG [2] that ready equivalence is a congruence w.r.t. the operations $+, \cdot, \|, \lfloor\!\lfloor, |, \partial_H$; when restricting ourselves to $+, \cdot$ we have the isomorphism

$$\mathbb{H}(+,\cdot)/\equiv_{\mathcal{R}} \;\cong\; I(BPA_\delta + R1,2)$$

where I(-) is the initial algebra of -. (This is part of the contents of Theorem 1.1 above.)

We now turn to ready trace semantics (RTS); here less processes (process expressions) are identified than in ready semantics (RS) as explained above. The essential difference is that while RS is based on the notion of ready pair (σ, X) (see Figure 2(a)), RTS is based on the notion of a ready trace (see Figure 2(b)), where also the "intermediate" ready sets X_i along trace σ are given.

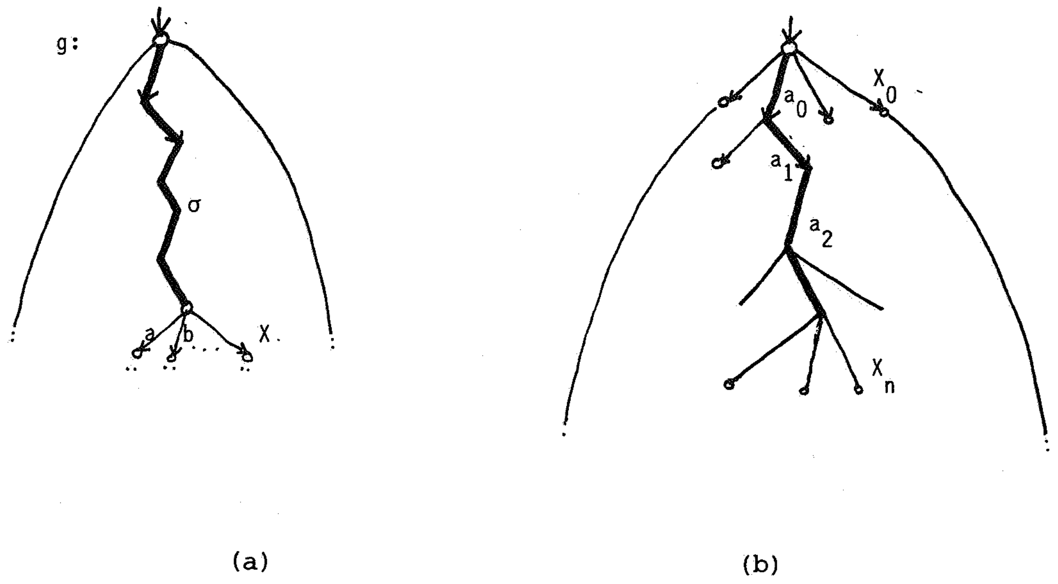


Figure 2.

2.5. DEFINITION. (i) Let $g \in \mathbb{H}$. A path π in g, starting from the root s_0 of g, is an alternating sequence of nodes of g and labeled edges in g:

$$\pi = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \ldots . \xrightarrow{a_{n-1}} s_n$$

with $a_i \in A$ (not A_δ). Here $n \geqslant 0$; if $n = 0$ we have the empty path. All paths in this paper will start from the root, so we omit that qualification. A path is maximal if it cannot be prolonged, that is, either ending in a terminal node ($\emptyset$) or in the initial node of a δ-step.

(ii) If $s \in$ NODES(g), $(g)_s$ will be the subgraph of g with root s and consisting of all labeled edges accessible from s.

(iii) As before, I(g) is the set of initial steps of g, with $I(\emptyset) = \{\varepsilon\}$ and $I(\delta) = \emptyset$. Instead of $I((g)_s)$ we write just I(s).

2.6. DEFINITION. (i) Let $g \in \mathbb{H}$ and $\pi = s_0 \xrightarrow{a_0} \ldots \xrightarrow{a_{n-1}} s_n$ be a path in g. Then rt(π), the ready trace corresponding to π, is the alternating sequence of ready sets $I(s_i)$ and steps a_i:

$$I(s_0), a_0, I(s_1), a_1, \ldots, a_{n-1}, I(s_n)$$

$(n \geqslant 0)$. We will use sometimes the notation $(\sigma ; \vec{X})$ for such a ready trace where $\sigma = a_0 a_1 \ldots a_{n-1}$ and $\vec{X} = I(s_0), I(s_1), \ldots, I(s_n)$. The ready trace corresponding to the empty path of g is just I(g), or in the $(\sigma ; \vec{X})$ notation, $(\lambda, I(g))$.

(ii) The ready trace set of g, notation $\mathcal{RT}[\![g]\!]$, is

$$\{rt(\pi) \mid \pi \text{ a path in g, starting from the root}\}.$$

(iii) $g \equiv_{\mathcal{RT}} h$ if $\mathcal{RT}[\![g]\!] = \mathcal{RT}[\![h]\!]$; in words: g,h are ready trace equivalent.

2.7. EXAMPLE. (i) Let g,h be as in Figure 3.

Figure 3.

Then $g \equiv_{\mathcal{RT}} h$.

(ii) g,h as in Figure 4 are not ready trace equivalent (cf. the counterexample in Proposition 1.4):

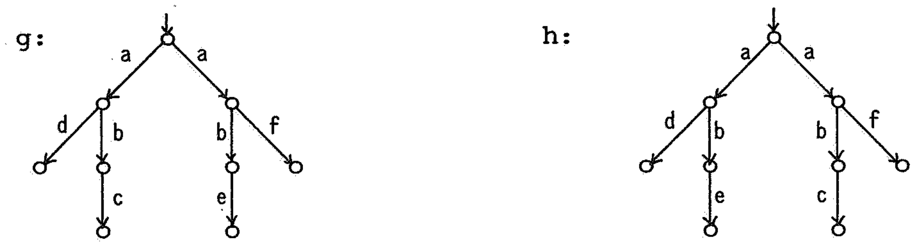


Figure 4.

Namely, $\mathcal{RT}[\![g]\!]$ contains the ready trace $\{a\},a,\{d,b\},b,\{c\},c,\{\varepsilon\}$ which is not present in $\mathcal{RT}[\![h]\!]$.

2.8. REMARK. In fact, in the present setting of finite acyclic process graphs, it would have been sufficient to consider for the definition of $\equiv_{\mathcal{RT}}$ only ready traces corresponding to maximal paths π. The present definition which includes also "prefixes" of such ready traces, anticipates working with infinite processes - which we will not do in this paper.

2.9. REMARK. A convenient "intuition" about a ready trace is this: Imagine an interactive session with a machine as follows.
At the start of the session the machine presents the user a menu of all possible actions which the user may perform (I(g)); one of these is chosen (a_0), whereupon the machine again flashes the menu of the options in that state (I(s_1)), and so on. Any moment the user may end the session, that is, leave the machine which has on its screen the last menu (I(s_n)). So a ready trace is a record of such a session.

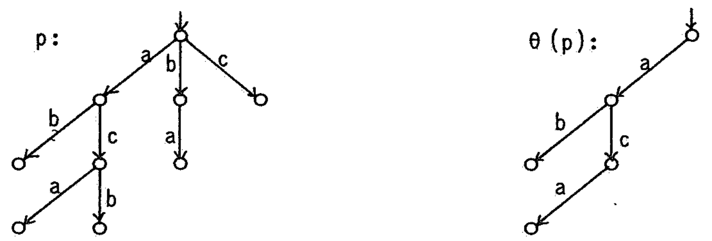
We will prove in the sequel that $\equiv_{\mathcal{RT}}$ is a congruence on $\mathbb{H}$ w.r.t. $+,\cdot,\|,\lfloor\!\lfloor,|,\partial_H$, and also w.r.t. the priority operator θ which will be defined now.

2.10. DEFINITION. Let a partial order $<$ on A_δ be given such that $\delta < a$ for all $a \in A$. Then $\theta_<$, or θ for short, is defined on $\mathbb{H}$ as follows:
$\theta(g)$ is the process graph arising from g by

(i) erasing all edges leaving node s which have a label 'a' majorised by label 'b' of some other edge leaving s;

(ii) discarding all parts of g which thus have become disconnected.

2.11. EXAMPLE. Let $a > b$ and $a > c$. Then for p as in Figure 5(a), $\theta(p)$ is as in Figure 5(b):



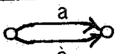
(a) (b)

Figure 5.

3. PROCESS GRAPH TRANSFORMATIONS FOR READY TRACE SEMANTICS

In order to prove that $\equiv_{\mathcal{RT}}$ is a congruence on $\mathbb{H}$, and to derive the completeness result in Theorem 4.2 below, we will introduce three process graph transformations of which the first two (already used in BERGSTRA, KLOP & OLDEROG [2]) are specific for bisimulation semantics and the third is specific for RTS.

3.1. The transformations double edge, sharing, narrowing.

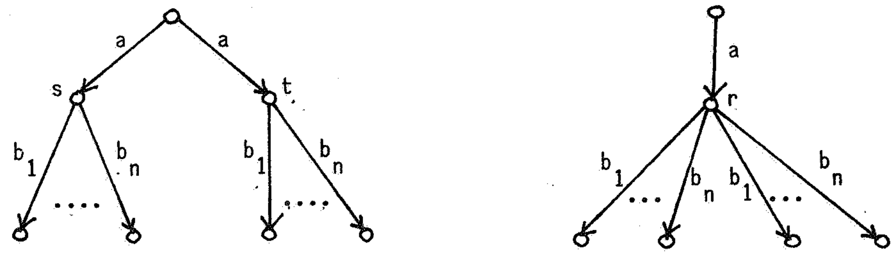
[i] double edge. This process graph transformation step removes in a "double edge"  ($a \in A_\delta$) one of the edges.

Notation: $g \overset{}{\underset{[i]}{\Longrightarrow}} h$.

[ii] sharing. Suppose $g \in \mathbb{H}$ contains two nodes s,t determining identical subgraphs $(g)_s, (g)_t$. Then the nodes s,t may be identified.

Notation: $g \underset{[ii]}{\Longrightarrow} h$.

[iii] narrowing. If $g \in \mathbb{H}$ contains a part as in Figure 6(a) this may be replaced by the part as in Figure 6(b). More precisely, a new node r is created together with edges as in Figure 6(b), and the old a-edges to s,t are discarded. The nodes s,t and the b_i-edges leaving them (in Figure 6(a)) are not discarded since s,t may have other incoming edges. (If not, then s,t are inaccessible from the root and disappear.)



(a) (b)

Figure 6.

Here $I(s) = I(t) = b_1, \ldots, b_n$; the b_i may have multiple occurrences among the labels of edges leaving nodes s,t.

Notation: $g \underset{[iii]}{\Longrightarrow} h$.

3.2. NOTATION. $\Longrightarrow$ is $\underset{[i]}{\Longrightarrow} \cup \underset{[ii]}{\Longrightarrow} \cup \underset{[iii]}{\Longrightarrow}$;

$\Longrightarrow\!\!\!\!\!\!\Rightarrow$ is the transitive reflexive closure of $\Longrightarrow$, and $\Longleftarrow\!\!\!\Longrightarrow$ is the equivalence relation generated by $\Longrightarrow$.

3.3. EXAMPLE. (i)

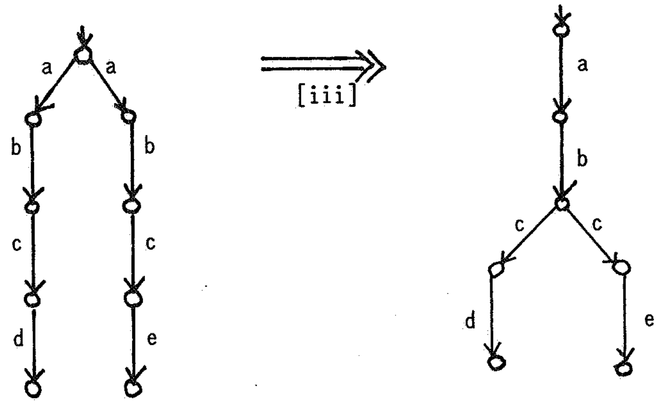


Figure 7.

(ii)

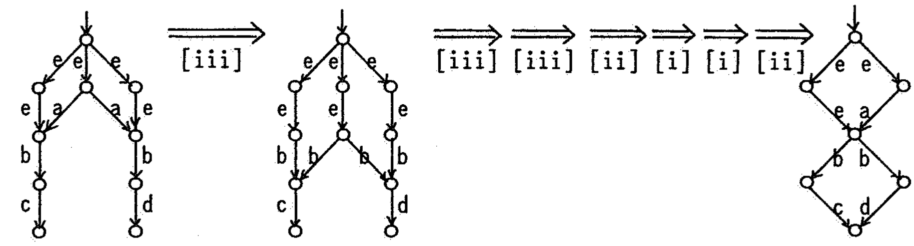


Figure 8.

3.4. **PROPOSITION.** *Let* $g,h \in \mathbb{H}$. *Then graph reduction is sound w.r.t. ready trace equivalence, i.e.*

$$g \Longrightarrow h \ \textit{implies}\ g \equiv_{\mathcal{RT}} h.$$

PROOF. It is simple to check that each of the three graph reductions keeps the ready trace set invariant. □

We will now establish the completeness of the graph transformations.

3.5. **DEFINITION.** $g \in \mathbb{H}$ is in ready trace normal form (rt-normal form) if none of the graph reductions is applicable to g.

3.6. **PROPOSITION.** *Every graph reduction* $g \Rightarrow g' \Rightarrow g'' \Rightarrow \ldots$ *must end eventually in a rt-normal form.*

PROOF. Let T(g) be the tree obtained from g by unsharing. Let card(g) be the number of nodes in T(g). Then transformations of type [iii] have the effect of decreasing card(g), while types [i],[ii] do not increase card(g). Since [i],[ii]-transformations clearly must end eventually, this proves the proposition.
(Note that the detour via T(g) is necessary since a type [iii] transformation may increase the number of nodes in g, as Example 3.3(ii) shows: the second graph has one node more than the first.) □

3.7. **PROPOSITION.** *Let* g,h *be in rt-normal form and suppose* $g \equiv_{\mathcal{RT}} h$. *Then* g,h *are identical.*

PROOF. The proof consists of remarking that an rt-normal form g can be uniquely reconstructed from its ready trace set. The reconstruction is as follows. Let the elements of $\mathcal{RT}[\![g]\!] \cup \{o\}$ be the nodes in a process graph g* to be constructed. The root of g^* is $I(g)$ (or $(\lambda\,;I(g))$ in the $(\sigma\,;\vec{X})$ notation). The edges of g^* are given by

$$(\sigma\,;\vec{X}) \xrightarrow{a} (\sigma a;\vec{X},Y) \quad \text{if the last entry of } \vec{X} \text{ is not } \emptyset$$

$$(\sigma\,;\vec{X},\emptyset) \xrightarrow{\delta} o.$$

Now it is a routine matter to show that g^* is in fact isomorphic to $T(g)$. So we have proved that $T(g)$ and $T(h)$ are identical (or isomorphic). From this it readily follows that g,h are identical. □

Although we will not need it, let us remark the following fact:

3.7.1. COROLLARY. *Process graph transformations* $\Rightarrow\!\!\!\!\Rightarrow$ *are confluent.*

PROOF. Immediate from Propositions 3.4, 3.6 and 3.7. □

3.8. LEMMA. $g \Leftarrow\!\!\!\Rightarrow\!\!\!\!\Rightarrow h$ *iff* $g \equiv_{\mathcal{RT}} h$.

PROOF. The implication $\Rightarrow$ is immediate from Proposition 3.4. For the reverse implication, suppose g,h are rt-equivalent. Let g^* be a rt-normal form of g, obtained by a maximal $\Rightarrow$ -graph reduction (it exists by Proposition 3.6). Likewise h^* is a rt-normal form of h. Then by Proposition 3.4, $g \equiv_{\mathcal{RT}} g^*$ and $h \equiv_{\mathcal{RT}} h^*$, and hence $g^* \equiv_{\mathcal{RT}} h^*$. So by Proposition 3.7, g^* and h^* are identical, and therefore $g \Rightarrow\!\!\!\!\Rightarrow g^* \Leftarrow\!\!\!\!\Leftarrow h$. □

3.9. LEMMA. $\equiv_{\mathcal{RT}}$ *is a congruence on* $\mathbb{H}(+,\cdot,\|,\|\!\_\,,|,\partial_H,\theta)$.

PROOF. Using the previous lemma, it suffices to prove:

if $g \Rightarrow g'$, $h \Rightarrow h'$ then (i) $g \square h \Leftarrow\!\!\!\Rightarrow\!\!\!\!\Rightarrow g' \square h'$ where $\square$ is $+,\cdot,\|,\|\!\_\,,|$.

(ii) $\partial_H(g) \Leftarrow\!\!\!\Rightarrow\!\!\!\!\Rightarrow \partial_H(g')$

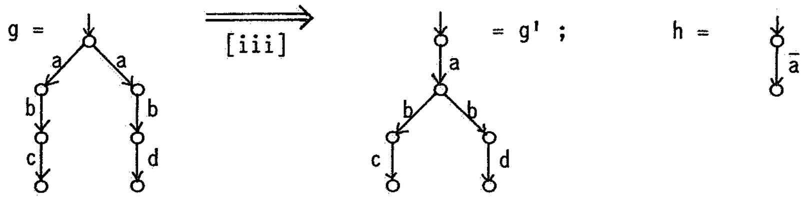
(iii) $\theta(g) \Leftarrow\!\!\!\Rightarrow\!\!\!\!\Rightarrow \theta\; g')$.

Of these implications (ii), (iii) are trivial.

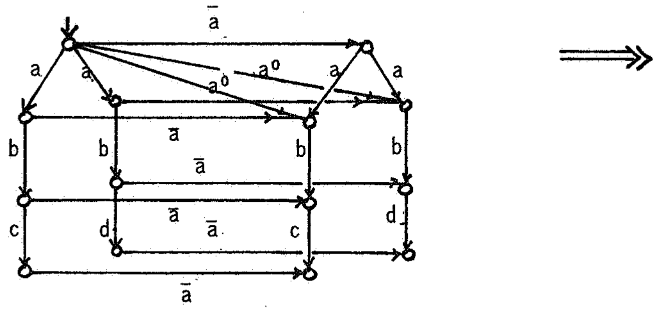
(i) is easy for the operations $+,\cdot$. For $\|$ it suffices to prove $g\|h \Rightarrow\!\!\!\!\Rightarrow g'\|h$, which best can be seen using some "geometrical intuition" (cf. Example 3.10) - or, alternatively, one proves more directly that $g \Rightarrow g'$ implies $g\|h \equiv_{\mathcal{RT}} g'\|h$. The details of a really rigorous proof would be extremely time and space consuming, and we will not attempt to do so in the present note.

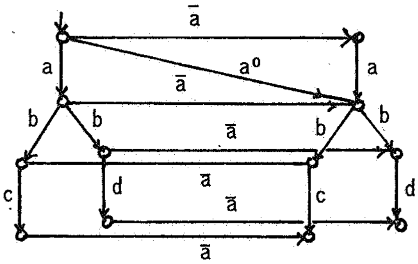
The operators $⌊⌊$ and $|$ present no special difficulties. □

3.10. EXAMPLE. See Figure 9. Let $a|\bar{a} = a°$ be the only nontrivial communication.



Then $g||h$ =





= $g'||h$.

Figure 9. (Note that while $g \xrightarrow[\text{[iii]}]{} g'$, the sequence of transformations $g||h \twoheadrightarrow g'||h$ uses all three types of transformation.)

3.11. REMARK. In BERGSTRA, KLOP & OLDEROG [2] it was proved that readiness equivalence on $\mathbb{H}$ ($\equiv_{\mathcal{R}}$) is generated by the graph transformations $\xrightarrow[\text{[i]}]{}$ (double edge), $\xrightarrow[\text{[ii]}]{}$ (sharing) as before, together with the transformation "cross" by which in a part as in Figure 10(a) two b-steps may be inserted to yield the part as in Figure 10(b).

Figure 10. (a) (b)

Using this fact we can pinpoint once more (cf. also Proposition 1.4) why $\equiv_{\mathcal{R}}$ is not a congruence w.r.t. θ . Namely, if g,g' are the graphs in Figure 11(a),(b), then $\theta(g)$, $\theta(g')$, with priorities $d > b > f$, are as in Figure 11(b):

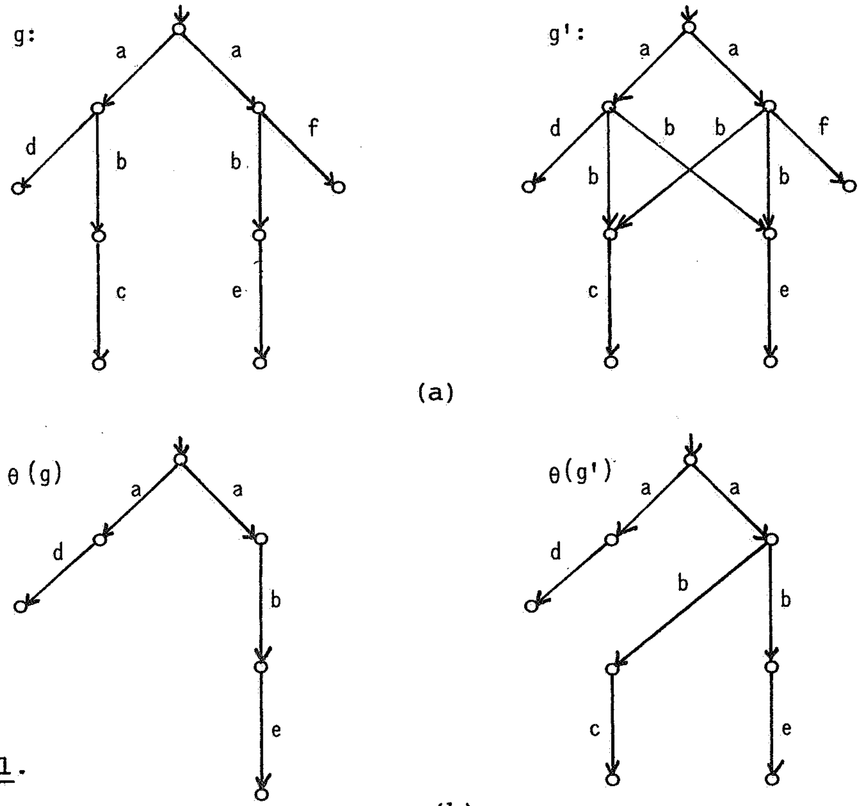


Figure 11.

Now clearly $\theta(g)$ and $\theta(g')$ are not convertible via [i],[ii], cross transformations. Contrast this with the present situation where via some casuistics it is easily seen that if $g \underset{[iii]}{\Longrightarrow} g'$ then: $\theta(g) \underset{[iii]}{\Longrightarrow} \theta(g')$ or $\theta(g)$, $\theta(g')$ coincide.

3.12. Some auxiliary operators on $\mathbb{H}$.

We will add two more operators on $\mathbb{H}$, which serve to axiomatise θ in a finite way and to formulate a proof rule typical for RTS.

3.12.1. DEFINITION. Let $g,h \in \mathbb{H}$. Then $g \lhd h$ (g 'unless' h) is defined as the result of erasing in g all initial steps which are majorised (w.r.t. the p.o. $<$ to which the priority operator refers) by some initial step in h. (Of course, disconnected pieces are discarded.)

3.12.2. EXAMPLE. Let $a < b < c$. Then:

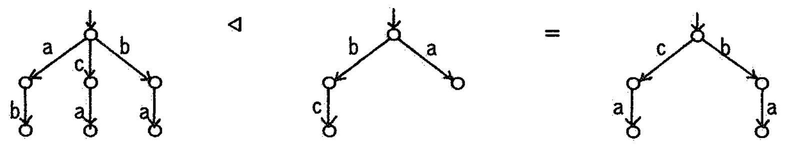


Figure 10.

3.12.3. DEFINITION. $\pi_n : \mathbb{H} \longrightarrow \mathbb{H}$ is the n-th projection operator $(n \geq 1)$ which cuts off all branches after n steps. E.g.

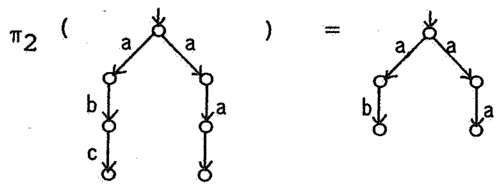


Figure 11.

It is left to the reader to check that Lemma 3.9 generalises to the presence of $\lhd$ and the π_n $(n \geq 1)$.

4. A COMPLETE AXIOMATISATION FOR READY TRACE SEMANTICS

The previous results lead at once to a complete axiomatisation RTS as in Table 2 for ready trace semantics on finite processes with communication (but without silent moves):

RTS

$x+y = y+x$	A1
$x+(y+z) = (x+y)+z$	A2
$x+x = x$	A3
$(x+y)z = xz+yz$	A4
$(xy)z = x(yz)$	A5
$x+\delta = x$	A6
$\delta x = \delta$	A7
$a\mid b = b\mid a$	C1
$(a\mid b)\mid c = a\mid(b\mid c)$	C2
$\delta\mid a = \delta$	C3
$x\parallel y = x ⌊⌊ y + y ⌊⌊ x + x\mid y$	CM1
$a ⌊⌊ x = ax$	CM2
$ax ⌊⌊ y = a(x\parallel y)$	CM3
$(x+y) ⌊⌊ z = x ⌊⌊ z + y ⌊⌊ z$	CM4
$(ax)\mid b = (a\mid b)x$	CM5
$a\mid(bx) = (a\mid b)x$	CM6
$(ax)\mid(by) = (a\mid b)(x\parallel y)$	CM7
$(x+y)\mid z = x\mid z+y\mid z$	CM8
$x\mid(y+z) = x\mid y+x\mid z$	CM9
$\partial_H(a) = a$ if $a\notin H$	D1
$\partial_H(a) = \delta$ if $a\in H$	D2
$\partial_H(x+y) = \partial_H(x)+\partial_H(y)$	D3
$\partial_H(xy) = \partial_H(x)\cdot\partial_H(y)$	D4
$\pi_m(a)=a$	PR1
$\pi_1(ax)=a$	PR2
$\pi_{m+1}(ax)=a\pi_m(x)$	PR3
$\pi_m(x+y)=\pi_m(x)+\pi_m(y)$	PR4
$\dfrac{\pi_1(x) = \pi_1(y)}{z(x+y) = zx+zy}$	RTR

TABLE 2

Here the upper part of Table 2 is ACP, Algebra of Communicating Processes; a,b vary over A_δ . It contains BPA_δ . The π_n are projection operators; to formulate RTR, the ready trace rule, π_1 is sufficient. Note that for finite processes, RTR is equivalent with the rule

$$\frac{\pi_1(x) = \pi_1(y)}{a(x+y) = ax + ay}$$

4.1. <u>LEMMA</u>. BPA_δ + PR1-4 + RTR *is a complete axiomatisation of* $\mathbb{H}(+,\cdot,\pi_n)/\equiv_{RT}$

<u>PROOF</u>. We refer to BERGSTRA, KLOP & OLDEROG [2] for the (obvious) interpretation of process expressions in the graph model and for the arguments concerning +, •. Clearly, RTR corresponds to the process graph transformation $\xRightarrow[\text{[iii]}]{}$. By Lemma 3.8 we have completeness. □

The extension to the priority operator θ can easily be done on the basis of the axiomatisation ACP_θ introduced and analysed in BAETEN, BERGSTRA & KLOP [1]. ACP_θ is the axiom system consisting of ACP (upper part of Table 2) and the nine axioms in Table 3:

$a \lhd b = a$ if not $(a < b)$	P1
$a \lhd b = \delta$ if $a < b$	P2
$x \lhd yz = x \lhd y$	P3
$x \lhd (y+z) = (x \lhd y) \lhd z$	P4
$xy \lhd z = (x \lhd z)y$	P5
$(x+y) \lhd z = x \lhd z + y \lhd z$	P6
$\theta(a) = a$	TH1
$\theta(xy) = \theta(x) \cdot \theta(y)$	TH2
$\theta(x+y) = \theta(x) \lhd y + \theta(y) \lhd x$	TH3

TABLE 3

We will refer to RTS together with the axioms in Table 3 as: RTS_θ .

4.2. <u>THEOREM</u>. RTS_θ *is a complete axiomatisation of the graph model*

$$\mathbb{H}(+,\cdot,\|,\llfloor,|,\partial_H,\pi,\lhd,\theta)/\equiv_{RT}$$

PROOF. The proof is entirely similar to the one for readiness semantics in BERGSTRA, KLOP & OLDEROG [2], using Lemma 4.1 and the fact that all operators except the basic operators +,. can be eliminated from process expressions. □

4.3. COROLLARY. RTS_θ *is a consistent (in the sense of Section 1.2) axiomatisation.*

PROOF. The elements of $\mathbb{H}(-)/\equiv_{\mathcal{RT}}$ are $\equiv_{\mathcal{RT}}$ -equivalence classes of graphs; each equivalence class is generated by the three graph transformations of section 2. These transformations preserve traces. Hence, by Lemma 4.2, the axiom system RTS_θ is consistent. □

5. AN EXPLICIT PRESENTATION OF THE MODEL FOR READY TRACE SEMANTICS

Above, we have obtained a model for RTS with as elements equivalence classes of process graphs. It is also possible to present this model in a more direct way, namely with the ready trace sets themselves as elements (without any mention of underlying process graphs). This requires formulating some closure properties of ready trace sets. (Cf. the analogous procedure in BERGSTRA, KLOP & OLDEROG [2] for failure semantics.) The end result will be an 'explicit' model for RTS, which is isomorphic to the 'graph model' above. In order to obtain this explicit representation, we have to define the operations $+, \cdot, \|$, $⌊⌊, |, \partial_H, \theta, \triangleleft, \pi_n$ directly on the ready trace sets. We will do this only for θ, and further be satisfied with the formulation of the closure properties inherent in a ready trace set.

5.1. DEFINITION. (i) Let $X_0, a_0, X_1, a_1, \ldots, a_{n-1}, X_n$ be an alternating sequence of sets X_i $(i = 0,\ldots,n)$ and actions a_i $(i = 0,\ldots,n-1)$ for some $n \geq 0$. Then this sequence is a <u>ready trace</u> if

(1) $a_i \in X_i$ $(i < n)$

(2) $X_0,\ldots,X_{n-1} \subseteq A$

(3) $X_n \subseteq A$ or $X_n = \{\varepsilon\}$ $(\varepsilon \notin A)$.

We recall the notation $(\sigma ; \vec{X}) = (a_0,\ldots,a_{n-1}; X_0,\ldots,X_n)$ for $X_0, a_0, \ldots, X_n$.

(ii) Let $\mathcal{X}$ be a collection of ready traces. Then $\mathcal{X}$ is a <u>ready trace set</u> if $\mathcal{X}$ satisfies the following clauses:

(1) if $(\sigma;\vec{X}) \in \mathcal{X}$ and $(\rho;\vec{Y}) \in \mathcal{X}$, then $X_0 = Y_0$ (<u>root condition</u>)

(2) if $(\sigma;\vec{X})$ is a ready trace and $(\sigma\rho;\vec{X}\vec{Y}) \in \mathcal{X}$ then $(\sigma;\vec{X}) \in \mathcal{X}$ (<u>prefix condition</u>)

(3) if $(a_0,\ldots,a_{k-1}; X_0,\ldots,X_k) \in \mathcal{X}$ and $a_k \in X_k$ $(a_k \neq \varepsilon)$, then $(a_0,\ldots,a_{k-1},a_k; X_0,\ldots,X_k,X_{k+1}) \in \mathcal{X}$ for some X_{k+1} $(\subseteq A$ or $= \{\varepsilon\})$ (<u>continuation condition</u>).

Clearly any $\mathcal{RT}[\![g]\!]$, $g \in \mathbb{H}$ is a ready trace set in this definition. (It holds moreover for $\mathcal{RT}[\![g]\!]$, g being in this paper finite and acyclic, that every ready trace in $\mathcal{RT}[\![g]\!]$ can be continued to one in which the last ready set is $\emptyset$ or $\{\varepsilon\}$.)

Vice versa, we can associate a process graph g_X (in fact a tree) to a ready trace set X as already explained in the proof of Proposition 3.7.

An alternative definition of ready trace set would be one in which the ready traces are allowed to be infinite. Under that definition, the resulting semantics would distinguish processes like $\sum_n a^n$ and $\sum_n a^n + a^\omega$.

It is possible to define the operators considered above directly on these ready trace sets. We will not do that, except for the case of θ, to give a better feeling why θ is compatible with RTS - and not with the coarser semantics as readiness semantics or failure semantics.

5.2. <u>DEFINITION</u>. Let a p.o. $<$ on A_δ be given.

(i) Let $X \subseteq A$. Then $\theta(X)$ is the set of <u>maximal</u> elements (w.r.t. $<$) in X.
If $X = \{\varepsilon\}$, $\theta(X) = X$.
If $\vec{X} = X_0, \ldots, X_n$, then $\theta(\vec{X}) = \theta(X_0), \ldots, \theta(X_n)$.

(ii) Let $(\sigma; \vec{X})$ be a ready trace. Then $(\sigma; \theta(\vec{X}))$ need not be a ready trace, since property (1) of Definition 4.4.1 may be violated. However, $(\sigma; \theta(\vec{X}))$ contains a maximal prefix (in the obvious sense) which is still a ready trace. Now we define

$$\theta(\sigma; \vec{X})$$

to be this maximal prefix ready trace of $(\sigma; \theta(\vec{X}))$.
(<u>Example</u>: if $a < b$ then $\theta(\{a,b\},b,\{a,b\},a,\emptyset) = \{b\},b,\{b\}$.)

(iii) Let X be a ready trace set. Then

$$\theta(X) = \{\ \theta(\sigma; \vec{X})\ |\ (\sigma; \vec{X}) \in X\ \}.$$

Now we claim (without proof) that θ and $\mathcal{RT}[\![\ ]\!]$ commute:

5.3. <u>CLAIM</u>. *Let* $g \in \mathbb{H}$. *Then:*

$$\mathcal{RT}[\![\theta(g)]\!] = \theta(\mathcal{RT}[\![g]\!]).$$

Note that in this explicit definition of the operator θ on ready trace sets it is essential to have the intermediate ready sets in a ready trace available.

6. CONCLUDING REMARKS

We have considered a process semantics RTS, ready trace semantics (called in PNUELI [8] 'barbed semantics'), which is intermediate between bisimulation semantics (BS) and readiness semantics (RS). The advantage of RTS above RS, FS is that it allows the presence of operators that may be important for process specification, such as the priority operator θ.

This in contrast with RS, FS which reject operators like θ, since in these semantics too many processes are identified to bear the presence of θ. This seems to be a general phenomenon: the finer the process equivalence, the more operators on processes (like θ) can be defined. Adding more equations, i.e. making the process equivalence coarser, increases the ease of process verifications, but at the cost of losing specification possibilities by means of operators as θ, which become undefinable.

From an intuitive point of view (see Remark 2.9) a semantics as RTS seems perfectly natural.

An evident direction for further work is the extension of the above to infinite processes, and to silent moves.*)

---

*) R. van Glabbeek has informed us that there is a neat complete axiomatisation of RTS for finite processes with τ-steps, however, without the priority operator θ.

REFERENCES

[1] BAETEN, J.C.M., BERGSTRA, J.A. & J.W. KLOP,
Syntax and defining equations for an interrupt mechanism in process algebra, CWI Report CS-R8503, Amsterdam 1985.

[2] BERGSTRA, J.A., KLOP, J.W. & E.-R. OLDEROG,
Readies and failures in the Algebra of Communicating Processes, CWI Report CS-R85.., Amsterdam 1985.

[3] BERGSTRA, J.A., KLOP, J.W. & E.-R. OLDEROG,
Failure semantics with fair abstraction, CWI Report CS-R85.., Amsterdam 1985.

[4] BROOKES, S.D.,
On the relationship of CCS and CSP, in: Proc. 10th ICALP, Barcelona, (J. Díaz, Ed.), Springer LNCS No.154, 83-96, 1983.

[5] BROOKES, S., HOARE, C. & W. ROSCOE,
A Theory of Communicating Sequential Processes, J. Assoc. Comput. Mach.31, No.3, p.560-599,1984.

[6] MILNER, R.,
A Calculus of Communicating Systems, Springer LNCS 92, 1980.

[7] OLDEROG, E.-R. & C.A.R. HOARE,
Specification-oriented semantics for communicating processes, in: Proc. 10th ICALP, Barcelona, (J. Díaz, Ed.), Springer LNCS 154, 1983. Expanded version: Technical Monograph PRG-37, Oxford Univ. Comput. Lab., 1984.

[8] PNUELI, A.,
Linear and branching structures in the semantics and logics of reactive systems,
in: Proc. 12th ICALP, Nafplion, (W. Brauer, Ed.), Springer LNCS 194, 1985, p.15-32.