Logic Colloquium '86
F.R. Drake and J.K. Truss (Editors)

A Complete Inference System for Regular Processes with Silent Moves

J.A.Bergstra
Department of Computer Science, University of Amsterdam,
Kruislaan 409, 1098 SJ Amsterdam;
Department of Philosophy, State University of Utrecht,
Heidelberglaan 2, 3584 CS Utrecht

J.W.Klop
Centre for Mathematics and Computer Science,
Kruislaan 413, 1098 SJ Amsterdam;
Department of Mathematics and Computer Science, Free University,
de Boelelaan 1081, 1081 HV Amsterdam.

We study the notion of bisimulation between process graphs with silent or invisible steps (τ-steps). This leads to a normalisation or minimalisation result for regular processes with τ-steps and subject to operations + (alternative composition), · (sequential composition) and || (parallel composition or free merge), thereby answering a question of MILNER [10] and proving the consistency of a version of Koomen's fair abstraction rule.

Note: This work was sponsored in part by ESPRIT contract 432: Meteor.

Introduction

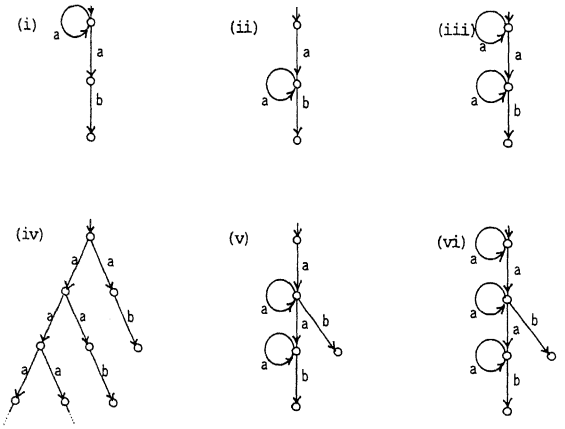
In 'classical' automata theory, the process behaviour of a system or agent which is capable of performing certain 'steps' or atomic actions, is taken to be a set of execution traces. An execution trace is a linear sequence of atomic actions, finite or infinite. In recent years it has become increasingly evident that for several applications such an abstract view is too abstract, i.e. one needs more information from a system than merely the set of possible execution traces. This is especially the case where *communicating* systems are concerned; and this consideration led Milner to his well-known CCS, Calculus of Communicating Systems (see MILNER [9]). Here processes are obtained as process graphs (transition diagrams) modulo some notion of observational equivalence or *bisimulation*.

The present paper also considers processes in bisimulation semantics, although the feature of communication is not considered here. It is one of a series about 'process algebra', including [1-5], where several process axiomatisations, each capturing certain process features, have been developed and studied. (The paper can be read independently, though.) One of the goals of process algebra is to give a uniform axiomatic framework for algebraic process theories such as CCS, by means of (mostly) equational logic. More specifically, process algebra allows one to provide a very detailed mechanical description of, say, a communication protocol, along the following lines.

Let P be the process that describes the entire mechanism of a given protocol. Now imagine that Q, the so-called server specification, describes the intended behaviour of the protocol as seen from outside. In many cases it is correct to assume that Q is also a process that can be recursively specified in process algebra. The difference between P and Q is that P takes into account many internal steps of the protocol. Now the situation that P *implements* Q is in terms of process algebra expressed by the equation $Q = \tau_I(P)$ where τ_I abstracts, i.e. replaces by τ, all internal steps of the

protocol. Here τ is Milner's silent step. Verification of correctness of the implementation then boils down to a proof that $Q = \tau_I(P)$ in some suitable model of process algebra. For some protocol verifications of this sort we refer to [5,8,15]. This application of protocol verification may be sufficient to suggest that the phenomenon of abstraction is vital in theories about processes. In this paper we investigate this matter in the bisimulation model under the additional and realistic assumption that P and Q are finite state (regular) processes. Indeed a complete proof system is given that allows to derive all true identities $P_1 = P_2$ between regular processes involving silent steps.

We will now discuss the contents of this paper in more detail. As said, we study regular processes, i.e. processes having finitely many subprocesses ('states'). Such processes can be obtained as equivalence classes of labeled transition diagrams, or as we will call them, process graphs. The equivalence relation used here is that of bisimulation or observational equivalence (congruence) as in MILNER's CCS ([9]). This means that processes are not equated as soon as their trace sets coincide: the notion of bisimulation is more refined and takes also account of the timing of the various alternative choices in a process. This is the crucial difference with classical automata theory. As an example, consider the following six process graphs:



All have trace set $a^*ab \cup \{a^\omega\}$. However, no horizontal pair of process graphs is bisimilar – e.g. (i), (ii) differ since in (ii) after one or more a-steps always a b-step is possible. On the other hand all vertical pairs are bisimilar.

This paper has been written in an attempt to answer a question in MILNER [10]: there a complete inference system is given for regular processes without τ-steps (Milner's system M as it is called below, in Section 5). It is asked (p. 459 in [10]) whether joining Milner's τ-laws to M results in a complete proof system for regular processes with τ-steps. We answer the question negatively but formulate additional proof principles which yield a complete system. (For some remarks on an alternative complete proof system recently given by Milner, see the end of this

Introduction.)

Summary

In the first section (see 'Contents' below) several domains of process graphs are introduced as well as three notions of bisimulation: $\leftrightarrows$, $\leftrightarrows_\tau$ and $\leftrightarrows_{r\tau}$. The first two correspond for finite process trees (and also for finite process graphs) to Milner's strong equivalence, respectively observational equivalence (see Section 6.1). The third one corresponds to Milner's observational congruence.

notation	name	for finite process graphs the same as
$\leftrightarrows$	bisimulation	strong equivalence
$\leftrightarrows_\tau$	τ-bisimulation	observational equivalence
$\leftrightarrows_{r\tau}$	rooted τ-bisimulation	observational congruence

After making some elementary observations concerning these bisimulations, in Section 2 the notion of $\leftrightarrows_{r\tau}$ is analysed leading to the main lemma (Corollary 2.4) used in the completeness proof of Section 6.

In Section 3 a normalisation result is proved which for the simple case of ordinary bisimilarity ($\leftrightarrows$) is used in the completeness proof of Section 5.

In Section 5 Milner's complete inference system M for regular processes without τ-steps in reviewed, as it is a basis for our completeness result in Section 6 where τ-steps are present. Also in Section 5 we formulate a proof system BPA_{LR} which is equivalent to M except that μ-expressions are expressed as systems of recursion equations, which we need for the extended proof system $BPA_{\tau LR}$ in Section 6. The formulation of this system is complicated due to the fact that recursion equations which are guarded by atoms some of which may be τ, need not have unique solutions (as ordinary guarded recursion equations do). Albeit in a rather different formalism, a similar observation is made by HOARE [13], where he notes in Section 3.5.5 that 'unfortunately the introduction of hiding permits the construction of recursion equations which do not have a unique solution'. In fact, it is not very surprising that the recursion equation

$$X = a + \tau X \qquad (*)$$

turns out to have infinitely many solutions, if one realizes that one of the τ-laws to which τ is subject states that $\tau x = \tau x + x$. For, (*) thus is the same as

$$X = a + \tau X + X$$

where the unguarded occurrence of X invites many unintended solutions. This matter is investigated in Section 7: there the general solution of recursion equations, in which some guards may be τ, is determined. (For (*) it is: $X = \tau(a + q)$, q arbitrary.) Also a criterion, which is necessary and sufficient, is given for recursion equations (or systems of them) guaranteeing unique solvability. The criterion is simply that no "τ-cycle" should occur in such a system of recursion equations. (Above, (*) has a τ-cycle from X to X.) Thus, e.g. the system $\{X = a+bY, Y = c + \tau X\}$ has a unique solution.

Finally, in Section 8 the merge operator ($\|$) without communication is added to the earlier proof system; the completeness is carried over easily.

We conclude this introduction with some remarks about *bisimulation versus trace equivalence*, in an attempt to answer the question why one does not simply stick to the 'easier' notion of trace equivalence (i.e. two processes are trace equivalent if the corresponding sets of traces coincide).

Suppose a class C of 'primary' objects p is given on which an equivalence relation E is defined reflecting our view on what the 'essence' of the primary objects is. The equivalence classes p/E are the *intended* objects of our universe of discourse. Now operations F_i ($i \in I$) are defined, on the representations of the intended objects, i.e. on C. A desirable state of affairs would be one in which E is a *congruence* w.r.t. the F_i; for this means in effect that we have defined operations F_i/E on the class of intended objects C/E. Otherwise, we must conclude that E 'abstracts too much' of the primary objects and a finer equivalence E' is called for. Let us call, in the case of the desirable state of affairs, the equivalence E *adequate for the operations* F_i.

In the present case, the primary objects are the process graphs; and the operations are +, ·, merge without communication ('free' merge), merge with communication, merge with or without communication in the presence of τ-steps. For E the candidates are: trace equivalence (if τ's are present, 'external' trace equivalence $=_\tau$ as defined below), and $\leftrightarrows$, $\leftrightarrows_\tau$, $\leftrightarrows_{r\tau}$ as above. All equivalences are adequate for +, ·, alternative and sequential composition. However, when the *parallel* composition operators (i.e. the various merges) are considered, the notions of equivalence separate:

(i) trace equivalence is adequate for free merge, also in the presence of τ-steps – but *not* for merge with communication as in CCS or ACP;
(ii) bisimulation ($\leftrightarrows$) is adequate for merge with communication;
(iii) τ-bisimulation ($\leftrightarrows_\tau$) is *almost* (but not quite) adequate for merge with communication plus τ-steps as in CCS or ACP_τ;
(iv) rτ-bisimulation ($\leftrightarrows_{r\tau}$) is adequate for merge with communication plus τ-steps.

It should be remarked that in this paper we do not deal with communication; a development including also communication seems possible along similar lines, though. But also for the communication-less case, bisimulation (and its variants) is in our opinion a fundamental notion with a clear elegance and a strong justification.

Addendum. After completion of the report version of this paper (Report CS-R8420, Centre for Mathematics and Computer Science, Amsterdam), MILNER [12] discovered an elegant complete axiom system for regular processes with τ-steps which is simpler than the one we give in Section 6; it is formulated in terms of μ-calculus, like Milner's system M for the τ-less case (see Section 5.2). Apart from the usual axioms for recursion such as μ1,μ2 (see Table 1 in Section 5.2), the 'critical' axioms are:

$$\mu X(\tau X + E) = \mu X(\tau E)$$
$$\mu X(\tau(X + E) + F) = \mu X(\tau X + E + F)$$

where E,F are arbitrary expressions. Using these axioms each expression (having a corresponding

process graph containing possibly τ-cycles) can be proved equal to one whose corresponding graph has no τ-cycles. Roughly, these axioms 'replace' our rule R4 in Table 9.

Contents

1. Process graphs and bisimulations

We will start with the preliminaries of introducing some *domains of process graphs* as well as (several variants of) the semantical notion of equivalence between process graphs called *bisimulation*. Furthermore we will state in this section several elementary observations concerning bisimulation.

1.1. Graph domains

First some standard terminology. All graphs considered in this paper are *connected, rooted multi-digraphs,* that is: the graph has a root ('starting node'), the edges between the nodes are directed and between two nodes there may be several edges, and every node is accessible from the root. The *out-degree* of a node s in graph g is the number of edges starting from s. Likewise the in-degree is defined. If the out-degree of s is zero, s is an end-point or *end-node*. A *path* π in g is an alternating sequence of nodes and edges:

$$\pi: s_0 \rightarrow s_1 \rightarrow .... \rightarrow s_k$$

for k≥0. The *length* of the path is k, its number of edges. If k≥1 and s_0 and s_k coincide, π is a *cycle*. In particular, if k=1, and s_0,s_1 coincide, π is a *loop*. If s is lying on a cycle, it is called *cyclic*, otherwise *acyclic*.

A graph g is *finitely branching* if the out-degree of every s ∈ NODES(g), the set of nodes of g, is a natural number. A graph is finite if NODES(g) and EDGES(g) are finite. The graph g is a *tree* if all its nodes are acyclic and the in-degree of every node is ≤1 (viz. 0 for the root and 1 for the other nodes).

If s is a node of g, the *subgraph* $(g)_s$ of g is the graph with root s and all the nodes and edges accessible from s in the obvious way.

Graphs differing only in their naming of the nodes are considered to be identical.

The graphs considered so far, are the underlying structure for *process graphs*: these are graphs labeled with *atomic actions* from the alphabet A_τ. Here $A_\tau = \{\tau, a, b, c, ...\}$ is a finite set containing a special element τ (the *'silent'* action). We use the variables a,b,c for $A_\tau - \{\tau\}$, and u,v,... for A_τ. (The finiteness of A_τ is in this paper not important.) Intuitively, a process graph like in Figure 1, is a process capable of performing sequences of actions ('execution traces') given by the various paths starting from the root. For instance, ababab..., or abτττ... .

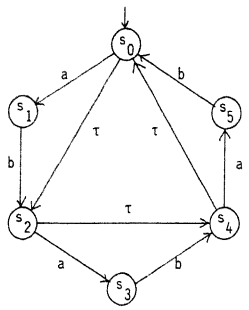


Figure 1

The process graph in Figure 1 contains *labeled paths* as e.g. π: $s_0 \to^a s_1 \to^b s_2 \to^\tau s_4$ also written as π: $s_0 \to^{ab\tau} s_4$. Here $ab\tau \in A_\tau^*$, the set of finite words over A_τ, including the empty word ε.

The crucial difference with 'classical' automata theory is that we are not exclusively interested in the set of execution traces determined by a process graph, as explained in the introduction.

Now we have the following process graph domains: $\mathbb{G}^\kappa{}_{A\tau}$, the set of finitely branching process graphs g over the alphabet A_τ such that the cardinality of NODES(g) does not exceed the cardinal number κ. Mostly, we will drop the subscript A_τ and moreover we will be only interested in the case $\kappa = \omega$, and write $\mathbb{G}$ instead of $\mathbb{G}^\omega$. $\mathbb{T}$ is the subdomain of $\mathbb{G}$ consisting of *process trees*. $\mathbb{R}$ is the domain of *finite* process graphs, also called *regular* process graphs. $\mathbb{F} = \mathbb{T} \cap \mathbb{R}$ is the set of finite process trees. $\mathbb{G}^\rho$ is the set of finitely branching process graphs with *acyclic root*. Likewise $\mathbb{R}^\rho = \mathbb{R} \cap \mathbb{G}^\rho$. (See Figure 2.)

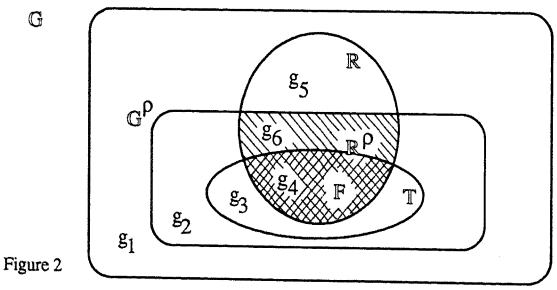


Figure 2

EXAMPLE 1.1.1. The following process graphs have a position in Figure 2 as indicated:

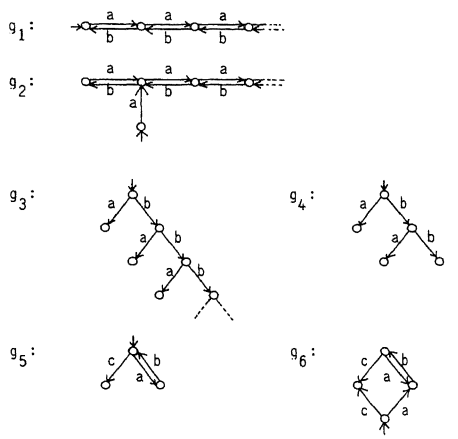


Figure 3

1.2. Root-unwinding

It will be convenient to have a canonical transformation of a process graph $g \in \mathbb{G}$ into an 'equivalent' root-acyclic one in $\mathbb{G}\rho$. (Here 'equivalent' is in a sense which will be explained below, in Proposition 1.3.1.2.)

DEFINITION 1.2.1. The map $\rho: \mathbb{G} \to \mathbb{G}$, *root-unwinding*, is defined as follows. Let $g \in \mathbb{G}$ have root r; then $\rho(g)$ is defined by the clauses

(i) $\mathrm{NODES}(\rho(g)) = \mathrm{NODES}(g) \cup \{r'\}$ where r' is a 'fresh' node;

(ii) the root of $\rho(g)$ is r';

(iii) $\mathrm{EDGES}(\rho(g)) = \mathrm{EDGES}(g) \cup \{r' \to^u s \mid r \to^u s \in \mathrm{EDGES}(g)\}$;

(iv) nodes and edges which are inaccessible from the new root r' are discarded.

EXAMPLE 1.2.2. (i) In Example 1.1.1 (Figure 3) we have $\rho(g_1) = g_2$ and $\rho(g_5) = g_6$. Further, $\rho(g_3) = g_3$ and $\rho(g_4) = g_4$. E.g. $\rho(g_4)$ is obtained as in Figure 4:

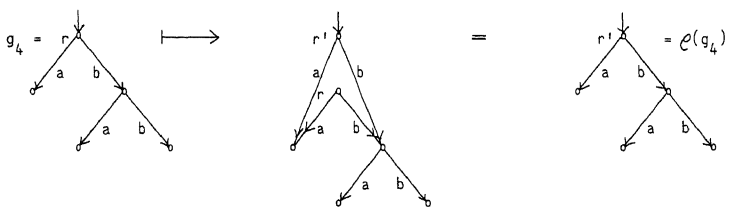


Figure 4

Note that ρ is idempotent: $\rho^2(g) = \rho(g)$.

1.3. Bisimulations

On the process graphs considered so far we have a semantical notion of equivalence, called *bisimulation*. The original notion is from PARK [14]; it was anticipated by Milner's *observational equivalence* (see [9]) which has a more involved definition but coincides (on the set of finite process graphs, not for infinite ones) with bisimulation. We consider three variants:

$\leftrightarrows$ *bisimulation* (on $\mathbb{G} \times \mathbb{G}$)

$\leftrightarrows_\tau$ *τ–bisimulation* (on $\mathbb{G} \times \mathbb{G}$)

$\leftrightarrows_{r\tau}$ *rooted τ–bisimulation* (on $\mathbb{G}^\rho \times \mathbb{G}^\rho$).

1.3.1. Bisimulation: $\leftrightarrows$

Let $g,h \in \mathbb{G}$. The relation $R \subseteq \text{NODES}(g) \times \text{NODES}(h)$ is a *bisimulation from* g *to* h, notation $R: g \leftrightarrows h$, if

(i) Domain(R) = NODES(g) and Range(R) = NODES(h)

(ii) (ROOT(g), ROOT(h)) $\in$ R

(iii) if $(s,t) \in R$ and $s \rightarrow^u s' \in \text{EDGES}(g)$ then there is an edge $t \rightarrow^u t' \in \text{EDGES}(h)$, such that $(s',t') \in R$. (See Figure 5.)

(iv) if $(s,t) \in R$ and $t \rightarrow^u t' \in \text{EDGES}(h)$ then there is an edge $s \rightarrow^u s' \in \text{EDGES}(g)$, such that $(s',t') \in R$. (See Figure 5.)

(In Figure 5 the dotted lines have the usual 'existential meaning.)

Further, we write $g \leftrightarrows h$ if $\exists R\ R: g \leftrightarrows h$. In this case g,h are called *bisimilar*.

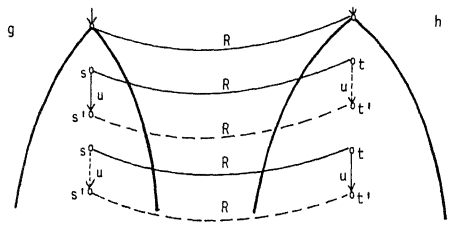


Figure 5

EXAMPLE 1.3.1.1. (i) See Figure 6; the curved lines denote the bisimulation.
(ii) The graphs in Figure 7 (a) (b) are not bisimilar.

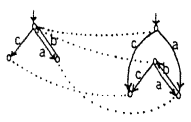


Figure 6

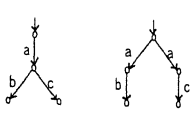


Figure 7 (a) (b)

It is easy to see that bisimilar process graphs have the same sets of traces. The reverse, however, does not hold as Example 1.3.1.1 (ii) shows. The following facts are easily proved:

PROPOSITION 1.3.1.2.

(i) *Let* $g \in \mathbb{G}$. *Then* $g \leftrightarrows \rho(g)$.

(ii) *The relation* $\leftrightarrows$ *(bisimilarity) is an equivalence relation on* $\mathbb{G}$.

(iii) *If* $g,h \in \mathbb{G}$, R: $g \leftrightarrows h$ *and for* $s \in$ NODES(g), $t \in$ NODES(h) *we have* $(s,t) \in R$, *then* R': $(g)_s \leftrightarrows (h)_t$, *where* R' *is the restriction of* R *to the nodes of* $(g)_s$ *and* $(h)_t$.

1.3.2. τ-Bisimulation: $\leftrightarrows_\tau$

Note that an equivalent definition for ordinary bisimulation can be given as follows: replace in the definition of 1.3.1 clauses (iii), (iv) by:

(iii)' if $(s,t) \in R$ and π: $s \to^w s'$ is a path in g (determining the 'word' $u_1u_2...u_k$ $(k \geq 0)$ of labels along the edges in π), then there is a path π': $t \to^{w'} t'$ in h such that $(s',t') \in R$ and such that $w \equiv w'$ (w,w' are identical).

(iv)' likewise with the role of g,h interchanged.

The definition of $\leftrightarrows_\tau$ now parallels that for $\leftrightarrows$, with as only alteration that $w \equiv w'$ is replaced by $w \equiv_\tau w'$. Here $w \equiv_\tau w'$ ($w,w' \in A_t^*$ are equivalent modulo τ) if w,w' are identical after deletion of τ's. E.g. $\tau \equiv_\tau \tau\tau\tau \equiv_\tau \varepsilon$ (the empty word); $ab\tau\tau\tau c\tau \equiv_\tau \tau a\tau b\tau c$. Processes $g,h \in \mathbb{G}$ such that $g \leftrightarrows_\tau h$ are called τ*–bisimilar*.

1.3.3. Rooted τ-bisimulation: $\leftrightarrows_{r\tau}$

Suppose $g,h \in \mathbb{G}^\rho$ and R: $g \leftrightarrows_\tau h$ in such a way that

$(s,t) \in R \Rightarrow$ $s = \text{ROOT}(g)$ and $t = \text{ROOT}(h)$, or:

$s \neq \text{ROOT}(g)$ and $t \neq \text{ROOT}(h)$.

(So a non-root cannot be related in the bisimulation to a root.) Then R is called a *rooted* τ*–bisimulation* between g,h and we write

R: $g \leftrightarrows_{r\tau} h$.

Such g,h are called $r\tau$*–bisimilar* (via R). Note that $g \leftrightarrows h \Rightarrow g \leftrightarrows_\tau h$ and $g \leftrightarrows_{r\tau} h \Rightarrow g \leftrightarrows_\tau h$. As before, $\leftrightarrows_{r\tau}$ and $\leftrightarrows_\tau$ are equivalence relations on $\mathbb{G}^\rho$ and $\mathbb{G}$, respectively. Also $\leftrightarrows_\tau$, $\leftrightarrows_{r\tau}$ are invariant under ρ (root-unwinding).

EXAMPLES 1.3.4.

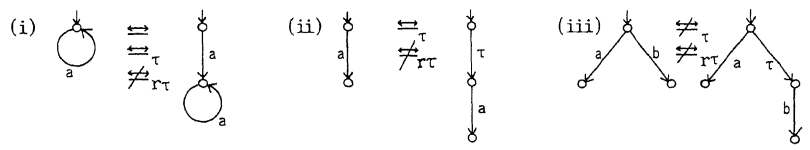


Figure 8

Some further obvious facts are:

PROPOSITION 1.3.5. (i) *Let* g,h ∈ G *be* τ*-bisimilar via* R. *Let* (s,t) ∈ R. *Then* $(g)_s$ *and* $(h)_t$ *are* τ*-bisimilar (via the appropriate restriction of* R*). (The nodes* s,t *are called in this case* τ*-bisimilar.)*
(ii) *Let* g,h ∈ $\mathbb{G}^\rho$ *and* $R: g \leftrightarrows_{r\tau} h$. *Let* (s,t) ∈ R. *Then* $(g)_s \leftrightarrows_\tau (h)_t$ *(in general not* rτ*-bisimilar).* □

PROPOSITION 1.3.6. *Let* g,h ∈ G *and suppose R:* g ≙ h *as well as R':* g ≙ h. *Then* R∪R': g ≙ h. *Similar for* $\leftrightarrows_\tau$ *and* $\leftrightarrows_{r\tau}$. □
(Note that the *intersection* of bisimulations R,R' need not be a bisimulation.)

DEFINITION 1.3.7. (i) A τ*-cycle* in a process graph g is a cycle
$$\pi: s_0 \to^\tau s_1 \to^\tau \ldots \to^\tau s_k \equiv s_0 \quad (k \geq 1).$$
(ii) A τ*–loop* is a τ-cycle of length 1:
$$\pi: s_0 \to^\tau s_0.$$

REMARK 1.3.8. The reason for restricting the notion of rτ-bisimulation to $\mathbb{G}^\rho$ (root-acyclic process graphs) is as follows. Consider the three graphs in Figure 9.

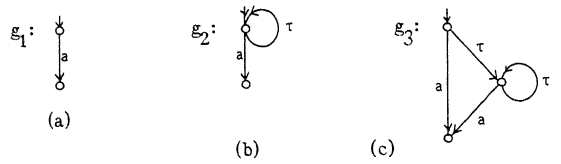


Figure 9

In view of later developments below we want *not* $g_1 \leftrightarrows_{r\tau} g_2$ but we do want $g_2 \leftrightarrows_{r\tau} g_3$ (= $\rho(g_2)$). A definition of $\leftrightarrows_{r\tau}$ on all $\mathbb{G}$ with these properties would be more involved. Also, the restriction to $\mathbb{G}^\rho$ makes sense since $\leftrightarrows_{r\tau}$ will prove to be a *congruence* on $\mathbb{G}^\rho$ w.r.t. the operations $+, \cdot, \|, \lfloor\!\lfloor$ defined in Section 4; and + is most naturally defined on $\mathbb{G}^\rho$. (On the other hand, $\leftrightarrows_\tau$ is not a congruence w.r.t. +; cf. our discussion in 6.1.)

PROPOSITION 1.3.9. *Let* g ∈ $\mathbb{G}$ *contain a* τ*–cycle passing through the nodes* s,t. *Then* s,t *are* τ*-bisimilar (i.e.* $(g)_s \leftrightarrows_\tau (g)_t$*).*

PROOF. (See Figure 10.) Note that every point in g accessible form s is accessible from t and vice versa. Hence the node sets of $(g)_s$ and $(g)_t$ coincide. Now let Id be the identity relation on NODES($(g)_s$). Then it is easy to verify that Id ∪ {(s,t)} is a τ-bisimulation from $(g)_s$ to $(g)_t$. □

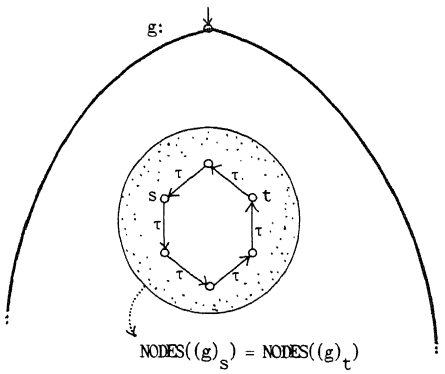


Figure 10

PROPOSITION 1.3.10. (i) *Let* g ∈ 𝔾 *contain* τ*-bisimilar nodes* s,t. *Let* $g_{\tau(s,t)}$ *be the result of adding a* τ*-edge from* s *to* t. *Then* g *and* $g_{\tau(s,t)}$ *are* τ*-bisimilar.*
(ii) *Let* g ∈ $\mathbb{G}^{\rho}$ *contain non-root nodes* s,t *which are* τ*-bisimilar. Then* g $\leftrightarrows_{r\tau}$ $g_{\tau(s,t)}$.

PROOF. (i) Let Id be the identity relation on NODES(g) (= NODES($g_{\tau(s,t)}$)). Then Id ∪ {(s,t)} is a required τ-bisimulation from g to $g_{\tau(s,t)}$. (ii) Similar. □

This proposition says that adding τ-steps between τ-bisimilar nodes in a graph g does not change the "τ-bisimilarity character" of g (and for the same reason, of any node q, or better, subgraph $(g)_q$ of g). Here the τ-bisimilarity character of g is the class of all g' ∈ 𝔾 which are τ-bisimilar with g. In particular, the τ-bisimilarity character is not disturbed by appending τ-loops to nodes of g. Vice versa, removing τ-loops also does not change the τ-bisimilarity character.

EXAMPLE 1.3.10.1.

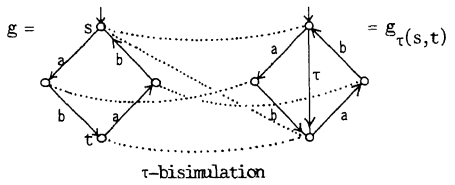


Figure 11

Just as all τ-loops can be removed from g without changing τ-bisimilarity (which follows from the previous proposition, by taking s = t), it is possible to remove all τ-cycles from g. We need a definition first:

DEFINITION 1.3.11. Let g ∈ 𝔾 contain nodes s,t. Then $g_{id(s,t)}$ is the process graph resulting

from the identification of s and t, in the obvious sense.

EXAMPLE 1.3.11.1. Let g be as in Figure 11. Then $g_{id(s,t)}$ is:

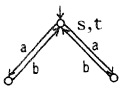


Figure 12

PROPOSITION 1.3.12. (i) *Let* $g \in \mathbb{G}$ *and suppose* $s,t \in$ NODES(g) *are* τ*-bisimilar. Then* g *and* $g_{id(s,t)}$ *are* τ*-bisimilar.*
(ii) *Let* $g \in \mathbb{G}^{\rho}$ *and suppose the non-root nodes* $s,t \in$ NODES(g) *are* τ*-bisimilar. Then* $g \leftrightarrows_{r\tau} g_{id(s,t)}$.

PROOF. Obvious. □

COROLLARY 1.3.13. (i) *Every* $g \in \mathbb{G}$ *is* τ*-bisimilar with some* $g' \in \mathbb{G}$ *without* τ*-cycles.*
(ii) *Every* $g \in \mathbb{G}^{\rho}$ *is* $r\tau$*-bisimilar with some* $g' \in \mathbb{G}^{\rho}$ *without* τ*-cycles.*
(iii) *Every* $g \in \mathbb{R}$ *is* τ*-bisimilar to some* $g' \in \mathbb{R}$ *without infinite* τ*-paths.*

PROOF. Follows from considering Figure 13. □

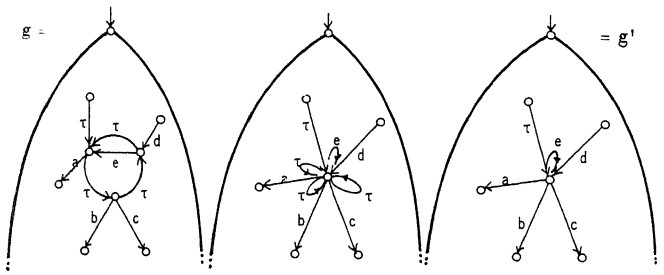


Figure 13

We conclude this section with a proposition needed in the sequel, which illuminates the difference between $\leftrightarrows_{\tau}$ and $\leftrightarrows_{r\tau}$.

PROPOSITION 1.3.14. . *Let* $g,h \in \mathbb{G}$ *and let* τg, τh *be the result of prefixing a* τ*-step. Then:* $g \leftrightarrows_{\tau} h \Leftrightarrow \tau g \leftrightarrows_{r\tau} \tau h$.

PROOF. ($\Rightarrow$) is trivial. ($\Leftarrow$): Let r,r' be the roots of τg, τh and let s,s' be the roots of g,h. Let R be an $r\tau$-bisimulation between τg and τh (see Figure 14). Then $R' = (R \cup \{(s,s')\}) - \{(r,r')\}$ is a τ-bisimulation between g,h. □

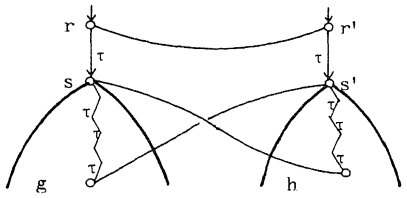


Figure 14

2. An analysis of rτ-bisimulation

The main result of this section is that an rτ-bisimulation R between g,h ∈ $\mathbb{G}$ can be analysed into more simple parts:

$$\begin{array}{ccc} g & \leftrightarrows_{r\tau} & h \\ \downarrow & & \downarrow \\ \Delta(g) & & \Delta(h) \\ \downarrow & & \downarrow \\ E(\Delta(g)) & \leftrightarrows & E(\Delta(h)) \end{array}$$

(Corollary 2.4). I.e. $g \leftrightarrows_{r\tau} h$ iff g,h after 'preprocessing' (by means of some simple operations Δ,E: $\mathbb{G} \to \mathbb{G}$), are bisimilar in the ordinary sense where τ does not play its special role. This analysis is the basis for the completeness theorem in Section 6 where syntax and axioms are given describing rτ-bisimulation.

In Section 3, the present analysis will be connected to a normalisation procedure for process graphs with τ-steps.

2.1. The operation Δ

First we need some terminology: if g ∈ $\mathbb{G}$, then an *arc* in g is a part of the form (a) in Figure 15 (here u ∈ A_τ). In case n = m = 0, the arc is a *double edge* as in (b). Other special cases are in Figure 15 (c), (d): these are called Δ-*arcs*. It is not required that the three nodes displayed in (a)–(d) are indeed pairwise different. The u-step between nodes s,t is called the *primary* edge of the arc.

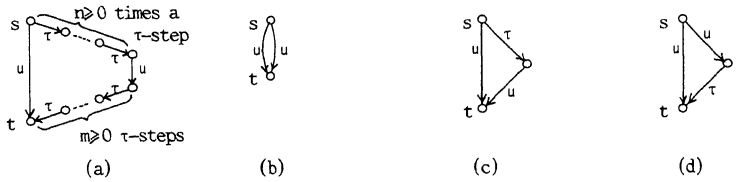


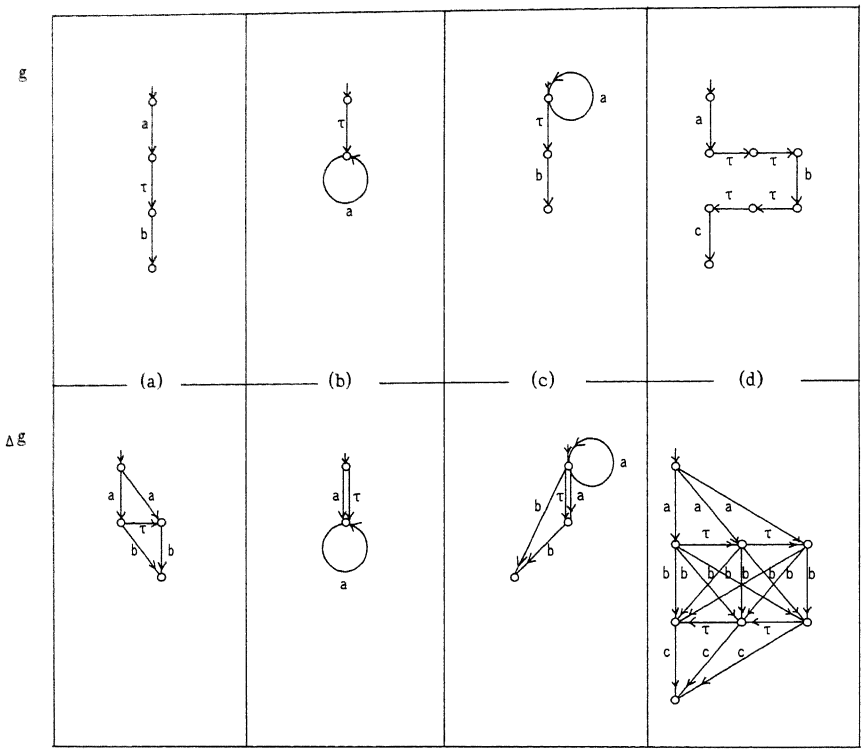
Figure 15

Now the operation Δ: $\mathbb{G} \to \mathbb{G}$ is defined as follows: whenever g ∈ $\mathbb{G}$ contains a path $s_1 \to^\tau s_2 \to^u s_3$ (where s_1,s_2,s_3 need not be pairwise different), an edge $s_1 \to^u s_3$ is added if not

yet present. Likewise for every path $s_1 \to^u s_2 \to^\tau s_3$. $\Delta(g)$ is the result of this completion of g with edges as indicated.

Further, we say that $g \in \mathbb{G}$ is Δ-*saturated* if $\Delta(g) = g$.

EXAMPLE 2.1.1. Figure 16



PROPOSITION 2.1.2. (i) $\Delta(g) \leftrightarrows_\tau g$ *if* $g \in \mathbb{G}$; (ii) $\Delta(g) \leftrightarrows_{r\tau} g$ *if* $g \in \mathbb{G}^\rho$.

PROOF. The identity relation R gives a (r)τ-bisimulation. □

2.2. The operation E

Call a node of $g \in \mathbb{G}^\rho$ *internal* if it is not the root, and an edge of g *internal* if it is between internal nodes. Further, call an internal τ-step $s \to^\tau t$ in $g \in \mathbb{G}^\rho$ an ε-*step* if s,t are τ-bisimilar. Finally, consider the set of internal nodes of $g \in \mathbb{G}^\rho$ and the equivalence relation on this set given by τ-bisimilarity. We will call the equivalence classes: *clusters*. So ε-steps always occur 'inside' a cluster (see Figure 17).

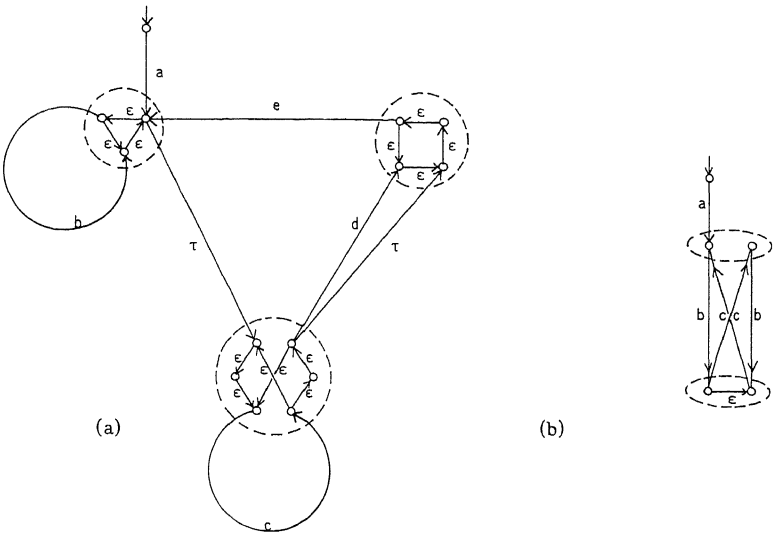


Figure 17
(Clusters are indicated with ◯.)

NOTATION. If s,s' are in the same cluster we write also s ~ s'.

The concept of clusters of nodes makes the structure of a process graph more perspicuous – of course, it also anticipates the normalisation procedure in Section 3, where in essence the clusters are collapsed to single nodes.

In particular, Δ-saturated process graphs g have a local structure as indicated in Figure 18:

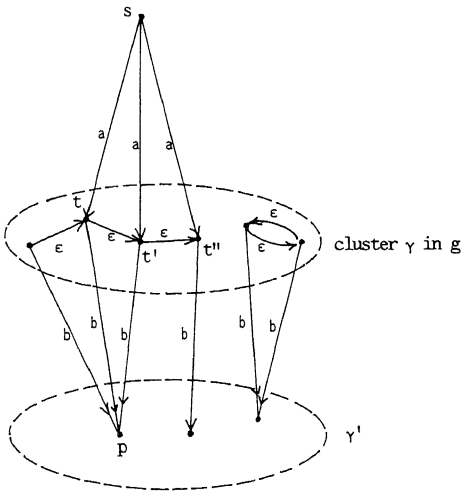


Figure 18

Namely, if γ is a cluster in g and $s \to^a t$ is an 'incoming' edge, then the endpoint t is carried in the direction of the ε-steps, thus providing arrows $s \to^a t'$, $s \to^a t''$. Vice versa, if $t' \to^b p$ is an outgoing edge, the starting point t' is carried backwards along ε-paths. This is a simple consequence of Δ-saturation and in fact it does not depend on the particular nature of ε-steps. Moreover, and this does depend on the definition of cluster in terms of $\leftrightarrows_\tau$, if γ has an outgoing edge $\to^b$ to some cluster γ', then from every point in γ there is an edge $\to^b$ to γ'. We will need this last fact so let us prove it:

PROPOSITION 2.2.1. *Let* $g \in \mathbb{G}^\rho$ *be* Δ*-saturated. Let* $s \to^u t$ *be an edge of* g *and let* $s' \sim s$. *Then* g *contains an edge* $s' \to^u t'$ *for some* $t' \sim t$.

PROOF. Consider an $r\tau$-bisimulation R of g with itself relating s to s'. (R can be taken to be the union of the identity relation on g and a τ-bisimulation from $(g)_s$ to $(g)_{s'}$.) Now by definition of τ-bisimulation, given the edge $s \to^u t$ and $s \sim s'$ there is a path π: $s' \twoheadrightarrow t'$ with label $\tau^n u \tau^m$ in g for some $n,m \geq 0$ and some t' with $t' \sim t$. By virtue of Δ-saturation, we now have an edge $s' \to^u t'$. □

Now we would like, in order to obtain the 'structure theorem' 2.4 concerning $(r)\tau$-bisimulation as well as the completeness result in Section 6, to omit all ε-steps in a Δ-saturated graph g, resulting in a graph g' which is still $r\tau$-bisimilar to g. Here the need for Δ-saturation comes in, for omitting ε-steps could make a non-Δ-saturated graph g disconnected, as in Example 2.1.1 (a): there the τ-step in g (which clearly is an ε-step) cannot be removed, but it can in $\Delta(g)$.

DEFINITION 2.2.2. E is the operation from $\mathbb{G}^\rho$ to $\mathbb{G}^\rho$ which removes in $g \in \mathbb{G}^\rho$ all ε-steps (as well as parts of g which become disconnected in that process). If $g = E(g)$, g is called *prenormal*.

PROPOSITION 2.2.3. E *preserves* Δ*-saturation.*

PROOF. Suppose $g \in \mathbb{G}$ is Δ-saturated. The only possibility for E(g) to have lost the property of Δ-saturation, is that one of the removed ε-edges was the primary edge of a Δ-arc:

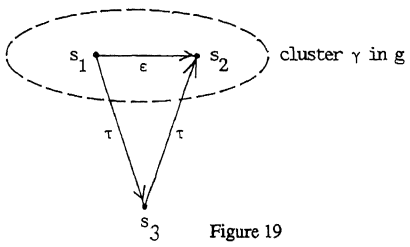


Figure 19

This is impossible, since then s_3 is τ-bisimilar to s_1,s_2. To see this, note that adding an edge $s_2 \to^\tau s_1$ does not change the τ–bisimilarity character of any point in g (Proposition 1.3.10). But then s_1,s_2,s_3 are τ-cyclic, hence τ–bisimilar (Proposition 1.3.9). This means that the τ-steps

$s_1 \to^{\tau} s_3$, $s_3 \to^{\tau} s_2$ are in fact ε-steps in g. However, then E would have removed them. □

PROPOSITION 2.2.4. (i) *If* $g \in \mathbb{G}^{\rho}$ *is* Δ*-saturated, then* $g \leftrightarrows_{r\tau} E(g)$.
(ii) *For* $g \in \mathbb{G}^{\rho}$: $E(\Delta(g)) \leftrightarrows_{r\tau} g$.

PROOF. (ii) follows from (i) and Proposition 2.1.2. As to (i): Because g is Δ-saturated, applying E does not disconnect nodes of g. That is, g and E(g) have the same nodes. Now let R_m be the *maximal* $r\tau$-bisimulation from g to itself. (By Proposition 1.3.6 there is a maximal one.) We will show that R_m is also an $r\tau$-bisimulation between g and E(g).

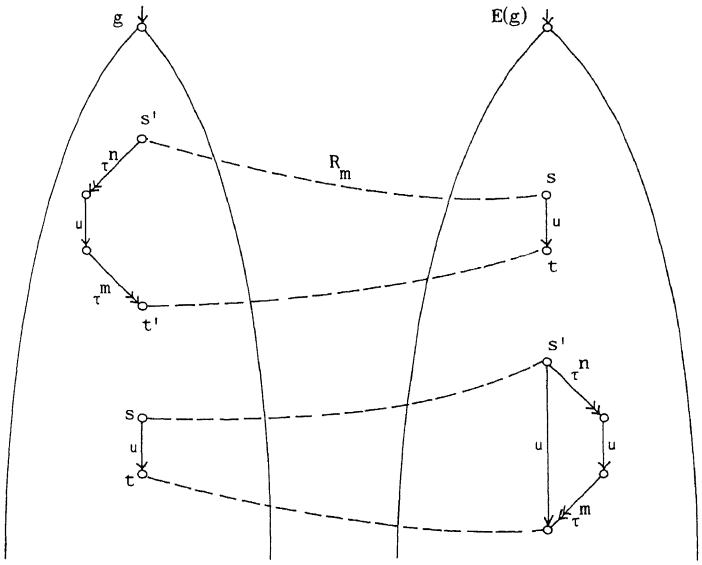


Figure 20

Now the easy half of checking that R_m is the required $r\tau$-bisimulation, is as follows. Let $s \to^{u} t$ be in E(g) such that s',s are related. Then there is a path $s' \twoheadrightarrow t'$ in g, with label $\tau^n u \tau^m$, with t',t related by virtue of the fact that R_m is an $r\tau$-bisimulation.

In the other direction, let $s \to^{u} t$ be in g. Again there is a path $s' \twoheadrightarrow t'$ in g with label $\tau^n u \tau^m$ such that t',t are related. If this is also a path in E(g) we are done. If not: since g is Δ-saturated, and hence (Proposition 2.2.3) E(g) too, we have a direct $s' \to^{u} t'$ step in g (see Figure 20). If this u-step is in fact an ε-step, it is omitted in E(g). But then t and s' are related (since R_m is maximal) and we are done. □

Now we arrive at a key lemma:

LEMMA 2.2.5. *Let* $g,h \in \mathbb{G}^{\rho}$ *be* Δ*-saturated and prenormal. Then:*

$$g \leftrightarrows_{r\tau} h \;\Rightarrow\; g \leftrightarrows h.$$

PROOF. (1) Let R be an $r\tau$-bisimulation between g,h. Then there is no τ-step in g which is "contracted" by R in h, as in Figure 21 (and likewise with g,h interchanged):

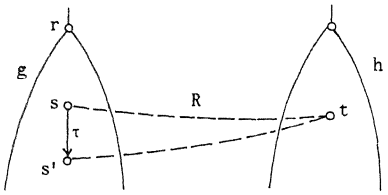


Figure 21

Namely, if s = r, the root of g, then this claim follows by definition of $\leftrightarroweq_{r\tau}$. Otherwise, $s \to^{\tau} s'$ is an internal step ($s' \neq r$ since $g \in \mathbb{G}^{\rho}$) and now by Proposition 1.3.5 (ii):

$$(g)_s \leftrightarroweq_{\tau} (h)_t \leftrightarroweq_{\tau} (g)_{s'}.$$

That is: $s \to^{\tau} s'$ is an ε-step. But then g is not prenormal.

(2) Let $s \to^{u} s'$ ($u \in A_{\tau}$) be a step in g (see Figure 22):

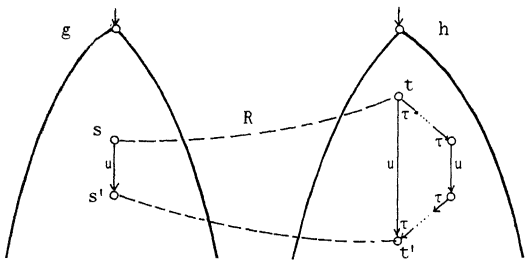


Figure 22

By definition of the $r\tau$-bisimulation R, there is given a t such that $(s,t) \in R$, a path $t \twoheadrightarrow t'$ with label $\tau^n u \tau^m$, for some t' such that $(s',t') \in R$. By Δ-saturation of h, there is now a step $t \to^{u} t'$.

(1) and (2) together imply that the $r\tau$-bisimulation R is in fact an ordinary bisimulation. □

REMARK 2.3. Note that the condition of Δ-saturation in the preceding lemma is necessary: for consider g,h as in Figure 23:

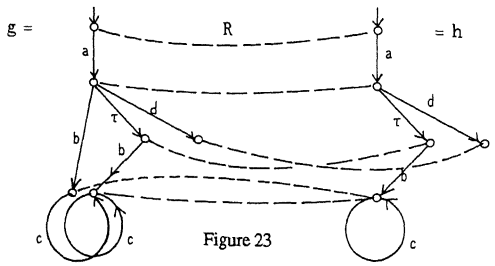


Figure 23

Here g,h are prenormal and rτ-bisimilar (by R as in the figure). However they are not bisimilar. After Δ-saturation they are bisimilar:

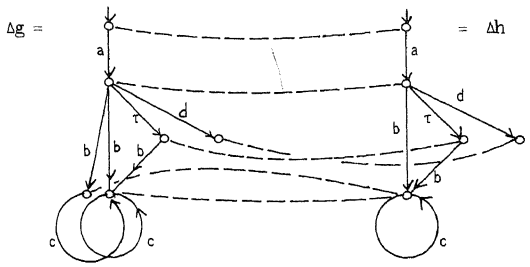


Figure 24

COROLLARY 2.4. *Let* g,h ∈ $\mathbb{G}^{\rho}$ *and let* g $\triangleq_{r\tau}$ h. *Then* E(Δ(g)) $\triangleq$ E(Δ(h)).

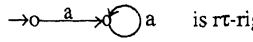
PROOF. By Proposition 2.1.2, Δg $\triangleq_{r\tau}$ Δh. By Proposition 2.2.4, E(Δg) $\triangleq_{r\tau}$ E(Δh). By Proposition 2.2.3, E(Δg) and E(Δh) are Δ-saturated. Hence by Lemma 2.2.5 these two graphs are bisimilar in the ordinary sense. □

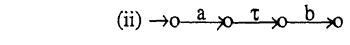
3. Normal and rigid process graphs

In this Section we prove (Corollary 3.3.3) a minimalisation result for finite process graphs. From the previous Section 2 we need only the concept of 'arc' (in 2.1) and 'cluster' (in 2.2). The results in this section are not used in the sequel except for Corollary 3.3.4 which describes a minimalisation procedure for process graphs without τ-steps.

3.1. Rigid process graphs

DEFINITION 3.1.1. (i) Let g ∈ $\mathbb{G}$, and let R be a bisimulation (τ-bisimulation, respectively) from g to itself. Then R is called an *autobisimulation* (τ-*autobisimulation*, respectively) of g.
(ii) Likewise we have for g ∈ $\mathbb{G}^{\rho}$ an rτ-*autobisimulation*.
(iii) If g ∈ $\mathbb{G}$ and the identity relation is the only autobisimulation of g, then g is called *rigid*. Likewise g is τ-*rigid* if it has only the trivial τ-bisimulation. If g ∈ $\mathbb{G}^{\rho}$, g is rτ-*rigid* if it has only the trivial rτ-autobisimulation.

EXAMPLE 3.1.2. (i)  is rτ-rigid but not rigid.

(ii)  is not rτ-rigid, not τ-rigid, but is rigid.

Note that for g ∈ $\mathbb{G}$, τ-rigid implies rigid; and for g ∈ $\mathbb{G}^{\rho}$, τ-rigid implies rτ-rigid. Furthermore, if g ∈ $\mathbb{G}^{\rho}$ and g is rτ-rigid then the subgraphs of g corresponding to internal nodes

are τ-rigid. Also (cf. Proposition 1.3.14) for $g \in \mathbb{G}$:

g *is* τ*-rigid* $\Leftrightarrow$ τg *is* $r\tau$*-rigid.*

PROPOSITION 3.1.3. *Let* $g \in \mathbb{G}^\rho$. *Then:* g *is* $r\tau$*-rigid* $\Leftrightarrow$ g *has only singleton clusters.*

PROOF. ($\Rightarrow$) Suppose g has a cluster containing different nodes s,t. So by definition of cluster, s,t are internal nodes which are τ-bisimilar (i.e. $(g)_s \leftrightarrows_\tau (g)_t$). Let R be a τ-bisimulation between $(g)_s$ and $(g)_t$. Then $Id_g \cup R$ is a non-trivial $r\tau$-autobisimulation of g, contradiction.
($\Leftarrow$) Suppose g has a non-trivial $r\tau$-autobisimulation R. Then R relates different internal nodes s,t to one another. But then s,t are in the same cluster, contradiction. □

3.2. Normal process graphs

DEFINITION 3.2.1. (i) An $r\tau$-rigid process graph $g \in \mathbb{G}^\rho$ is *minimal* if g contains no double edges, no τ-loops and no arcs (see the definition in 2.1).
(ii) If g is $r\tau$-rigid and minimal, we will call g $r\tau$*-normal.*

THEOREM 3.2.2. *Let* $g,h \in \mathbb{G}^\rho$ *be* $r\tau$*-normal and suppose* $g \leftrightarrows_{r\tau} h$. *Then* g,h *are identical.*

PROOF. Let $g \leftrightarrows_{r\tau} h$ via R. Then R is a bijection from NODES(g) to NODES(h), because g,h are $r\tau$-rigid (so they have only singleton clusters; now apply Proposition 1.3.5). Furthermore R maps the edges of g bijectively to those of h; more precisely:
Claim.

(i) If $s \to^u t$ with $s \neq t$ is an edge of g and $(s,s') \in R$, then there is an edge $s' \to^u t'$ in h for some t' with $s' \neq t'$ and $(t,t') \in R$.
(ii) If $s \to^a s$ ($a \in A$) is a loop in g and $(s,s') \in R$ then there is a loop $s' \to^a s'$ in h.
(iii) Likewise vice versa.

With the claim we are through, since R is then an *isomorphism* between labeled graphs. (Intuitively, this can easily be seen by noting that a process graph g without double edges can be considered as an algebraic structure, in the sense of model theory, with universe NODES(g), a constant ROOT(g) and binary relations a,b,c,... (the labels of EDGES(g)).)

Proof of the claim: Let $s \to^u t$ be an edge as in (i) or (ii) of the claim. Let $(s,s') \in R$. (See Figure 25.)

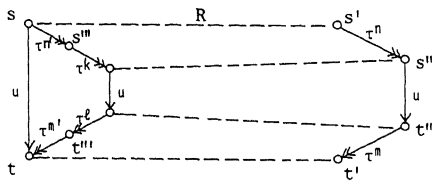


Figure 25

Suppose there is not a 'direct' step $s' \to^u t'$, $(t,t') \in R$. Then since R is an $r\tau$-bisimulation there is

a path s' $\to$ s" (with label τ^n), s" $\to^u$ t", t" $\to$ t' (with label τ^m) from s' to t', (t,t') $\in$ R as in the figure, such that n + m $\neq$ 0. Say n $\neq$ 0. Now s' $\to^{\tau\tau..\tau}$ s" cannot be a τ-cycle (i.e. s $\neq$ s") since by Proposition 1.3.8 the points on a τ-cycle are τ-bisimilar and here we have singleton clusters. So, going backwards from s" we find s"' with s$\neq$s"' as in the figure. Likewise the step s" $\to^u$ t" can be carried backwards to a path s"' $\to$ t"' with label $\tau^k u \tau^l$ in g as in the figure. Finally, carrying t" $\to^{\tau\tau..\tau}$ t' backwards to g we must end in t since R is a bijection between the node sets. However, the result is an arc in g, in contradiction with the normality of g.

Hence there is a direct step s' $\to^u$ t', (t,t') $\in$ R. By the bijectivity of R it follows from s $\neq$ t that s' $\neq$ t', which proves (i); and it follows from s =t that s' =t ', which proves (ii). Claim (iii) is like (ii) with g,h interchanged. □

COROLLARY 3.2.3. *Let* g $\in \mathbb{G}^\rho$.*Then there is a unique* $N_{r\tau}(g) \in \mathbb{G}^\rho$, *the* r$\tau$*-normalisation of* g, *such that* $N_{r\tau}(g)$ *is* rτ*-normal and* g $\leftrightarrows_{r\tau}$ $N_{r\tau}(g)$.

PROOF. Given g, one collapses the clusters (in the sense of identifying nodes as in Definition 1.3.11) to singletons. The resulting g' is rτ-rigid and g $\leftrightarrows_{r\tau}$ g' follows from Proposition 1.3.12. Then superfluous edges (double edges, τ-loops and the primary edges of arcs) are removed. These removals preserve rτ-bisimilarity. The uniqueness follows from Theorem 3.2.2. □

Specialising to the τ-less case, we obtain the following result (used in Section 5 for the completeness proof of BPA_{LR}).

COROLLARY 3.2.4. (i) *Let* g $\in \mathbb{G}^\rho$. *Then there is a unique* N(g), *the normalisation of* g, *such that* N(g) *is normal (i.e. rigid and minimal) and* g $\leftrightarrows$ N(g), *obtained by repeatedly identifying bisimilar nodes and removing double edges.*
(ii) *Let* g,h $\in \mathbb{G}^\rho$ *and* g $\leftrightarrows$ h. *Then* N(g) = N(h). □

4. Operations on process graphs

We will now define some operations on process graphs, namely:

+	(alternative composition or sum)
·	(sequential composition or product)
$\|$	(parallel composition or merge)
$\lfloor\!\lfloor$	(left merge).

4.1. Alternative composition

Let g,h $\in \mathbb{G}$. Then g + h is the result of glueing ρ(g) and ρ(h) together by identifying their roots.
EXAMPLE:

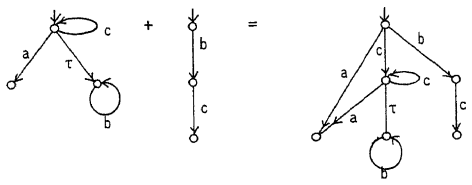


Figure 26

4.2. Sequential composition

Let g,h ∈ $\mathbb{G}$. Then g·h is the result of appending h at all endpoints and all τ–endpoints of g. Here a τ-endpoint is a node s from which only τ-steps are possible and such that $(g)_s$ has no endpoints.

EXAMPLE:

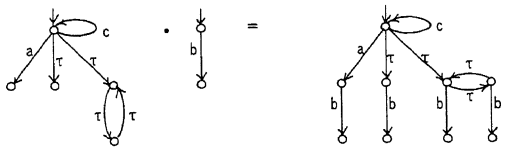


Figure 27

4.3. Parallel composition

Let g,h ∈ $\mathbb{G}$. Then g||h is the result of taking the cartesian product of g,h.

EXAMPLE:

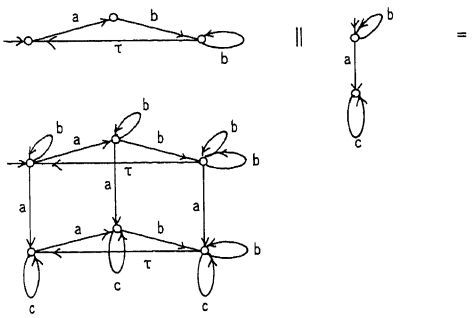


Figure 28

4.4. Left-merge

For g,h ∈ $\mathbb{G}$, the left-merge g$\lfloor\!\lfloor$h is defined as the subgraph of ρ(g)||h obtained by stipulating that an initial step must be one from ρ(g).

EXAMPLE:

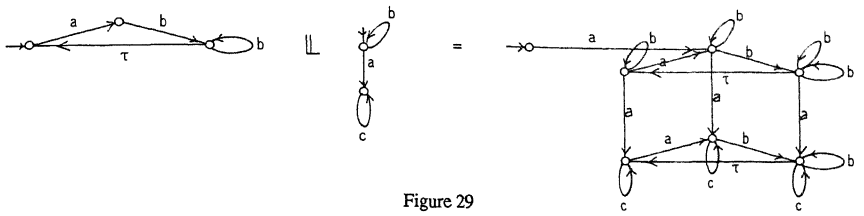


Figure 29

THEOREM 4.5. *rτ-bisimilarity is a congruence w.r.t. the operations* $+,\cdot,\|,\mathbin{\lfloor\!\!\lfloor}$ *on* $\mathbb{G}^{\rho}$. *That is, for* g,g',h,h' *with* $g \triangleq_{r\tau} g'$, $h \triangleq_{r\tau} h'$ *we have* $g \,\square\, g' \triangleq_{r\tau} h \,\square\, h'$ *for* $\square \in \{+,\cdot,\|,\mathbin{\lfloor\!\!\lfloor}\}$.

PROOF. Routine. In all four cases the required rτ-bisimulation (between $g \,\square\, g'$ and $h \,\square\, h'$) is constructed in a straightforward way from given rτ-bisimulations between g,g' and h,h'. □

REMARK 4.5.1. Note that τ-bisimilarity is not a congruence: $a \triangleq_{\tau} \tau a$, $b \triangleq_{\tau} b$ but $a+b \triangleq_{\tau} \tau a+b$ does not hold. Nor is it a congruence w.r.t. $\mathbin{\lfloor\!\!\lfloor}$: $a \mathbin{\lfloor\!\!\lfloor} b \triangleq_{\tau} \tau a \mathbin{\lfloor\!\!\lfloor} b$ does not hold.

4.6. Collapsing process graphs

In this subsection, in which process graphs with +,· and ≙ only will be considered, we define a useful operation on process graphs, namely *collapsing isomorphic subgraphs:*

$$collaps : \mathbb{G} \to \mathbb{G}.$$

If $g \in \mathbb{G}$, let Sub(g) be the set of sub-process graphs of g modulo process graph isomorphism (≃): we abstract from the identity of the nodes. The isomorphism class of a subgraph h of g is h~.

EXAMPLE. (i) If g = →o —a→ o —b→ o —a→ o —b→ o— ... (infinitely many consecutive steps alternatingly labeled with a,b), Sub(g) consists of g~ and g'~, where g' is g with a,b interchanged.

(ii) If h = →o —a→ o —b→ o —a→ ⟲ a, b , then Sub(h) contains five elements.

Now *collaps(g)* is the process graph with node set: Sub(g) ∪ {O}, if g has a terminating path, and Sub(g) else. The *root* of *collaps*(g) is g~, and for $g_1\sim, g_2\sim \in$ Sub(g) there are edges

$g_1\sim \to^a g_2\sim$ whenever $g_1 = a\cdot g_2$ or $g_1 = a\cdot g_2 + h$ for some h,

and $g_1\sim \to^a O$ whenever $g_1 = a$ or $g_1 = a + h$ for some h.

In the example above:

collaps(g) = →(g~ ⟲ b, a → g'~)

and *collaps*(h) ≃ h.

Note that *collaps* does not identify subgraphs which are merely bisimilar. Now there is the following theorem:

THEOREM 4.6.1. *Let* $g,h \in \mathbb{G}$. *Then:*

(i) *collaps*(g) ≙ g

(ii) *collaps*(g □ h) ≙ *collaps*(g) □ *collaps*(h), *for* □ = +,·.

PROOF. (i) The *collaps* operation gives the bisimulation in a direct way: $s \in$ NODES(g) is related to $(g)_S \sim \in$ NODES(*collaps*(g)). It is easy to check that this relation is a bisimulation.
(ii) By Theorem 4.5 restricted to $\leftrightarrows$, $\leftrightarrows$ is a congruence w.r.t. $+,\cdot$ on $\mathbb{G}$. Hence by (i):

$$g \,\square\, h \leftrightarrows \mathit{collaps}(g) \,\square\, \mathit{collaps}(h).$$

Again by (i): $\mathit{collaps}(g \,\square\, h) \leftrightarrows g \,\square\, h$. Therefore (ii) follows. □

5. Proof systems for regular processes without silent moves

As a preparation for the completeness result in Section 6 for "recursion plus τ-steps", we first treat the case without τ-steps. In particular, Milner's complete proof system for this case is presented and compared with a variant (BPA_{LR}) in which the μ-formalism is replaced by systems of recursion equations.

5.1. Preliminary syntax definitions

We will now specify the syntax necessary to deal with the semantic domain of regular processes $\mathbb{R}^{\rho}(+,\cdot)/\leftrightarrows_{RT}$, and at the end of this paper, $\mathbb{R}^{\rho}(+,\cdot,\|,\lfloor\!\lfloor)/\leftrightarrows_{RT}$ (Section 8). At the basis of this syntax is the finite alphabet $A_\tau = A \cup \{\tau\}$, where $A = \{a,b,c,...\}$, of atomic actions, used above as labels of the edges of the process graphs. (We will not notationally distinguish between the formal alphabet symbols and the edge labels; nor will we distinguish the syntactic function symbols $+,\cdot,\|,\lfloor\!\lfloor$ from their semantical counterparts in $\mathbb{R}^{\rho}(+,\cdot,\|,\lfloor\!\lfloor)$. In turn, the latter will not be distinguished from the corresponding operations induced in the quotient structures modulo $\leftrightarrows_{RT}$.)

5.1.1. Linear and guarded terms over A_τ, VAR, $+$, $\cdot$

Let VAR be a denumerably infinite set of variables $\{X,Y,Z,...\}$. Terms (or expressions) T over A_τ, VAR, $+$, $\cdot$ are defined as usual. We will assume commutativity and associativity of $+$, and associativity of $\cdot$. A term T is A_τ-*guarded* if every occurrence of a variable in T is preceded by some $u \in A_\tau$. More formally:
(i) $\tau,a,b,c,...$ are A_τ-guarded,
(ii) if T is A_τ-guarded and T' is an arbitrary term, then $T \cdot T'$ is A_τ-guarded,
(iii) if T,T' are both A_τ-guarded then so is $T + T'$.
Likewise we define: T is A-*guarded*, by omitting τ from the previous definition; $A = \{a,b,c,...\}$ stands for $A_\tau - \{\tau\}$. Instead of A_τ-guarded and A-guarded, we will also say for short: τ-*guarded*, *guarded* respectively.

EXAMPLE. $abX + \tau XX$ is τ-guarded but not guarded, $(aX + b(Y + a))(X + a)$ is guarded hence τ-guarded, $(aX + Y + a)(X + a)$ is not (τ-)guarded.

Further, we define a term T to be *linear* if all occurrences of variables are 'at the end'. More precisely:
(i) variables are linear,

(ii) closed terms (i.e. terms not containing variables) are linear,

(iii) if T,T' are linear, then T + T' is linear,

(iv) if T is closed and T' is linear, then T·T' is linear.

EXAMPLE. The three terms in the previous example are not linear; aX + bY + c is linear; (a + τ)(Y + b) is linear. A term T is *strictly linear* if it is of the form

$$\Sigma_{1\leq i\leq n}\, u_i + \Sigma_{1\leq j\leq m}\, v_j X_j$$

for some $n,m \geq 0$, $u_i, v_j \in A_\tau$ and $X_j \in VAR$.

5.1.2. Canonical LR-expressions

R-expressions are syntactical constructs of the form

$$\langle X_1 \mid E\rangle$$

where $X_1 \in VAR$ and $E = \{X_i = T_i(\mathbf{X}) \mid i = 1,...,n\}$ is a set of "recursion" equations such that the $T_i(\mathbf{X})$ (the *bodies* of E) are A_τ-guarded. The $T_i(\mathbf{X})$ may contain variables from $\mathbf{X} = X_1,...,X_n$, a list of *pairwise different* variables. *Superfluous* equations in E may be omitted; an equation $X_i = T_i(\mathbf{X})$ is superfluous if X_i is not 'accessible' from X_1, in the obvious sense.

An example of an R-expression:

$$\langle X \mid X = a(X\|Y) + bXX,\ Y = (a + b)XY\rangle,$$

also written as

$$X = a(X\|Y) + bXX$$
$$Y = (a + b)XY.$$

Another notation occurring in the literature for $\langle X_1 \mid E\rangle$ is:

X_1 *where* $X_1 = T_1(\mathbf{X}), ..., X_n = T_n(\mathbf{X})$.

LR-expressions are R-expressions where the bodies $T_i(\mathbf{X})$ are linear; this entails that only +,· are admitted as operators in such expressions. In this paper we will only consider LR-expressions.

An LR-expression $\langle X_1 \mid E\rangle$ as above is *canonical* if E does not contain superfluous equations and the bodies $T_i(\mathbf{X})$ are strictly linear.

It is understood that LR-expressions that differ only by a renaming of variables, are identical.

DEFINITION 5.1.2.1.

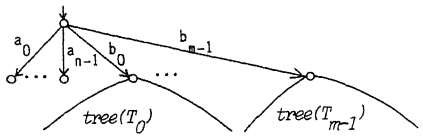
(i) A linear term is in *prefix normal form* if it has no subterm (x + y)z. (Note that a strictly linear $\langle X_1 \mid E\rangle$ is in prefix normal form, but not conversely (cf. $\langle X \mid X = a(X + b)\rangle$).

(ii) If $\langle X_1 \mid E\rangle$ is an LR-expression, then prefix($\langle X_1 \mid E\rangle$) is the LR-expression obtained by reducing the $T_i(\mathbf{X})$ in E via the rewrite rule $(x + y)z \rightarrow xz + yz$ until the prefix normal form of $T_i(\mathbf{X})$ is reached.

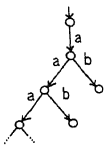
(iii) In the sequel we will need the operation *tree* which unfolds a prefix LR-expression into a possibly infinite tree $\in \mathbb{T}$. It is defined as follows: if

$$\langle X_1 \mid E\rangle = \langle X \mid X = \Sigma_{i<n}\, a_i + \Sigma_{j<m}\, b_j T_j,\ E'\rangle,$$

then *tree*$\langle X \mid E\rangle$ is



EXAMPLE. *tree*<X | X = a(X + b)> =



This definition could be given in a more formal way, but it is standard how to do so. (Just think of $\langle X_1 \mid E\rangle$ where $E = \{X_i = T_i(X_1,...,X_n) \mid i = 1,...,n\}$ as a TRS (Term Rewrite System) with rewrite rules $X_i \rightarrow T_i(X)$. The *tree* of X_1, or of an arbitrary term, as given by this TRS $\langle X_1 \mid E\rangle$ is now defined in the usual way.) The well-definedness of the *tree* operation is a consequence of our requirement that the equations in E are A_τ-guarded.

Before presenting the proof system BPA_{LR} using these LR-expressions, we will first consider Milner's complete inference system M.

5.2. Milner's proof system M

In [10], Milner has given a complete proof system for regular processes with +, prefix multiplication a·, but without τ-steps. Since Milner's completeness theorem will play an important role in the sequel, we will exhibit this proof system, which is called here 'M'; furthermore we formulate an equivalent proof system BPA_{LR} which conforms to the notations of this paper and is the basis of the complete proof systems $BPA_{\tau LR}$ and $PA_{\tau LR}$ in Sections 6 and 8.

The set of terms Ter(M) in Milner's proof system is slightly different from the set of terms as introduced in the previous section.

DEFINITION 5.2.1. Ter(M) is defined as follows. A = {a,b,c,...} is a set of *unary operators* (rather than constants as in 5.1). There is one constant, 0. Further, VAR = {X,Y,Z,...}. Now:

(i) $0 \in$ Ter(M)

(ii) $a \in A, T \in$ Ter(M) $\Rightarrow a(T) \in$ Ter(M).
(Notation: instead of a(T) we write a·T or aT. This construction is called *prefix multiplication.*)

(iii) $T,T' \in$ Ter(M) $\Rightarrow T + T' \in$ Ter(M)

(iv) VAR $\subseteq$ Ter(M)

(v) $X \in$ VAR, $T \in$ Ter(M) $\Rightarrow \mu X(T) \in$ Ter(M). (Alternative notation: μX.T for μX(T).)

5.2.2. Semantics of M

First we need to enlarge the domain $\mathbb{R}$ of process graphs:

DEFINITION 5.2.2.1.

(i) Let $\mathbb{G}' = \mathbb{G} \cup \{\mathbb{O}\}$ where $\mathbb{O}$ is the 'zero' process graph consisting of just one node.

(ii) Let $\mathbb{C} = \{(g,v) \mid g \in \mathbb{G}', v: \text{NODES}(g) \to \mathcal{P}_{\text{fin}}(\text{VAR})\}$.

Here $\mathcal{P}_{\text{fin}}(\text{VAR})$ consists of the finite subsets of VAR, and v is a map assigning to each node of process graph g such a finite set of variables.

A pair (g,v) is called a *chart* in MILNER [10]. Elements from $\mathbb{C}$ will simply be denoted by g,h,....

Note that $\mathbb{G} \subseteq \mathbb{C}$, if (g,Ø) is identified with g, Ø being the map assigning $\emptyset \subseteq$ VAR to each node in g.

EXAMPLE. →o X,Z , 

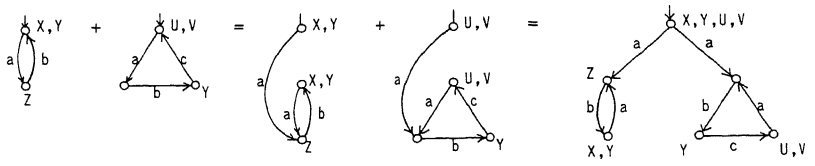
are elements of $\mathbb{C}$.

Process graphs without variables assigned to the nodes (so elements of $\mathbb{G}'$) are called *closed.*

DEFINITION 5.2.2.2. On $\mathbb{C}$ the following operations are defined. Let $g,h \in \mathbb{C}$.

(i) The *sum* g + h is defined as in Section 4: cyclic roots have to be unwound first, and furthermore the variables at the roots which are glued together, are joined. (See MILNER [10] for a more formal definition.)

EXAMPLE:



(ii) a·G (*prefix multiplication*) is defined in the obvious way.

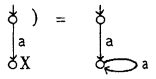
(iii) *Recursion:* μX.g is defined for every X ∈ VAR as follows. Every node s in g with an X assigned to it, gets in the chart μX.g the 'facilities' that the root r of g has:

– whenever $r \to^a t$ is an edge in g, $s \to^a t$ is added in μX.g.

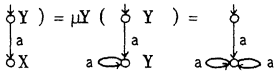
– Moreover, s gets the variables of the root r.

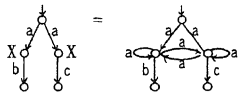
– Finally, all X's are erased.

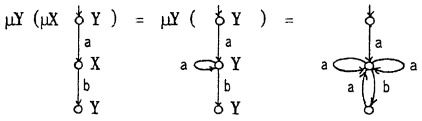
EXAMPLE:

(1) μX (

(2) $\mu X.X = \mu X(X) = \emptyset = \emptyset$

(3) μY (μX 

(4) 

(5) 

On $\mathbb{C}$ the notions of bisimulation and bisimilarity ($\leftrightarrows$) are defined as above (in 1.3.1) with the requirement that related nodes have the same variables assigned to them. So in particular, the restriction of the notion of bisimilarity to $\mathbb{G} \subseteq \mathbb{C}$ coincides with the one defined above. As before, $\leftrightarrows$ is an equivalence relation, and moreover a congruence w.r.t. the operations +, a·, μX on $\mathbb{C}$. We denote by $g/\leftrightarrows$ the congruence class of $g \in \mathbb{C}$ modulo $\leftrightarrows$, and $\mathbb{C}/\leftrightarrows = \{g/\leftrightarrows \mid g \in \mathbb{C}\}$.

To define the semantics $[T]_M \in \mathbb{C}/\leftrightarrows$ of a term $T \in \mathrm{Ter}(M)$ we first define

$$[\,]_M : \mathrm{Ter}(M) \to \mathbb{C}$$

by:

$$[0]_M = \mathbb{O}\ (= \to o),\ [X]_M = \to o\, X$$
$$[T + T']_M = [T]_M + [T']_M$$
$$[a \cdot T]_M = a \cdot [T]_M$$
$$[\mu X.T]_M = \mu X.[T]_M.$$

Finally, $[\![\,]\!]_M : \mathrm{Ter}(M) \to \mathbb{C}$ is defined by

$$[\![T]\!]_M = [T]_M/\leftrightarrows.$$

5.2.3. The proof system M

This system consists of the axioms and rule in Table 1:

M

$x + 0 = x$	A0
$x + y = y + x$	A1
$(x + y) + z = x + (y + z)$	A2
$x + x = x$	A3
$\mu X.T(X) = \mu Y.T(Y)$	$\mu 0$
$\mu X.T(X) = T(\mu X.T(X))$	$\mu 1$
$\dfrac{x = T(x)}{x = \mu X.T(X)}$ $T(X)$ guarded	$\mu 2$
$\mu X(X + T) = \mu X(T)$	$\mu 3$

Table 1

REMARK 5.2.3.1. (i) It is implicitly assumed that '=' is a congruence – this saves us some axioms as compared with the presentation of M in MILNER [10] (p. 454).

(ii) The axiom (scheme) $\mu 0$ allows renaming of variables. Our notation T(X) is slightly informal and intends to avoid the use of explicit substitution operators. In itself, writing T(X) does not say anything about T: it may contain X but also other variables. Only when T(X) and T(S) occur in the same "textual" context, they denote T and T[X:=S] respectively where [X:=S] denotes the appropriate substitution of S for the free occurrences of X.

(iii) One can show that $\mu 0$ is superfluous as every instance of it can be derived from the other axioms and rules (from $\mu 1$-3). E.g.

$$\mu X(X + a\cdot X) = \qquad (\mu 3)$$
$$\mu X(a\cdot X) = \qquad (\mu 1)$$
$$a\cdot\mu X(a\cdot X) = \qquad (\mu 3)$$
$$a\cdot\mu X(X + a\cdot X)$$

and likewise

$$\mu Y(Y + a\cdot Y) = a\cdot\mu Y(Y + a\cdot Y);$$

hence by $\mu 2$:

$$\mu X(X + a\cdot X)\ (= \mu Z(aZ)) = \mu Y(Y + a\cdot Y).$$

(iv) As to $\mu 2$, the definition of 'guarded' as in Section 5.1.1 has to be extended by:

$$T \text{ guarded } \Rightarrow \mu X.T \text{ guarded.}$$

THEOREM 5.2.3.2. (Milner). *For all* $T,S \in Ter(M)$:

$$M \vdash T = S \iff [T]_M = [S]_M. \qquad \square$$

EXAMPLE 5.2.3.3.

(i) $\mu X(aX) = \mu Y(aY + \mu X(aX))$.
PROOF: Let L be $\mu X(aX)$ and $R = \mu Y(aY + \mu X(aX))$. Then $L = aL$, hence $L = aL + aL$; and $R = aR + L = aR + aL$. So both L,R satisfy the guarded recursion equation $x = ax + aL$. Therefore by $\mu 2$, $L = R$.

(ii) $\mu X\mu Y(Y + aX) = \mu X(aX)$

(iii) $\mu X\mu Y(Y + aX + bY) = \mu X(aY + bY)$

(iv) Suppose $M,N,M',N' \in Ter(M)$ satisfy

$$M = aM + bN \qquad M' = aM' + bN'$$
$$N = cM + dN \qquad N' = cM' + dN'$$

Then $\vdash M = M'$, since both solve the guarded equation

$$x = ax + b\cdot\mu Y(cx + dY).$$

(v) Every closed $T \in Ter(M)$ is provably equal to a term T' where all subterms are guarded.

5.3. The proof system BPA_{LR}

We will now give an 'equivalent' proof system BPA_{LR}. The terms of BPA_{LR} as in Section 5.1, that is: 0 is absent, multiplication is general, $a,b,c,... \in A$ are constants and R-expressions take the place of μ-expressions in M. Moreover, the 'bodies' of the LR-expressions must be guarded.

BPA_{LR}

$x + y = y + x$	A1
$(x + y) + z = x + (y + z)$	A2
$x + x = x$	A3
$(x + y)z = xz + yz$	A4
$(xy)z = x(yz)$	A5
$\dfrac{x_i = \langle X_i \mid E\rangle,\ i=1,\ldots,n}{x_1 = T_1(\mathbf{x})}$	R1
$\dfrac{x_i = T_i(\mathbf{x}),\ i=1,\ldots,n}{x_1 = \langle X_1 \mid E\rangle}\ T_i(\mathbf{X})$ is A-guarded	R2

Table 2

Here $E = \{X_i = T_i(X_1,\ldots,X_n) \mid i = 1,\ldots,n\}$. The rules R1,2 correspond to μ1,2 in Table 1. In particular, R1 implies the following axiom (which is equivalent to R1):

$$\langle X_1 \mid E\rangle = T_1(\langle X_1 \mid E\rangle, \ldots, \langle X_n \mid E\rangle)$$

and this axiom corresponds exactly to μ1.

The axiom μ3 in M has no counterpart in BPA_{LR}, by the restriction on the T_i in an LR-expression. The axioms A4, A5 come in here since multiplication is general.

5.3.1. Semantics of BPA_{LR}

As for M, we define the semantics [M] of $M \in Ter(BPA_{LR})$ via the intermediate semantics [M] in $\mathbb{R}$. The main difference is that while μ-terms obtained their intermediate semantics in an 'inside-out' way, via charts $\in \mathbb{C}$, LR-expressions (which are always closed) obtain their semantics 'at once'. More precisely:

DEFINITION 5.3.1.1. [ ]: $Ter(BPA_{LR}) \to \mathbb{R}$ is defined by the following inductive clauses:

(i) $[a] = \to o \xrightarrow{a} o$

(ii) $[S + T] = [S] + [T]$

(iii) $[S \cdot T] = [S] \cdot [T]$

(iv) $[\langle X \mid E\rangle] = \rho$ *collaps tree prefix* $\langle X \mid E\rangle$.

Further, [ ]: $Ter(BPA_{LR}) \to \mathbb{R}/\leftrightarrows$ is defined by $[T] = [T]/\leftrightarrows$.

It is easy to see that the operation [ ] thus defined, indeed yields process graphs $\in \mathbb{R}$ (i.e. finite ones). This is a consequence of the fact that tree$\langle X \mid E\rangle$ has only finitely many subtrees: namely not more than the number of subterm occurrences in E.

REMARK 5.3.1.2. (i) If $\langle X_1 \mid E\rangle$ is a canonical LR-expression, define the *direct intermediate semantics* $\lfloor\langle X_1 \mid E\rangle\rfloor$ where $E = \{X_i = T_i(\mathbf{X}) \mid i = 1,\ldots,n\}$ as the graph $\in \mathbb{R}$ with nodes $X_1,\ldots,X_n$ (and possibly the termination node O), root X_1, and transitions as given in the obvious way by the equations in E. Now it is not hard to prove that $[\langle X_1 \mid E\rangle] \leftrightarrows \lfloor\langle X_1 \mid E\rangle\rfloor$. In the sequel these two are sometimes confused - but only in contexts where we work modulo $\leftrightarrows$.

(ii) Instead of using the present *collaps* operation in the definition of the intermediate semantics of LR-expressions, we could equally well have used a collapsing operation which collapses all *bisimilar* subgraphs (rather than *isomorphic* ones).

The proof system BPA_{LR} turns out to be "equivalent" to Milner's system M in a sense which will be made more precise in the next section. In itself this is not very surprising – the rationale for our introduction of BPA_{LR} is the wish to extend Milner's completeness theorem (5.2.3.2) to the case where τ-steps are present; and at least in the treatment below, LR-expressions are more suitable for that purpose than μ-expressions. The presence of general multiplication in BPA_{LR} is not essential here (but would be in extensions of BPA_{LR} to include nonlinear recursion equations); nor is the absence of '0' essential.

As for M, there is the following completeness theorem (5.3.2) for BPA_{LR}. Part of the proof is mutatis mutandis the same as for Milner's completeness theorem: there we give a sketch of one crucial argument for the sake of completeness. Another part of the proof employs a new argument based on the normalisation procedure of Section 3.

THEOREM 5.3.2. *Let* T,S *be closed terms* $\in$ $Ter(BPA_{LR})$. *Then*

$$BPA_{LR} \vdash T = S \;\Leftrightarrow\; [T] \leftrightarrows [S] \;\Leftrightarrow\; [\![T]\!] = [\![S]\!].$$

PROOF. **Soundness of BPA_{LR}.** The soundness of axioms A1-5 is easy to verify. As to R1: let ξ_i, $\langle X_i \mid E\rangle$ be as in the formulation of R1 in Table 2. (In R1 in Table 2, ξ_i is x_i.) We have to prove

$$[\xi_1] \leftrightarrows [T_1(\xi_1,\ldots,\xi_n)].$$

We may suppose, by definition of [ ], that $\langle X_1 \mid E\rangle$ is in prefix normal form. For definiteness, let us consider the LR-expression

$$\langle X \mid E\rangle \equiv \langle X \mid X = abX + a(X + Y + cY) + b,\ Y = b(aX + dY) + e\rangle,$$

abbreviated by ξ. Further, $\eta \equiv \langle Y \mid E\rangle$. We have to prove

$$[\xi] \leftrightarrows [ab\xi + a(\xi + \eta + c\eta) + b],$$

or

$$[\xi] \leftrightarrows ab[\xi] + a([\xi] + [\eta] + c[\eta]) + b$$

(with a slight abuse of notation: a,b,c in the last RHS stand for $\rightarrow o \xrightarrow{a} o$ etc., +,· are sum, product in $\mathbb{R}$.) That is (abbreviating *collaps tree* by $\underline{ct}$ and *tree* by $\underline{t}$):

$$\underline{ct}(\xi) \leftrightarrows ab\cdot\underline{ct}(\xi) + a(\underline{ct}(\xi) + \underline{ct}(\eta) + c\cdot\underline{ct}(\eta)) + b.$$

By Theorem 4.6.1 this amounts to proving

$$\underline{t}(\xi) \leftrightarrows ab\cdot\underline{t}(\xi) + a(\underline{t}(\xi) + \underline{t}(\eta) + c\cdot\underline{t}(\eta)) + b.$$

Indeed this holds, even with '=' for '$\leftrightarrows$', by definition of *tree*.

The soundness proof of R2 can be found in a detailed way in MILNER [10]. We give a sketch of the main idea involved (again considering a definite example): suppose, as the premiss of the rule R2 that for some terms M_1,M_2,M_3:

$$\begin{aligned} [M_1] &\leftrightarrows [a + bM_2 + cM_3] && (= a + b[M_2] + c[M_3]) \\ [M_2] &\leftrightarrows [bM_1 + aM_3] && (= b[M_1] + a[M_3]) \\ [M_3] &\leftrightarrows [b + cM_3 + aM_2] && (= b + c[M_3] + a[M_2]). \end{aligned}$$

Then we must prove:

$$[M_1] \leftrightarrows [\ \langle X \mid X = a + bY + cZ,\ Y = bX + aZ,\ Z = b + cZ + aY \rangle\ ]$$

I.e. (by Remark 5.3.1.2(i)) we must prove that $[M_1]$ is bisimilar to the graph in Figure 31.

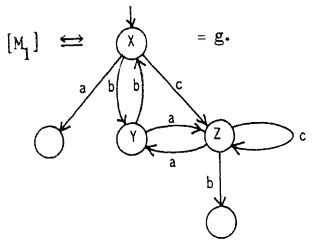


Figure 31

Let g be the displayed process graph. Let $g_i = [M_i]$. Now we want to 'lay out' the graph g_1 on g in such a way that edges are respected. We sketch the procedure:

STEP 1. $g_1 \leftrightarrows a + bg_2 + cg_3$. Therefore, by elementary properties of $\leftrightarrows$, it follows that the direct subgraphs of g_1, call them $g_{11},\ldots,g_{1n}$, can be matched with the direct subgraphs of $a + bg_2 + cg_3$. In a picture, we can lay out the $g_{11},\ldots,g_{1n}$ along the initial part of g: e.g.

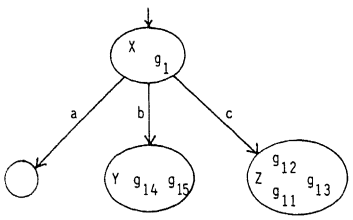


Figure 32

STEP 2. Now in the example, $g_{11} \leftrightarrows g_3 \equiv [M_3]$. Since $g_{11} \leftrightarrows g_3 \leftrightarrows b + cg_3 + ag_2$, the subgraphs of g_{11}, call them $g_{111},\ldots,g_{11k}$, can be matched with the direct subgraphs of $b + cg_3 + ag_2$. So we lay out the graphs $g_{111},\ldots,g_{11k}$ onto g_3,g_2, say as follows:

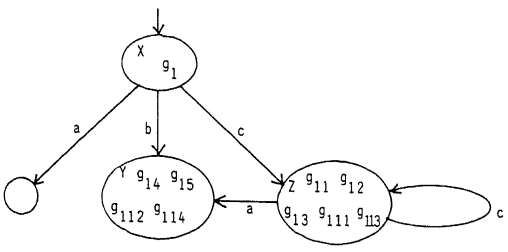


Figure 33

Likewise the subprocesses of g_{12},g_{13},... are laid out. Continuing this way we can lay out all the subprocesses of g_i onto g, in such a way that edges leading from one subprocess to its direct subprocesses, are respected by g. But this lay-out then gives a bisimulation as required in the obvious way.

Completeness. (This argument differs from the completeness proof in MILNER [10].) First we transform an expression to a canonical LR-expression. That products of LR-expressions can be eliminated is demonstrated by Remark 5.3.4 (v). From Section 3 we know that if graphs g,h are bisimilar, repeatedly identifying bisimilar nodes in g and likewise in h, and removal of double edges leads to a common "reduct" of g and h. Now removal of double edges is provable (from A3, R1, R2). Also identification of bisimilar nodes is provable, namely by the following rule whose instances are provable as shown in the example below (5.3.3). In this 'identification rule' the following notation is used: $E = \{X_i = T_i(X_1,...,X_n) \mid i = 1,...,n\}$, and $E_{k=k'}$ results from E by replacing $X_k = T_k(\mathbf{X})$ by $X_k = T_k(\mathbf{X}) + T_{k'}(\mathbf{X})$, removing $X_{k'} = T_{k'}(\mathbf{X})$, and replacing all occurrences of $X_{k'}$ by X_k.

EXAMPLE: If $E = \{X_1 = aX_1 + bX_2 + cX_3, X_2 = aX_1 + cX_2, X_3 = aX_1 + aX_3\}$, then $E_{2=3} = \{X_1 = aX_1 + bX_2 + cX_2, X_2 = aX_1 + cX_2 + aX_1 + aX_2\}$.

Identification rule
$$\frac{\langle X_k \mid E\rangle = \langle X_{k'} \mid E\rangle \text{ for some } k,k'}{\langle X_1 \mid E\rangle = \langle X_1 \mid E_{k=k'}\rangle}$$

The identification rule is easily proved from A3, R1, R2 (see Example 5.3.3). Hence BPA_{LR} is complete. □

EXAMPLE 5.3.3. Let $\xi \equiv \langle X \mid E\rangle \equiv \langle X \mid X = bY + cZ, Y = aY, Z = aZ + aU, U = aU\rangle$ and $\eta \equiv \langle Y \mid E\rangle$, $\zeta \equiv \langle Z \mid E\rangle$, $\upsilon \equiv \langle U \mid E\rangle$.

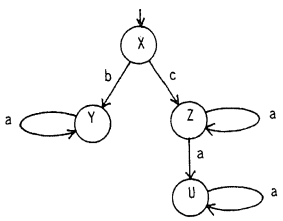


Figure 34

Now clearly $BPA_{LR} \vdash \eta = \zeta$. Hence by the identification rule:

$$\xi = \langle X \mid X = bY + cY, Y = aY + aY + aU, U = aU\rangle \qquad (*).$$

Indeed $BPA_{LR} \vdash (*)$; namely:

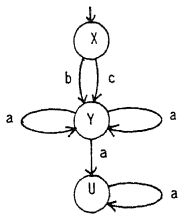


Figure 35

by R1, $\vdash \xi = b\eta + c\zeta, \eta = a\eta, \zeta = a\zeta + a\upsilon, \upsilon = a\upsilon$. Since $\vdash \eta = \zeta$, we have by A3:
$\vdash \xi = b\eta + c\zeta, \eta = \eta + \zeta = a\eta + a\zeta + a\upsilon; \upsilon = a\upsilon$ and by substitution of η for ζ:
$\vdash \xi = b\eta + c\eta, \eta = a\eta + a\eta + a\upsilon, \upsilon = a\upsilon$. Hence (*), by R2.

REMARK 5.3.4. A standard manoeuvre to prove identities between LR-expressions in BPA_{LR} is using R1 to 'externalize' an LR-expression, apply some axioms and 'internalize' the result with R2. In this way one proves:

(i) $BPA_{LR} \vdash T\cdot\langle X \mid E\rangle = \langle Y \mid Y = T\cdot X, E\rangle$ for T closed, Y not in $\langle X \mid E\rangle$.

(ii) Renaming of bound variables.

(iii) Omitting of superfluous equations: $BPA_{LR} \vdash \langle X \mid E\rangle = \langle X \mid E \cup F\rangle$.

(iv) The rule

$$\frac{T_i(\mathbf{X}) = T_i'(\mathbf{X})}{\langle X_1 \mid X_i = T_i(\mathbf{X}), E\rangle = \langle X_1 \mid X_i = T_i'(\mathbf{X}), E\rangle}$$

is provable, i.e. all its instances are provable.

(v) Multiplication of LR-expressions is realised by appending the right factor at all terminal nodes of the left factor. This is also provable: e.g.

$$\vdash \langle X \mid X = aX + b\rangle\cdot\langle Y \mid Y = cY\rangle = \langle X \mid X = aX + bY, Y = cY\rangle$$

is proved as follows. Let $\xi \equiv \langle X \mid X = aX + b\rangle$, $\eta \equiv \langle Y \mid Y = cY\rangle$. Then $\xi = a\xi + b$, $\eta = c\eta$. Hence $\xi\eta = (a\xi + b)\eta = a\xi\eta + b\eta$. Therefore (R2) $\xi\eta = \langle Z \mid X = aZ + bY, Y = cY\rangle$.

REMARK 5.3.5. (The Expansion Rule.) The following 'proof rule' is often convenient to prove in BPA_{LR} the equality of canonical LR-expressions. It is the syntactical counterpart of the normalisation theorem 3.3.4.

(1) Let $\langle X_1 \mid E\rangle \equiv \langle X_1 \mid \{X_i = T_i(\mathbf{X}) \mid i = 1,...,n\}\rangle$ be a canonical LR-expression. Then we may 'expand' the j-th equation $X_j = T_j(\mathbf{X})$ as follows:
let $X_j, X'_j, X''_j, ..., X^{(l_j)}$ be X_j with some variant variables (fresh symbols). Add to E the equations $X_j^{(k)} = T_j(\mathbf{X})$, $k = 1,...,l_j$. Result: E'. Now replace, in an arbitrary way, occurrences of X_j in the RHS's of equations in E', by occurrences of $X_j^{(k)}$, $k = 0,...,l_j$. Result: E*.

It is not hard to prove that

$$BPA_{LR} \vdash \langle X_1 \mid E\rangle = \langle X_1 \mid E^*\rangle.$$

EXAMPLE: $\vdash \langle X \mid X = aX + bX\rangle = \langle X \mid X = aX + bY, Y = aY + bX\rangle$.

(2) In a more refined version of this procedure, we may take copies of the aX_j in the RHS's of E' (i.e. replace aX_j by $aX_j + aX_j + ... + aX_j$) and then substitute the variant variables $X_j^{(k)}$.

Moreover, the procedure may be applied simultaneously to several $X_{j1}, X_{j2}, ...$ in X.

EXAMPLE: $\vdash$ <X | X = aX + bY + cX, Y = aY + bX>

= <X | X = aX' + bY'' + cX'' + cX + cX,
X' = aX + aX' + bY + bY' + cX,
X'' = aX + aX'' + bY'' + cX + cX' + cX',
Y = aY' + aY'' + bX',
Y' = aY' + aY'' + aY + bX'' + bX,
Y'' = aY + bX>.

Now it follows from the normalisation procedure described in Section 3, that this expansion procedure is complete as far as canonical LR-expressions are concerned.

Note that root-unwinding (ρ) is an instance of Expansion. Likewise the procedure in the proof of Thm. 7.3 to remove loops.

5.4. A comparison between Milner's M and BPA_{LR}

Although there is an obvious resemblance between M and BPA_{LR}, the semantic mappings [[ ] for BPA_{LR} are rather differently defined. Comparing the effects of these semantic mappings is the purpose of the present section. The first task is to find syntactic translations between closed M-terms and closed BPA_{LR}-terms. We will define mappings

$\varphi: Ter_c(BPA_{LR}) \to Ter_c(M)$
$\psi: Ter_c(M) \to Ter_c(BPA_{LR})$

(Ter_c denotes 'closed terms') such that $\psi \circ \varphi = id$ and every $T \in Ter_c(M)$ is provably equal to some T' in the range of φ.

DEFINITION of φ. Let $T \in Ter_c(BPA_{LR})$. Then $\varphi(T) = \phi(\textit{prefix}\ T)$, and φ is inductively defined as follows:

$\phi(T_1 + T_2) = \phi(T_1) + \phi(T_2)$
$\phi(aT) = a \cdot \phi(T)$
$\phi(a) = a$
$\phi(<X_1 \mid E>) = \mu X_1.\phi^E_{X_1} T_1(X_1,...,X_n)$

(here $E = \{X_i = T_i(X_1,...,X_n) \mid i = 1,...,n\}$). For $V \subseteq \{X_1,...,X_n\}$ the auxiliary operators ϕ^E_V are defined by:

$\phi^E_V(T_1 + T_2) = \phi^E_V(T_1) + \phi^E_V(T_2)$
$\phi E_V(aT) = a \cdot \phi^E_V(T)$
$\phi^E_V(a) = a$
$\phi^E_V(X_i) = X_i$ if $X_i \in V$
$\phi^E_V(X_i) = \mu X_1.\phi^E_{V \cup \{Xi\}}\ T_i(\mathbf{X})$ if $X_i \notin V$.

EXAMPLE. Let T be ab<X | E> ≡ ab<X | X = aY, Y = bX>. Then

$\phi(T) = ab\phi<X \mid E> = ab\mu X.\phi^E_X(aY) =$
$ab\mu X.a\phi^E_X(Y) = ab\mu X.a\mu Y.\phi^E_{X,Y}(bX) =$

$$ab\mu X.a\mu Y.b\phi^{E}_{X,Y}(X) = ab\mu X.a\mu Y(bX).$$

The operation ψ is in fact not defined on all closed M-terms, but only on those where the bodies of the μ-expressions are of the form $\Sigma a_i T_i$. (Here T_i may be 0.) It is not hard to prove that every $T \in Ter_c(M)$ is provably equal to such a term. Now ψ replaces in a closed M-term as described, every maximal (hence closed) μ-expression by an LR-expression in the obvious way, by assigning to the μ-expression $\mu X.T(X)$ the variable X and using $\mu 1$ to obtain $X = T(X)$. The μ-expressions in T(X) are eliminated likewise.

EXAMPLE. $\psi(ab(\mu X.a(\mu Y.bX))) = ab\psi(\mu X.a(\mu Y.bX)) =$
$ab\langle X \mid X = aY, Y = bX\rangle$.

Now φ and ψ are not exactly each other's inverse; e.g.:

$$\varphi(\langle X \mid X = aY + bY, Y = cX + dY\rangle) =$$
$$\mu X(a\mu Y(cX + dY) + b\mu Y(cX + dY)) =$$
$$\mu X(a\mu Y(cX + dY) + b\mu Z(cX + dZ)) \equiv T,$$

and

$$\psi(T) = \langle X \mid X = aY + bZ, Y = cX + dY, Z = cX + dZ\rangle.$$

But they are inverse "modulo $\leftrightarrows$". In a theorem:

THEOREM 5.4.1. *Let* $T \in Ter_c(BPA_{LR})$ *and* $S \in Ter_c(M)$, $S \in Dom\ \psi$. *Then:*
(i) $[\varphi(T)]_M = [T]$
(ii) $[\psi(S)] = [S]_M$
(iii) $[\psi(\varphi(T))] = [T]$
(iv) $[\varphi(\psi(S))]_M = [S]_M$.

In fact the situation is as in the following diagram (see Figure 36). Here the diagrams formed by the heavy arrows are commuting diagrams.

PROOF. (iii) and (iv) follow at once from (i) and (ii).
PROOF of (i). The proof consists of seven parts, some of which only will be sketched.
(1) First we define a *derivation* to be a triple $T \rightarrow^a S$ (S is called *derived*) where $T,S \in Ter(BPA_{LR})$, $a \in A$, and such that $T = aS + R$ or $T = aS$ for some R.

EXAMPLE: $a(X + bY) \rightarrow^a X + bY$; $X + bY \rightarrow^b Y$.

Next, given an LR-expression $\langle X_1 \mid E\rangle$ where $E = \{X_i = T_i(X) \mid i = 1,...,n\}$, we define the *derived subterm occurrences* (dso's) of $\langle X_1 \mid E\rangle$ as follows:
- the occurrence of X_1 in the LHS of $X_1 = T_1(X)$ is a dso;
- dso's are closed under derivation;
- if $X_i + T$ is a dso, then the derived subterms of $T_i(X)$ are dso's.

Dso's will be denoted by *underlining* these subterm occurrences. Since we want to distinguish all

occurrences, we imagine these underlinings to have different *colours* (e.g. a natural number).

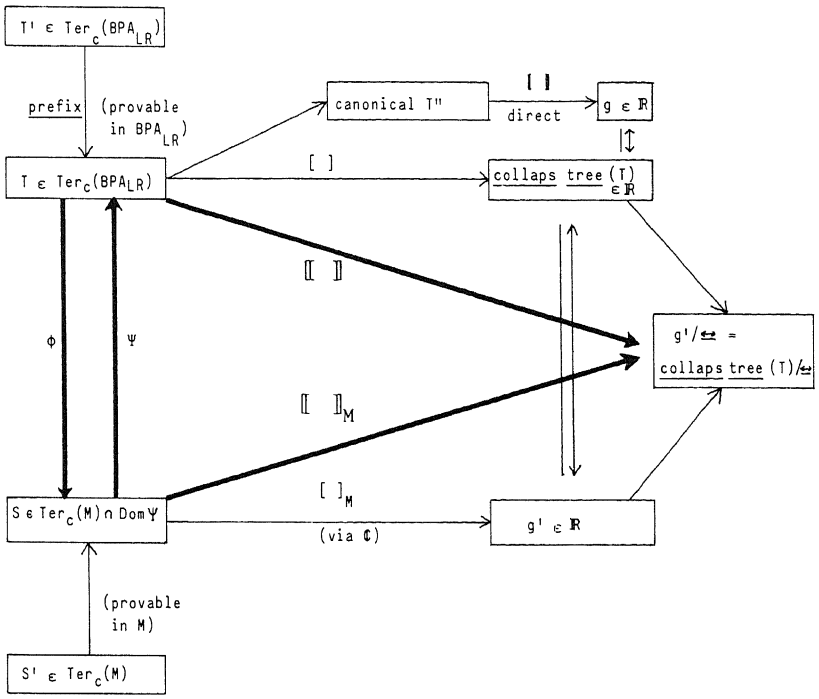


Figure 36

EXAMPLE. Let $\langle X \mid E\rangle \equiv \langle X \mid X = aX + b(Y + c) + d,\ Y = a(X + bY)\rangle$. Then the dso's are given by the underlining:

$$\langle X \mid \underline{X} = a\underline{X} + b(\underline{Y + c}) + d,\ Y = a(\underline{X + b\underline{Y}})\rangle.$$

Now the *derived subterm graph* of $\langle X_1 \mid E\rangle$, $\mathrm{dsg}\langle X_1 \mid E\rangle$, has root $\underline{X}$, nodes: the dso's with an identification of all occurrences of $\underline{X_i}$, and edges: the derivations.

EXAMPLE:

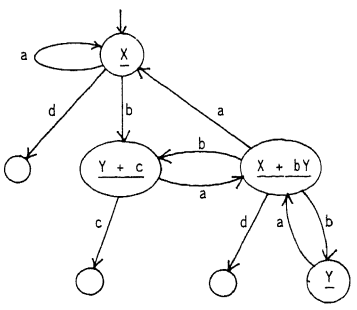


Figure 37

(2) CLAIM: $dsg\langle X_1 \mid E\rangle \triangleq [\langle X_1 \mid E\rangle]$.

PROOF of the claim. $\langle X_1 \mid E\rangle$ can be converted (in BPA_{LR}) to a canonical $\langle X_1 \mid F\rangle$ having the same dsg, by replacing the non-variable dso's by fresh variables.

EXAMPLE: for $\langle X \mid E\rangle$ as above:

$$\langle X \mid F\rangle \equiv \langle X \mid X = aX + bY' + d,\ Y' = aZ + c,\ Z = aX + bY' + d + bY,\ Y = aZ\rangle.$$

For a canonical $\langle X \mid F\rangle$, the dsg coincides with $[\langle X \mid F\rangle]$. Hence

$$dsg\langle X \mid E\rangle = dsg\langle X \mid F\rangle = [\langle X \mid F\rangle] \triangleq [\langle X \mid E\rangle],$$

where the last step is by soundness of BPA_{LR}.

(3) We repeat parts (1), (2) now for μ-expressions $\in Ran(\varphi)$.

Derivations $T \to^a S$ are defined by:

- $aS \to^a S$
- if $T \to^a S$ then $T + T' \to^a S$
- if $T \to^a S$ then $\mu X(T) \to^a S$.

(The first two clauses are as for BPA_{LR}-terms; the third is new.)

Now the *dso's* of $T \in Ter_c(M)$ are:

- T itself
- dso's are closed under derivation.

EXAMPLE: Let T be $\varphi(\langle X \mid E\rangle)$ from above. Then its dso's are:

$$\mu X(a\underline{X} + b(\mu Y(a(\underline{X} + b\underline{Y})) + c) + d).$$

Further, the dsg of T is defined as follows:

every dso is a node (now there is no identification of occurrences of the same variable); T itself is the root. Edges are given corresponding to the definition of dso's, together with the stipulation that if $(\mu X.T) \to^a S$ and $\underline{X}$ is a dso in T, then $\underline{X} \to^a S$.

(4) CLAIM. Let $T \in Ter_c(M) \cap Ran(\varphi)$. Then:

$$dsg(T) = [T]_M.$$

The proof of this claim follows straightforwardly from the definitions of RHS and LHS.

EXAMPLE: For the μ-expression in the example above we have the graph

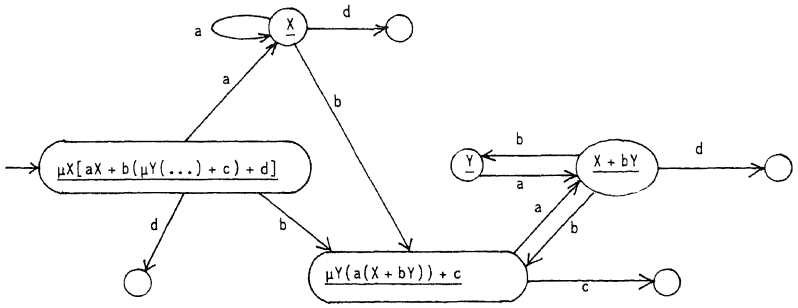


Figure 38

(5) We now extend the translation φ from BPA_{LR}-terms to M-terms to $\underline{\varphi}$, accepting underlined BPA_{LR}-terms and delivering underlined M-terms. Some typical clauses are:

$$\underline{\varphi}(\langle \underline{X} \mid E \rangle) = \underline{\mu X.\phi^E{}_X(E)}$$

$$\underline{\phi^E{}_V(T + S) = \phi^E{}_V(T) + \phi^E{}_V(S)}$$

$$\phi^E{}_V(\underline{X_i}) = \underline{X_i} \qquad \text{if } \underline{X_i} \in V$$

$$= \underline{\mu X_i.\phi_{V \cup \{X_i\}}\, T_i(X)} \quad \text{else.}$$

(6) Next one proves that $\underline{\varphi}$ carries over the dso's of a BPA_{LR}-term T into the dso's of φ(T).

(7) Bearing in mind that underlinings have a colour, we now define a bisimulation R between dsg(T) and dsg(φ(T)): the relation R is simply: having the same colour. It follows easily that R is a bisimulation indeed.

The proof of (ii) is left to the reader. □

6. A proof system for regular processes with silent moves

In this section we will formulate a proof system $BPA_{\tau LR}$ for regular processes with τ-steps, subject to the operations +, ·. The proof of the completeness of this system has as main ingredients: the analysis of rτ-bisimulation described in Theorem 2.4, and the completeness of Milner's M, or, as we will use, its equivalent formulation BPA_{LR}. As an introductory step we consider first Milner's τ-laws for the simple case of finite process trees - i.e. finite processes not involving recursion.

6.1. Milner's τ-laws

To place matters in some perspective, we review some well-known facts about Milner's notions of observational equivalence, observational congruence, Park's notion of bisimulation and Milner's τ-laws. This will aim at an understanding of the difference between τ-bisimulation ($\leftrightarrows_\tau$) and its variant $\leftrightarrows_{r\tau}$, introduced in [4].

(1) Let $\mathbb{T}$ be the domain of finite process trees, and let $\mathbb{T}' = \mathbb{T} \cup \{\mathbb{O}\}$ where $\mathbb{O}$ is the zero graph $\rightarrow$O, consisting of one node only. MILNER [9] defines a decreasing sequence $\sim_0 \supseteq \sim_1 \supseteq \ldots \supseteq \sim_k \supseteq \ldots$ of equivalence relations on $\mathbb{T}'$, and calls $\sim = \cap_{k\geq 0} \sim_k$ *strong equivalence*. This is a congruence w.r.t. + and 'prefix multiplication' $u\cdot$ ($u \in A_\tau$). PARK [14] replaced the construction of $\sim$ by his more directly defined notion of bisimulation $\leftrightarrows$, which is not the same as $\sim$ on $\mathbb{T}'$ but which does coincide with $\sim$ in the restriction to finite trees $\mathbb{F}'$ ($= \mathbb{F} \cup \{\mathbb{O}\}$). It turns out that $\mathbb{F}'(+, u\cdot, \mathbb{O}) / \leftrightarrows$ is isomorphic to the initial algebra corresponding to the present signature and axioms:

$x + 0 = x$	A0
$x+y = y + x$	A1
$(x + y) + z = x + (y + z)$	A2
$x + x = x$	A3

Table 4

(2) For the signature favoured in this paper, +, ·, u ($\in A_\tau$), there is the very similar result that $\mathbb{F}(+, \cdot, u) / \leftrightarrows$ is isomorphic to the initial algebra of the axioms in BPA:

BPA

$x+y = y + x$	A1
$(x + y) + z = x + (y + z)$	A2
$x + x = x$	A3
$(x + y)z = xz + yz$	A4
$(xy)z = x(yz)$	A5

Table 5

(3) Next, τ is introduced, i.e. its special properties are postulated now. This leads Milner to the notion of $\simeq$ (*observational equivalence*), defined as in (1) as the limit of a decreasing sequence $\simeq_i$. Again, the definition is smoother via the corresponding notion of τ-bisimulation $\leftrightarrows_\tau$, which although different on infinite trees, coincides with $\simeq$ on the finite trees in $\mathbb{F}'$. This equivalence, as pointed out by MILNER [9], is *not a congruence*, notably not w.r.t. +. For, $\tau a \leftrightarrows_\tau a$, $b \leftrightarrows_\tau b$ but not $\tau a+b \leftrightarrows_\tau a+b$. Additive contexts are the only ones where $\leftrightarrows_\tau$ 'misbehaves'.

Still, the equivalence relation $\leftrightarrows_\tau$ can be axiomatised as MILNER [9] shows (see also for a clear discussion on these matters: BROOKES [6]), as follows:

$x \simeq \tau x$
$C[x + \tau x] \simeq C[\tau x]$
$C[ux + u(\tau x + y)] \simeq C[u(\tau x + y)]$.

Here C[ ] is an arbitrary context. Note that the first axiom may not be used in a context. MILNER [9] states the completenes of this set of axioms for $\simeq$ (or $\leftrightarrows_\tau$). (The proof is in HENNESSY - MILNER [7] and also in MILNER [11].)

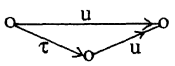
(4) Although $\simeq$ ($\leftrightarrows_\tau$) is not a congruence on all finite trees in $\mathbb{F}'$, it is one after restriction to *stable* trees. A tree is called 'stable' in [9] if it has no initial τ-step. Thus 'a+b' is stable but 'τa + b' is not.

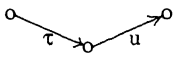
(5) $\simeq$ not being a congruence w.r.t. +, Milner defines $\simeq^C$, *observational congruence*, by 'brute force': $x \simeq^C y \Leftrightarrow \forall C[\ ]\ C[x] \simeq C[y]$. This $\simeq^C$ is by definition a congruence, and it turns out that $\simeq^C$ can be axiomatised too, namely by A0-3 and on top of those the so-called τ-laws of Milner:

$u\tau x = ux$	T1'
$x + \tau x = \tau x$	T2
$u(x + \tau y) = u(x + \tau y) + uy$	T3

That is, compared to the axiom schemes in (3) one has only to prefix both sides of the first axiom by a guard $u \in A_\tau$. However, by the brute force definition of $\simeq^C$, the direct connection with (a notion of) bisimulation is not clear now. On the other hand, for stable trees nothing of this connection is lost: the τ-laws plus A0-3 axiomatise precisely the notion of $\leftrightarrows_\tau$ for them.

(6) A completeness proof for stable trees was also given in [2], where 'stable' is called 'externally guarded'. That proof uses as an extension of the domain $\mathbb{F}$ of finite trees the domain $\mathbb{H}$ of finite acyclic graphs. The proof is given by 'graph reductions' on the elements of $\mathbb{H}$, e.g. a part as in (a) may be replaced by (b) (cf. the definition of Δ-arc above, in Section 2.1):

(a) 

(b) 

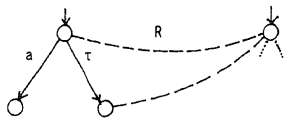
The τ-laws of Milner are, in the signature used in [2] and this paper, slightly different:

$x\tau = x$	T1
$\tau x + x = \tau x$	T2
$a(\tau x + y) = a(\tau x + y) + ax$	T3

(In T3, $a \in A$. The case $\tau(\tau x + y) = \tau(\tau x + y) + \tau x$ is derivable from T2 and A3 ($x + x = x$).)

(7) Instead of either not having $\simeq^C$ correspond directly to a notion of bisimulation (as in (5)) or restricting the trees to stable ones, the following point of view was introduced in [4]: define a restriction of $\leftrightarrows_\tau$, called $\leftrightarrows_{r\tau}$ (Definition 1.3.3 above) by requiring that roots may only be related to roots. The upshot of this restriction is that *initial τ-steps* may not be "contracted" in a bisimulation

R:



That is, initial τ's are treated as if they were not τ's. In effect, the graphs are then stable. One proves easily:

PROPOSITION 6.1.1. *Let* g,h ∈ $\mathbb{G}$. *Let* g',h' *be the results of replacing initial* τ*'s by a fresh symbol* t. *Then :* $g \leftrightarrows_{r\tau} h \Leftrightarrow g' \leftrightarrows_{\tau} h'$. □

Also, $\leftrightarrows_{r\tau}$ is a congruence on the domain of *all* process graphs. In fact, $\leftrightarrows_{r\tau}$ coincides with $\simeq^c$. Thus, the advantage is that again there is a good correspondence between the semantics, $\mathbb{F}(+,\cdot,u)/\leftrightarrows_{r\tau}$ ($u \in A_\tau$), and the syntax, BPA_τ:

BPA_τ

$x+y = y+x$	A1
$(x+y)+z = x+(y+z)$	A2
$x+x = x$	A3
$(x+y)z = xz+yz$	A4
$(xy)z = x(yz)$	A5
$x\tau = x$	T1
$\tau x + x = \tau x$	T2
$a(\tau x + y) = a(\tau x + y) + ax$	T3

Table 6

Namely, $\mathbb{F}(+,\cdot,u)/\leftrightarrows_{r\tau}$ and the initial algebra of BPA_τ are isomorphic.

(8) Of course, a similar result holds for the signature used by Milner. There one has that the process domain $\mathbb{F}'(+,u\cdot)/\leftrightarrows_{r\tau}$ is isomorphic to the initial algebra of

$x+0 = x$	A0
$x+y = y+x$	A1
$(x+y)+z = x+(y+z)$	A2
$x+x = x$	A3
$u\tau x = ux$	T1'
$x+\tau x = \tau x$	T2
$a(x+\tau y) = a(x+\tau y)+ay$	T3

Table 7

(9) The remarks in (1)-(8) all concerned processes corresponding to finite process trees (terms without recursion). It also is the case that for finite process *graphs* (yielding regular processes)

$\leftrightarrows_{r\tau}$ coincides with observational congruence:

PROPOSITION 6.1.2. *Let* $g,h \in \mathbb{R}^{\rho}(+,\cdot,u)$. *Then* $g \leftrightarrows_{r\tau} h \Leftrightarrow \forall C[\,]\ C[g] \leftrightarrows_{\tau} C[h]$.

PROOF. ($\Rightarrow$) $g \leftrightarrows_{r\tau} h \Rightarrow C[g] \leftrightarrows_{r\tau} C[h] \Rightarrow C[g] \leftrightarrows_{\tau} C[h]$. ($\Leftarrow$) Suppose $g \not\leftrightarrows_{r\tau} h$. We must prove: $C[g] \not\leftrightarrows_{\tau} C[h]$ for some context $C[\,]$. If $g \not\leftrightarrows_{\tau} h$ we are done: take the trivial context. Otherwise we are in the situation that $g \not\leftrightarrows_{r\tau} h$ but $g \leftrightarrows_{\tau} h$. For convenience, unwind g,h to the (possibly infinite) trees Tg,Th. Then also $Tg \not\leftrightarrows_{r\tau} Th$, $Tg \leftrightarrows_{\tau} Th$. Now it is easy to prove that either Tg must contain an initial τ-step to a node s such that s in *every* τ-bisimulation from Tg to Th is related to the root of Th, or vice versa (with the role of g,h interchanged). Write

$$(Tg)_S = \Sigma_{i=1,..,n}\, u_i + \Sigma_{j=1,..,m}\, v_j T_j$$

($i,j \geq 0$, $u_i,v_j \in A_\tau$). Next, let $q \in \mathbb{R}^{\rho}$ be such that (i) q contains no τ-steps, (ii) q is not τ-bisimilar to any summand of $(Tg)_S$, i.e. $\Sigma_{i\in X}\, u_i + \Sigma_{j\in Y}\, v_j T_j$ where $X \subseteq \{1,...,n\}$, $Y \subseteq \{1,...,m\}$. Now consider Tg + q and Th + q. These are not τ-bisimilar. For, by (i) the node s in Tg must be related in a supposed τ-bisimulation to the root of Th + q; but this would entail a τ-bisimilarity of q with a summand of $(Tg)_S$ as indicated. Hence also $Tg + q \not\leftrightarrows_{\tau} Th + q$. □

6.2. Recursion together with silent moves

In view of the completeness results mentioned above (in Sections 5 and 6.1), it is a natural question, posed by MILNER [10], whether the 'join' of M and τ-laws (or equivalently, BPA_{LR} and τ-laws) is complete for recursion with silent moves. The answer is *no*, for various reasons.

(1) In the first place, the rule $\mu 2$ in M (or R2 in BPA_{LR}) does not hold for A_τ-guarded recursion equations in the presence of the τ-laws. For, consider $X = a + \tau X$. Even though this recursion equation determines in an intuitively clear sense (which will be made precise below, via the abstraction operator τ_I) a unique process tree, by unfolding, it has *infinitely many solutions*, already in the domain of finite processes. Namely, every $\underline{X} = \tau(a + p)$ for arbitrary p satisfies the equation:

$$\underline{X} = \tau(a + p) =^{*} \tau(a + p) + a = \tau\tau(a + p) + a = \tau\underline{X} + a.$$

Here (*) is by the equation $\tau(x + y) = \tau(x + y) + x$ which follows easily from BPA_τ. As shown below, $\tau(a + p)$ is also the *general* solution of $X = a + \tau X$.

(Remark: already the equation $X = \tau X$ admits infinitely many solutions: $\underline{X} = \tau p$, for arbitrary p.)

So $\mu 2$ (R2) is *false* for A_τ-guarded recursion equations, although it remains sound for A-guarded recursion equations. Therefore we restrict $\mu 2$ (R2) in this way. However, now too much is lost: one does want to prove e.g. $\langle X \mid X = \tau X\rangle = \langle X \mid X = \tau Y + \tau X,\ Y = \tau X\rangle$ (**) (or: $\mu X.\ \tau X = \mu X.(\tau X + \mu Y.\ \tau Y)$), since such equations do not depend on the special nature of τ. Restricting $\mu 2$ (R2) to the A-guarded case would prevent us from proving (**).

To compensate for this loss we use the operator τ_I where $I \subseteq A$ (which was introduced in

[4] on different grounds, namely to be able to distinguish formally between *internal* moves i and *invisible* or silent moves τ; cf. also Section 7.12). This *abstraction operator* τ_I is axiomatised by:

$\tau_I(X) = X$	TI0
$\tau_I(\tau) = \tau$	TI1
$\tau_I(a) = \tau$ if $a \in I$	TI2
$\tau_I(a) = a$ if $a \notin I$	TI3
$\tau_I(x + y) = \tau_I(x) + \tau_I(y)$	TI4
$\tau_I(\tau y) = \tau \cdot \tau_I(y)$	TI5'
$\tau_I(ay) = \tau_I(a) \cdot \tau_I(y)$	TI5"
$\tau_I(<X_1 \mid E>) = <X_1 \mid \tau_I(E)>$	TI6

Table 8

Here $E = \{X_i = T_i(\mathbf{X}) \mid i = 1,...,n\}$, $\mathbf{X} = X_1,...,X_n$ and $\tau_I(E) = \{X_i = \tau_I(T_i(\mathbf{X})) \mid i = 1,...,n\}$. Now equation (**) above can be proved: Let $i \in A$ be a fresh symbol, acting as a 'stand-in' for τ. Write $\tau_{\{i\}}$ as τ_i. Then

$$\vdash <X \mid X = iX> = <X \mid X = iY + iX, Y = iX>$$

by using R2 for the restricted case of A-guarded recursion equations. Hence

$$\vdash \tau_i<X \mid X = iX> = \tau_i<X = iY + iX, Y = iX>$$

and by TI6:

$$\vdash <X \mid X = \tau_i(iX)> = <X \mid X = \tau_i(iY + iX), Y = \tau_i(iX)>.$$

So $\vdash$ (**).

(2) Even with this restriction of R2 and addition of TI0-6 the proof system would not be complete. Namely, the equation

$$<X \mid X = \tau X> = \tau \qquad (***)$$

(or $\mu X.\ \tau X = \tau 0$ in Milner's signature) is *valid* as the corresponding graphs are clearly $r\tau$-bisimilar. However, (***) is not provable with the proof system proposed thus far. We will not rigorously prove this incompleteness, but sketch a proof. In the notion of $\leftrightarrows_{r\tau}$ and $\leftrightarrows_\tau$ a choice is made (as MILNER [9] also remarks) of treating the possibly infinite execution of a τ-loop ...○⟲τ as if a fairness condition was imposed: viz. that after some finite number of loop executions no further executions of it are performed, and either *an alternative is chosen* (as is possible in e.g. ...$_\tau$○⟲^{a}→○) or we have *succesful termination*. Here a different choice could be made, if a constant δ denoting deadlock or failure is present as in ACP or ACP_τ (see [4]). Then a τ-loop without alternative can be treated as δ. That is, there is a notion of $r\tau\delta$-bisimulation ($\leftrightarrows_{r\tau\delta}$) in which

→○$\xrightarrow{a}$○⟲$_\tau$ $\leftrightarrows_{r\tau\delta}$ aδ.

We will not define $\leftrightarrows_{r\tau\delta}$ here more precisely, but state merely that process graphs modulo this

notion of bisimulation are also a model, $\mathcal{F} = \mathbb{F}(+,\cdot,u,\delta)/ \leftrightarrows_{r\tau\delta}$ $(u \in A_\tau)$, of the proof system proposed thus far, $P = BPA^*_{LR}$ + T1-3 + TI0-6 where * denotes the restriction of R2 to A-guarded equations. Now $\mathcal{F} \models a\cdot\langle X \mid X = \tau X\rangle = a\delta \neq a\tau = a$, and hence $P \not\vdash a\cdot\langle X \mid X = \tau X\rangle = a\tau$, i.e.$P \not\vdash$ (***).

We will now present, in Table 9, a proof system $BPA_{\tau LR}$ which is claimed to be complete for $\mathbb{F}(+,\cdot,u)/ \leftrightarrows_{r\tau}$. In the table the following notation is used: $a \in A$, $I \subseteq A$; further E will always be $\{X_i = T_i(\mathbf{X}) \mid i = 1,...,n\}$, where the terms $T_i(\mathbf{X})$ are linear and may contain variables from $\mathbf{X} = X_1,...,X_n$ but no other. We write $\tau_I(E)$ for $\{X_i = \tau_I(T_i(\mathbf{X})) \mid i = 1,...,n\}$ and $E_{-k} = E - \{X_k = T_k(\mathbf{X})\}$. With $BPA_{\tau I}$ the proof system without the recursion part is meant; that is: A1-5, T1-3, TI0-6. Now $BPA_{\tau I} \vdash E = E'$ as in the premiss of R3 in the table below denotes a conversion of some of the $T_i(\mathbf{X})$ in E to $T_i'(\mathbf{X})$ by means of the axioms in $BPA_{\tau I}$, result: E'.

$BPA_{\tau LR}$

$x+y = y + x$	A1
$(x + y) + z = x + (y + z)$	A2
$x + x = x$	A3
$(x + y)z = xz + yz$	A4
$(xy)z = x(yz)$	A5
$x\tau = x$	T1
$\tau x + x = \tau x$	T2
$a(\tau x + y) = a(\tau x + y) + ax$	T3
$\tau_I(X) = X$	TI0
$\tau_I(\tau) = \tau$	TI1
$\tau_I(a) = \tau$ if $a \in I$	TI2
$\tau_I(a) = a$ if $a \notin I$	TI3
$\tau_I(x + y) = \tau_I(x) + \tau_I(y)$	TI4
$\tau_I(\tau y) = \tau \cdot \tau_I(y)$	TI5'
$\tau_I(ay) = \tau_I(a) \cdot \tau_I(y)$	TI5"
$\tau_I(\langle X_1 \mid E\rangle) = \langle X_1 \mid \tau_I(E)\rangle$	TI6
$\dfrac{x_i = \langle X_i \mid E\rangle,\ i = 1,...,n}{x_1 = T_1(x)}$	R1
$\dfrac{x_i = T_i(x),\ i = 1,...,n}{x_1 = \langle X_1 \mid E\rangle}$ $T_i(\mathbf{X})$ is A-guarded	R2
$\dfrac{BPA_{\tau I} \vdash E = E'}{\langle X_1 \mid E\rangle = \langle X_1 \mid E'\rangle}$	R3
$\dfrac{\tau\langle X_k \mid E\rangle = \tau\langle X_{k'} \mid E\rangle \text{ for some } k,k' \neq 1}{\langle X_1 \mid E\rangle = \langle X_1 \mid E_{-k}, X_k = T_k(\mathbf{X}) + \tau X_{k'}\rangle}$	R4
$\langle X_1 \mid E_{-k}, X_k = \tau X_k\rangle = \langle X_1 \mid E_{-k}, X_k = \tau\rangle$	R5

Table 9

6.2.1. Discussion of $BPA_{\tau LR}$

Axioms A1-5, T1-3, TI0-6 have been discussed above. R1,R2 were already present in BPA_{LR}; here with the proviso on R2 discussed above. R3 says that conversions may take place inside the bodies of an LR-expression. In the case of BPA_{LR} this rule was provable; here it is not, if the conversions in the $T_i(\mathbf{X})$ of $E = \{X_i = T_i(\mathbf{X}) \mid i = 1,...,n\}$ are applications of the τ-laws T1-3 or TI0-6. (For applications of A1-5 we do not need R3, in fact. Then applying R1,2 and TI0-6 suffice.)

EXAMPLE 6.2.1.1. To prove: $\langle X \mid X = aY + aZ, Y = \tau Z + b, Z = cZ\rangle = \langle X \mid X = aY, Y = \tau Z + b, Z = cZ\rangle$. (See Figure 39(a),(b).)

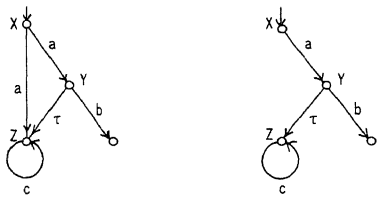


Figure 39 (a) (b)

PROOF. Let i be a 'fresh' atom. Let $\underline{X}^i \equiv \langle X \mid X = aY + aZ, Y = iZ + b, Z = cZ\rangle \equiv \langle X \mid E\rangle$, and $\underline{Y}^i \equiv \langle Y \mid E\rangle$, $\underline{Z}^i \equiv \langle Z \mid E\rangle$. Then

$$\vdash \underline{X}^i = a\underline{Y}^i + a\underline{Z}^i,\ \underline{Y}^i = i\underline{Z}^i + b,\ \underline{Z}^i = c\underline{Z}^i. \qquad (R1)$$

$$\vdash \underline{X}^i = a(i\underline{Z}^i + b) + a\underline{Z}^i,\ \underline{Z}^i = c\underline{Z}^i.$$

$$\vdash \underline{X}^i = \langle X \mid X = a(iZ + b) + aZ, Z = cZ\rangle \qquad (R2)$$

$$\vdash \tau_{\{i\}}(\underline{X}^i) = \tau_{\{i\}}(\langle X \mid X = a(iZ + b) + aZ, Z = cZ\rangle)$$

$$\vdash \underline{X} = \langle X \mid X = a(\tau Z + b) + aZ, Z = cZ\rangle$$

Here $\underline{X}$ is the LR-expression in the LHS of the identity to prove. Further:

$$\vdash \underline{X} = \langle X \mid X = a(\tau Z + b), Z = cZ\rangle \qquad (T3)$$

Finally, let $\underline{X}_i \equiv \langle X \mid X = a(iZ + b), Z = cZ\rangle \equiv \langle X \mid F\rangle$ and $\underline{Z}_i \equiv \langle Z \mid F\rangle$, then

$$\vdash \underline{X}_i = \langle X \mid X = aY, Y = iZ + b, Z = cZ\rangle \qquad (R1,R2)$$

Hence: $\vdash \underline{X} = \langle X \mid X = aY, Y = \tau Z + b, Z = cZ\rangle$ (TI0-6) □

The axiom R5 was already discussed above and reflects the bias towards fairness w.r.t. τ-steps of $\leftrightarrows_{r\tau}$. We should note here that in a more definitive treatment of LR-expressions in which nesting of such expressions is allowed, R5 could be simplified to: $\langle X \mid X = \tau X\rangle = \tau$.

The rule R4, finally, is the only one of the axioms and rules in $BPA_{\tau LR}$ which, it seems, cannot very well be expressed in the μ-formalism of Milner's M. Its role is to enable one to insert τ-steps between τ-bisimilar (non-root) nodes. A special case of R4 is the case that $k = k'$: then we have the axiom $\langle X_1 \mid E\rangle = \langle X_1 \mid E_{-k}, X_k = T_k(\mathbf{X}) + \tau X_k\rangle$ which enables one to provably append a τ-loop at 'node' X_k ($k \neq 1$). The rule R4 could be called 'internal τ-introduction', where 'internal' refers to the fact that $k,k' \neq 1$ in the premiss of the rule (i.e. the 'nodes' $X_k, X_{k'}$ are non-root

nodes, or 'internal' nodes). To profit from R4 we need also the following version R4* which is slightly weaker but in which the asymmetry in R4 is removed:

LEMMA 6.2.2. (General τ-introduction) *The following rule is provable in* $BPA_{\tau LR}$*:*

$$\frac{\tau\langle X_k \mid E\rangle = \tau\langle X_{k'} \mid E\rangle \text{ for some } k,k'}{\tau\langle X_1 \mid E\rangle = \tau\langle X_1 \mid E_{-k}, X_k = T_k(\mathbf{X}) + \tau X_{k'}\rangle} \quad \text{R4*}$$

PROOF. (Note that by the symmetry in the premiss of R4* we have as a consequence even: $\tau\langle X_i \mid E\rangle = \tau\langle X_i \mid E_{-k}, X_k = T_k(\mathbf{X}) + \tau X_{k'}\rangle$ for all i = 1,...,n.) Now consider $\langle X_0 \mid X_0 = \tau X_1, E\rangle$ where $E = \{X_i = T_i(X_1,...,X_n) \mid i = 1,...,n\}$. The 'nodes' $\langle X_k \mid E\rangle$ and $\langle X_{k'} \mid E\rangle$ from the premiss of R4* are internal w.r.t. $\langle X_0 \mid X_0 = \tau X_1, E\rangle$. Hence R4 applies and yields, acting on the same premiss: $\vdash \langle X_0 \mid X_0 = \tau X_1, E\rangle = \langle X_0 \mid X_0 = \tau X_1, E_{-k}, X_k = T_k(\mathbf{X}) + \tau X_{k'}\rangle$.
Now clearly $\vdash \langle X_0 \mid X_0 = \tau X_1, E\rangle = \tau\langle X_1 \mid E\rangle$
and $\vdash \langle X_0 \mid X_0 = \tau X_1, E_{-k}, X_k = T_k(\mathbf{X}) + \tau X_{k'}\rangle = \tau\langle X_1 \mid E_{-k}, X_k = T_k(\mathbf{X}) + \tau X_{k'}\rangle$.
Hence $\vdash \tau\langle X_1 \mid E\rangle = \tau\langle X_1 \mid E_{-k}, X_k = T_k(\mathbf{X}) + \tau X_{k'}\rangle$. □

EXAMPLE 6.2.3. We want to prove that the LR-expressions corresponding to the following graphs are equal:

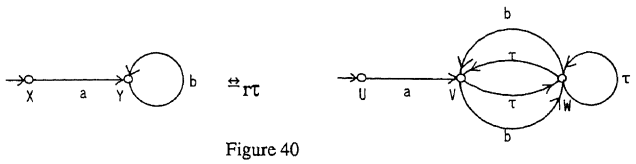


Figure 40

So to prove: $\langle X \mid X = aY, Y = bY\rangle = \langle U \mid U = aV, V = bW + \tau W, W = bV + \tau V + \tau W\rangle$.

PROOF.
By R1,R2: $BPA_{\tau LR} \vdash \langle X \mid X = aY, Y = bY\rangle = \langle U \mid U = aV, V = bW, W = bV\rangle$.
Abbreviate $\underline{U} \equiv \langle U \mid E\rangle$, $\underline{V} \equiv \langle V \mid E\rangle$, $\underline{W} \equiv \langle W \mid E\rangle$
where $E = \{U = aV, V = bW, W = bV\}$.
Now $\vdash \underline{V} = \underline{W}$,
hence $\vdash \tau\underline{V} = \tau\underline{W}$,
hence by R4: $\vdash \underline{U} = \langle U' \mid U' = aV', V' = bW' + \tau W', W' = bV'\rangle$ $(\equiv \underline{U}')$.
By R4* in Lemma 6.2.2,
$\vdash \tau\underline{V} = \tau\underline{V}', \tau\underline{W} = \tau\underline{W}'$.
Hence $\vdash \tau\underline{V}' = \tau\underline{W}'$.
By R4: $\vdash \underline{U}' = \underline{U}'' \equiv \langle U'' \mid U'' = aV'', V'' = bW'' + \tau W'', W'' = bV'' + \tau V''\rangle$.
Finally by R4, $\vdash \underline{U}'' = \langle U''' \mid U''' = aV''', V''' = bW''' + \tau W''', W''' = bV''' + \tau V''' + \tau W'''\rangle$. □

DEFINITION 6.2.4. The *semantics* of $BPA_{\tau LR}$ is defined, analogously to that of BPA_{LR}, via an

intermediate semantics [ ]: $Ter_c(BPA_{\tau LR}) \to \mathbb{R}(+,\cdot,\tau_I,u)$ $(u \in A_\tau)$, with as only extra clause that $[\tau_I(T)] = \tau_I([T])$. Here τ_I in the RHS is the operator on graphs renaming the $a \in I$ into τ. Further, for $T \in Ter_c(BPA_{\tau LR})$ we define $[T] = [T]/\leftrightarrows_{r\tau}$.

For the completeness proof of the next theorem we need the following lemma stating that the operation Δ which makes a graph Δ-saturated and the operation E on Δ-saturated graphs (see Sections 2.1 and 2.2), are "provable in $BPA_{\tau LR}$".

LEMMA 6.2.5. (i) *For every canonical LR-expression* T *there is a canonical LR-expression* T' *such that* $BPA_{\tau LR} \vdash T = T'$ *and* $\Delta([T]) = [T']$.
(ii) *For every canonical LR-expression* T' *such that* [T'] *is* Δ*-saturated, there is a canonical LR-expression* T'' *such that* $BPA_{\tau LR} \vdash T' = T''$ *and* $E([T']) = [T'']$.
(iii) *(Combining (i) and (ii):) For every canonical LR-expression* T *there is a canonical LR-expression* T'' *such that* $BPA_{\tau LR} \vdash T = T''$ *and* $[T''] = E(\Delta([T]))$.

PROOF. (i) The operation Δ consists of successively adding edges to form Δ-arcs. As Example 6.2.1.1 shows, each such addition is provable in $BPA_{\tau LR}$.
(ii) Using R4 and R4* in Lemma 6.2.2, one can (as illustrated in Example 6.2.3) provably add (or omit) "ε-steps" as they were called in Section 2.2. □

For the proof of the completeness theorem 6.2.7 we need the following simple fact:

PROPOSITION 6.2.6. *Let* T,S *be canonical LR-expressions. Then:*

$$BPA_{LR} \vdash T = S \;\Rightarrow\; BPA_{\tau LR} \vdash T = S. \qquad \square$$

THEOREM 6.2.7. *Let* $T,S \in Ter_c(BPA_{\tau LR})$. *Then:*

$$BPA_{\tau LR} \vdash T = S \;\Leftrightarrow\; [T] \leftrightarrows_{r\tau} [S] \;\Leftrightarrow\; [T] = [S].$$

PROOF. *Soundness*. Part of the soundness proof is identical to that for BPA_{LR} in Theorem 5.3.2, another part is a consequence of properties of $\leftrightarrows_{r\tau}$. We will not work out the tedious details here.
Completeness. Suppose [T] = [S]. We may suppose T,S are canonical LR-expressions. Here we use soundness of $BPA_{\tau LR}$, and the same procedure as for BPA_{LR} (in Theorem 5.3.2) for eliminating products of LR-expressions, this time with the additional help of R5 in Table 9. Namely: suppose the node U has as subgraph a bunch of possibly intersecting τ-cycles, i.e. from U there is no terminating path and all paths from U contain only τ-steps. See Figure 41(a). Then, using τ_I, R1, R2 we replace this subgraph by one τ-loop: see Figure 41(b). Now R5 removes this τ-loop (Figure 41(c)), after which a right factor can be appended.

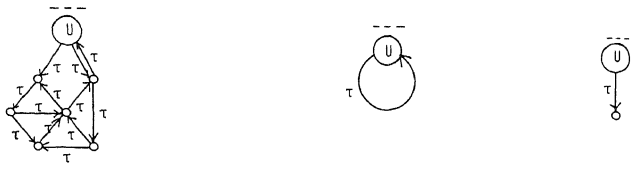
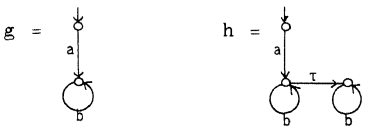


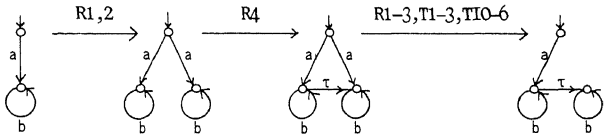
Figure 41 (a) (b) (c)

Now consider the hypothesis $[T] \leftrightarrows_{r\tau} [S]$. By Corollary 2.4, $E\Delta[T] \leftrightarrows E\Delta[S]$ (ordinary bisimulation). By Lemma 6.2.5(iii), there are T", S" provably equal to T,S respectively such that $[T"] = E\Delta[T]$ and $[S"] = E\Delta[S]$. Hence $[T"] \leftrightarrows [S"]$. So, by the completeness theorem (5.3.2) for BPA_{LR} we have: $BPA_{LR} \vdash T" = S"$. By Proposition 6.2.6: $BPA_{\tau LR} \vdash T" = S"$. Hence $BPA_{\tau LR} \vdash T = S$. □

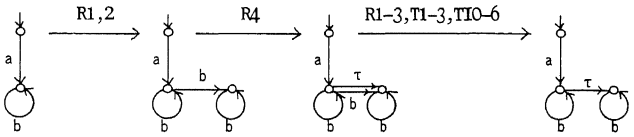
EXAMPLE 6.2.8. We conclude this section with one more example of a proof in $BPA_{\tau LR}$. To prove the equality of the terms corresponding to graphs g,h:



A possible proof employs the following transformations (of the corresponding expressions):



Another proof uses the transformation:



7. Solving systems of A_τ-guarded recursion equations

As already remarked in the previous section, a system of recursion equations that are A_τ-guarded (rather than A-guarded as in Section 5) need not have a unique solution, e.g. $X = a + \tau X$ admits infinitely many solutions in $\mathbb{R}^{\rho}(+,\cdot,u) / \leftrightarrows_{r\tau}$. We will now determine the *solution set* of such

systems of A_τ-guarded recursion equations. To this end we employ the following theorem (7.3).

DEFINITION 7.1. Let $\langle X_1 \mid E\rangle \equiv \langle X_1 \mid \{X_i = T_i(X) \mid i = 1,...,n\}\rangle$ be a LR-expression. We say that closed terms $M_i \in \mathrm{Ter}(\mathrm{BPA}_{\tau LR})$ solve $\langle X_1 \mid E\rangle$ if $[\![M_i]\!] = [\![T_i(\mathbf{M})]\!]$, or equivalently, $[M_i] \leftrightarrows_{r\tau} [T_i(\mathbf{M})]$ $(i = 1,...,n)$. Likewise we say that the process graphs $g_i \in \mathbb{R}^\rho(+,\cdot,u)$ solve $\langle X_1 \mid E\rangle$ if $g_i \leftrightarrows_{r\tau} T_i(\mathbf{g})$, $i = 1,...,n$. Here $\mathbf{M} = M_1,...,M_n$ and $\mathbf{g} = g_1,...,g_n$.

DEFINITION 7.2. Let $\langle X_1 \mid E\rangle$ be a canonical LR-expression. Then $\langle X_1 \mid E\rangle$ is τ-*cycle free* if the process graph $[\langle X_1 \mid E\rangle]$ does not contain a τ-cycle.

THEOREM 7.3. *Let* $\langle X_1 \mid E\rangle$ *be a canonical* τ*-cycle free LR-expression. Then* $\langle X_1 \mid E\rangle$ *has a unique solution in* $\mathbb{R}^\rho(+,\cdot,u) / \leftrightarrows_{r\tau}$.

PROOF. The existence of a solution is clear. Now suppose that the graphs $g_i \in \mathbb{R}^\rho(+,\cdot,u)$ solve $\langle X_1 \mid E\rangle$.

Claim 1. Without loss of generality we may suppose that $\langle X_1 \mid E\rangle$ has no *loops*, i.e. no cycles of length 1. (This assumption is not essential but simplifies the proof somewhat.)

Proof of claim 1. Suppose $\langle X_1 \mid E\rangle$ has a loop: say E contains the equation $X_i = aX_i + \ldots$. Then we transform $\langle X_1 \mid E\rangle$ to the LR-expression $\langle X_1 \mid E'\rangle$ by introducing a new variable X_i', replacing every occurrence of cX_i in E by $cX_i + cX_i'$, replacing the equation $X_i = aX_i + \ldots$ by equations $X_i = aX_i' + \ldots$, $X_i' = aX_i + \ldots$. Graphically, this amounts to placing a new node X_i' on the a-loop from X_i to itself and copying for X_i' the in- and out-edges which X_i has, i.e. if $X_i \rightarrow^u X_k$ then $X_i' \rightarrow^u X_k$ and if $X_i \leftarrow^u X_k$ then $X_i' \leftarrow^u X_k$. (In fact, this procedure is just an example of *expansion* as in 5.3.5.) Now if $\langle X_1 \mid E\rangle$ has two different solutions, clearly also $\langle X_1 \mid E'\rangle$ has two different solutions. Hence to prove unique solvability of $\langle X_1 \mid E\rangle$ it suffices to prove this for $\langle X_1 \mid E'\rangle$. *End of proof of claim 1.*

So suppose we have the system of bisimilarities $\{g_i \leftrightarrows_{r\tau} T_i(\mathbf{g}) \mid i = 1,...,n\}$ (*). We want to prove that $g_i \leftrightarrows_{r\tau} [\langle X_i \mid E\rangle]$, the 'canonical' solution of $\langle X_1 \mid E\rangle$. Note that by the definition in 5.3.1, $[\langle X_1 \mid E\rangle]$ is root-unwound. By symmetry, it suffices to prove the case for $i = 1$.

Now we construct a process graph G which is an amalgam of the process graph $[\langle X_1 \mid E\rangle]$ and the process *trees* $T(g_i)$, $i = 1,...,n$. Here $T(g_i)$ is g_i completely unwound to a tree. G is constructed by glueing the root r_i of $T(g_i)$ and the node X_i in $[\langle X_1 \mid E\rangle]$ together. For the exposition, the part of G originating from $[\langle X_1 \mid E\rangle]$ is drawn in a horizontal plane and the appended $T(g_i)$, $i = 1,...,n$, are hanging downwards as in Figure 42.

In the example of the figure, the system of bisimilarities (*) (or rather the root-unwound version w.r.t. g_1) is $\{g_1 \leftrightarrows_{r\tau} ag_3 + cg_2,\ g_2 \leftrightarrows_{r\tau} ag_4,\ g_3 \leftrightarrows_{r\tau} bg_4 + \tau g_2,\ g_4 \leftrightarrows_{r\tau} \tau g_3\}$. We call the part of G consisting of $aT(g_3) + cT(g_2)$, the *successor graph* of $T(g_1)$, likewise the part of G consisting of $aT(g_4)$ is the successor graph of g_2, etc. The assumption given by the system (*) of bisimilarities entails that each $T(g_i)$ is $r\tau$-bisimilar to its successor graph. (This follows since the operation T respects $\leftrightarrows_{r\tau}$ and $\leftrightarrows_{r\tau}$ is a congruence w.r.t. $\cdot$ and $+$.)

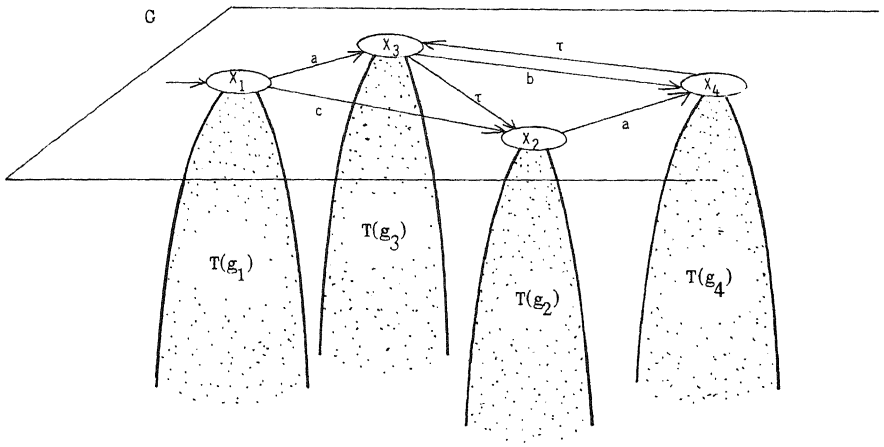


Figure 42

Now choose an rτ-bisimulation from $T(g_i)$ to its successor graph and connect each node s in $T(g_i)$ by an arrow ($\rightarrowtail$) to the nodes t in the successor graph whenever s,t are related by the chosen bisimulation. Call the $\longrightarrow$-paths originating in this way in G, *bisimulation threads*. For a bisimulation thread $t \rightarrowtail t' \rightarrowtail t'' \rightarrowtail \ldots \rightarrowtail t^{(n)}$ we also write $t \rightarrowtail t^{(n)}$.

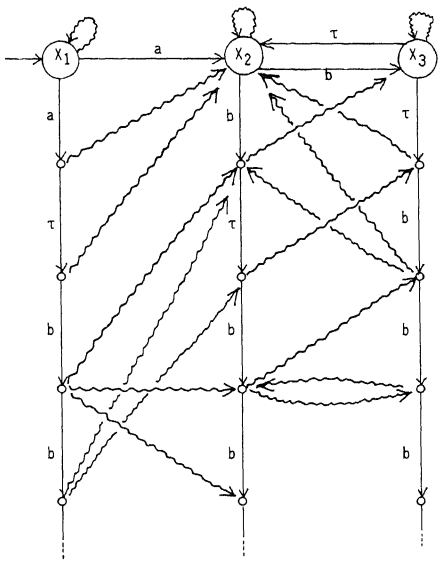
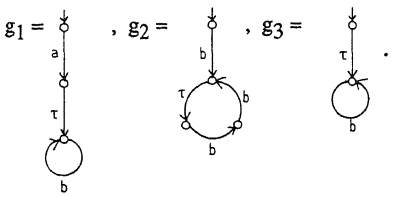


Figure 43

EXAMPLE. Let the system (*) be $\{g_1 \triangleq_{r\tau} ag_2, g_2 \triangleq_{r\tau} bg_3, g_3 \triangleq_{r\tau} tg_2\}$, and



Then G together with some bisimulation threads is as in Figure 43 above.

Claim 2. (i) From every node s in $T(g_1)$ there is a bisimulation thread leading to a node X_i in G.
(ii) Let $\mathcal{R}$ be the relation between NODES($T(g_1)$) and the process graph $[<X_1 \mid E>]$ (in the horizontal plane) given by (i), i.e.

$s \mathcal{R} X_i \Leftrightarrow s \rightarrow\!\!\!\!\!\sim^{\pi} X_i$ for some bisimulation thread π.

Then R is a $r\tau$-bisimulation between $T(g_1)$ and $[<X_1 \mid E>]$.

With Claim 2(ii) we are done, since then $g_1 \triangleq_{r\tau} T(g_1) \triangleq_{r\tau} [<X_1 \mid E>]$. In order to prove the claim we need two concepts:

- the *depth* of a node s in a process tree T is the number of steps which it takes to reach s from the root r,
- the *external depth* of a node s in a process tree T is the number of non-τ-steps it takes to reach s from the root r.

Proof of Claim 2(i): Let $s \in$ NODES($T(g_1)$), and let π: $r_1 \twoheadrightarrow s$ be the (unique) path in $T(g_1)$ from its root r_1 to s. Then there is a path π^* from the root of the successor graph of $T(g_1)$ to some s' such that $\pi \equiv_\tau \pi^*$ (i.e. π,π^* are externally equivalent, that is: determine the same words over A after skipping τ's). See Figure 44(a). The first step of this path π^* is "horizontal". Leaving out this first step of π^* we have a vertical path π' in some $T(g_i)$.

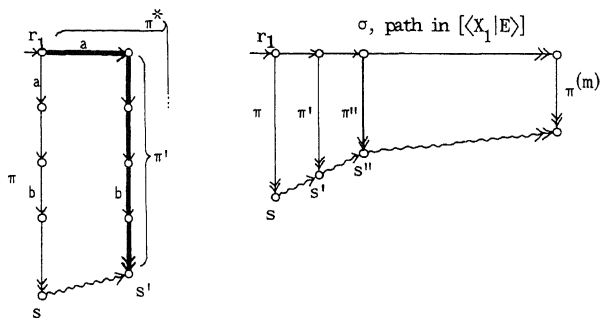


Figure 44 (a) (b)

Continuing this procedure, we find vertical paths $\pi,\pi',\pi'', \ldots, \pi^{(m)}$ (see Figure 44(b)) connected by a horizontal path σ such that $\pi \equiv_\tau \sigma\pi^{(m)}$. Since σ can be prolonged arbitrarily, and *since the horizontal graph* $[<X_1 \mid E>]$ *was supposed to be* τ*-cycle free*, eventually all non-τ-steps in π will be 'absorbed' by a horizontal path σ. I.e. for some m, $\pi^{(m)}$ consists only of τ-steps. In other words, we have pushed s upwards to external depth 0. Further we can get at the "surface" (depth 0) since eventually a node X_i must be reached in the upper plane from where no τ-step is possible. Then we have arrived at a situation as in Figure 45(a), which proves Claim 2(i).

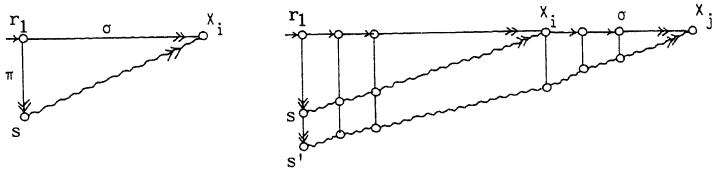


Figure 45 (a) (b)

Proof of Claim 2(ii) is straightforward. (See Figure 45(b).) Given a path π: s $\twoheadrightarrow$ s' in $T(g_1)$, and a bisimulation thread from s to some X_i, we find a path σ: $X_i \twoheadrightarrow X_j$ for some X_j by following a bisimulation thread 'below' the one from s to X_i. The same argument applies for the other direction: given a path σ: $X_i \twoheadrightarrow X_j$ we find going backwards a bisimulation thread below the one from s to X_i ending up in a point s' below s. Hence bisimulation threads constitute an $r\tau$-bisimulation between $T(g_1)$ and $[<X_1 \mid E>]$. □

REMARK 7.3.1. The following example shows that if the condition of absence of τ-cycles is omitted, it is indeed not always possible to 'surface' via a bisimulation thread. Let $g_1 = g_2 = g_3 = \tau(a + b)$. Then g_1,g_2,g_3 solve $<X \mid X = \tau Y, Y = a + \tau Z, Z = \tau Y>$, i.e. $g_1 \triangleq_{r\tau} \tau g_2$, $g_2 \triangleq_{r\tau} a + \tau g_3$ and $g_3 \triangleq_{r\tau} \tau g_2$. Now G together with all possible bisimulation threads is as in Figure 46. Indeed the end-point of the b-step cannot be related with a node X,Y,Z.

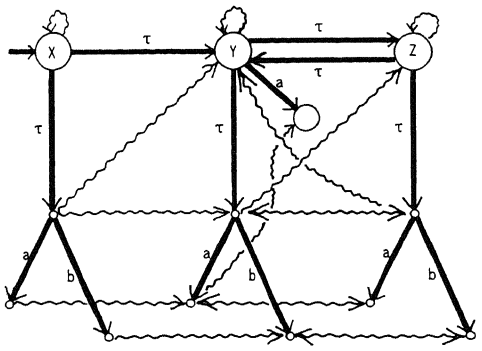


Figure 46

7.4. In order to formulate the general solution of A_τ-guarded systems of recursion equations $\langle X_1 \mid E\rangle$, we first perform a syntactical transformation on $\langle X_1 \mid E\rangle$. The system $\langle X_1 \mid E\rangle$ is supposed to be canonical.

Call two nodes X_i, X_j $(i,j = 1,...,n)$ τ-*cyclic equivalent* if there is a τ-cycle through X_i, X_j in the corresponding graph. (Special case: if $i = j$ and X_i supports a τ-loop.) *Notation:* $X_i \approx_\tau X_j$. Further, call a node X_i τ-*cyclic* if $\exists j\ X_i \approx_\tau X_j$. So $\approx_\tau$ is an equivalence relation on the set of τ-cyclic nodes (not on the set of all nodes).

Now transform $\langle X_1 \mid E\rangle$ as follows:

Step 1. Partition the nodes $X_1,...,X_n$ into τ-cyclic and non-τ-cyclic nodes.

Step 2. Partition the τ-cyclic nodes in τ-cyclic equivalence classes. Denote these equivalence classes by $X_i/\approx_\tau$.

Step 3. Remove in the RHS of equation $X_i = T_i(\mathbf{X})$ all summands τX_j such that $X_i \approx_\tau X_j$ in the original system $\langle X_1 \mid E\rangle$. Result: $\langle X_1 \mid E'\rangle$.

Step 4. Let X_i be τ-cyclic (in the original system). Add to the RHS of $X_i = T_i'(\mathbf{X})$ in E', all RHS's of the equations $X_j = T_j'(\mathbf{X})$ in E' whenever $X_i \approx_\tau X_j$ in the original E, plus an *arbitrary* Q (fixed for one τ-cyclic equivalence class). Prefix the result with τ.

EXAMPLE 7.4.11(i) $\langle X \mid E\rangle$ where $E = \{X = aY + bZ,\ Y = bY + \tau Z,\ Z = aZ + \tau Y\}$. So $Y \approx_\tau Z$. Now denote a $\approx_\tau$-class by a box:

$$E:\ \begin{array}{l} X = aY + bZ \\ \boxed{\begin{array}{l} Y = bY + \tau Z \\ Z = aZ + \tau Y \end{array}} \end{array}$$

This system is transformed to

$$E_Q:\ \begin{array}{l} X = aY + bZ \\ \boxed{\begin{array}{l} Y = \tau(bY + aZ\ (+Q)) \\ Z = \tau(bY + aZ\ (+Q)) \end{array}} \end{array}$$

(ii) $X = \tau X$ is transformed to $\boxed{X = \tau Q}$

(iii) $X = \tau X + a$ is transformed to $\boxed{X = \tau(a\ (+Q))}$

(iv) $X = \tau(X + a)$ has the same general solution as $X = \tau X + a$: first transform $X = \tau(X + a)$ to $\{X = \tau Y,\ Y = \tau Y + a\}$ which has general solution $\{X = \tau Y,\ Y = \tau(a\ (+Q))\}$, hence $X = \tau\tau(a\ (+Q)) = \tau(a\ (+Q))$.

(v) $X_1 = a + bX_2 + \tau X_3$
$X_2 = \tau X_2 + a$
$X_3 = a + bX_1 + \tau X_4$

$X_4 = c + dX_2 + \tau X_5 + \tau X_3$
$X_5 = d + aX_4 + \tau X_3 + \tau X_5 + \tau X_4$

is transformed to

$X_1 = a + bX_2 + \tau X_3$
$X_2 = \tau(a\,(+Q_1))$
$X_3 = \tau(a + c + d + bX_1 + dX_2 + aX_4\,(+Q_2))$
$X_4 = X_3$
$X_5 = X_3$

(vi) The system $\{X = \tau(a(\tau X + Y) + b),\ Y = \tau(\tau aX + \tau Y)\}$ has the general solution $\{X = \tau(a(\tau X + Y) + b),\ Y = \tau(\tau aX\ (+\ Q))\}$ (Q arbitrary) as can be seen via conversion to a canonical system and back. The last system, parametrised by Q, has a unique solution by the following theorem.

NOTATION 7.4.12. The transformation of $\langle X_1 \mid E\rangle$ will be written as $\langle X_1 \mid E_{\mathbf{Q}}\rangle$ where $\mathbf{Q} = Q_1,\ldots,Q_k$ are arbitrary closed terms occurring, as in Step 4 of the transformation procedure, in the $\approx_\tau$-classes $X_{i1}/\approx_\tau, \ldots, X_{ik}/\approx_\tau$.

Now the results $\langle X_1 \mid E_{\mathbf{Q}}\rangle$ of this transformation are τ-*cycle free* systems of recursion equations. Hence, by the preceding theorem, they have unique solutions. Par abus de langage, let us denote these solutions also by $\langle X_1 \mid E_{\mathbf{Q}}\rangle$.

A useful generalisation of the preceding theorem can be phrased in terms of the following concept:

DEFINITION 7.5. A term $T(\mathbf{X})$ containing no other variables than those in $\mathbf{X}$, is *essentially* A-*guarded* if every occurrence of X is preceded by some $a \in A$. More precisely:
(i) closed terms (i.e. without variables) are essentially A-guarded,
(ii) for every term S and $a \in A$, aS is essentially A-guarded,
(iii) if S_1, S_2 are essentially A-guarded then so is $S_1 + S_2$,
(iv) if S is essentially A-guarded then so is $S \cdot T$ for all T.

EXAMPLE 7.6. $\tau + \tau(a\tau XY + b\tau X)$ is essentially A-guarded.

THEOREM 7.7. *Let* $E = \{X_i = T_i(X_1,\ldots,X_n) \mid i = 1,\ldots,n\}$ *be a system of essentially* A-*guarded, linear recursion equations. Then* E *has a unique solution in* $\mathbb{R}^\rho(+,\cdot,u)/\hat{=}_{r\tau}$ $(u \in A_\tau)$.

PROOF. Converting E to a canonical system of recursion equations yields a system which is τ-cycle free. □

EXAMPLE 7.8. $\{X = \tau + \tau(a\tau X + b\tau(X + \tau Y)), Y = a + \tau(aX + \tau bY)\}$ converts to $\{X = \tau + \tau U, U = aV + bW, V = \tau X, W = \tau Z, Z = \tau + \tau U + \tau Y, Y = a + \tau A, A = aX + \tau B, B = bY\}$, which has no τ-cycles.

Now we arrive at the main theorem of this section:

THEOREM 7.9. *Let* $\langle X_1 \mid E\rangle$ *be a canonical LR-expression, so containing* A_τ*-guarded recursion equations. Then every solution of* $\langle X_1 \mid E\rangle$ *in* $\mathbb{R}^\rho(+,\cdot,u)/\leftrightarrows_{\tau\tau}$ $(u \in A_\tau)$ *is of the form* $\langle X_1 \mid E_Q\rangle$ *for some* Q*, and vice versa. In particular, if* $\langle X_1 \mid E\rangle$ *is* τ*-cycle free,* Q *is empty and the solution is unique.*

PROOF SKETCH. We sketch the proof by demonstration on Example 7.4.1(i) above: Let $\underline{X}, \underline{Y}, \underline{Z}$ be the unique solution of $\langle X \mid E_Q\rangle$. So

$$\underline{X} = a\underline{Y} + b\underline{Z}$$
$$\underline{Y} = \tau(b\underline{Y} + a\underline{Z} (+ Q))$$
$$\underline{Z} = \tau(b\underline{Y} + a\underline{Z} (+ Q)).$$

Now $\underline{X},\underline{Y},\underline{Z}$ is a solution of $\langle X \mid E\rangle$:

$$\underline{X} = a\underline{Y} + b\underline{Z}$$
$$\underline{Y} = \tau(b\underline{Y} + a\underline{Z} (+ Q)) = b\underline{Y} + \tau(b\underline{Y} + a\underline{Z} (+ Q)) = b\underline{Y} + \tau\tau(b\underline{Y} + a\underline{Z} (+ Q)) = b\underline{Y} + \tau\underline{Z}$$
$$\underline{Z} = \ldots = a\underline{Z} + \tau\underline{Y}.$$

Vice versa, let X^*,Y^*,Z^* solve $\langle X \mid E\rangle$, then for some Q we have that X^*,Y^*,Z^* are the unique solution of $\langle X \mid E_Q\rangle$:

$$X^* = aY^* + bZ^*$$
$$Y^* = bY^* + \tau Z^* = bY^* + \tau(aZ^* + \tau Y^*) = bY^* + \tau(aZ^* + bY^* + \tau Y^*) =$$
$$= \tau(aZ^* + bY^* + \tau Y^*) = \tau(aZ^* + bY^* + Q)$$
$$Z^* = aZ^* + \tau Y^* = aZ^* + \tau(bY^* + \tau Z^*) = aZ^* + \tau(bY^* + aZ^* + \tau Z^*) =$$
$$= \tau(bY^* + aZ^* + \tau Z^*) = \tau(bY^* + aZ^* + Q').$$

Now $Q = Q'$, i.e. $\tau Y^* = \tau Z^*$, is seen as follows:

$$Y^* = bY^* + \tau Z^* = bY^* + \tau(aZ^* + \tau Y^*) = bY^* + aZ^* + \tau(aZ^* + \tau Y^*) = bY^* + aZ^* + \tau Z^*$$

hence $\tau Y^* = \tau(bY^* + aZ^* + \tau Z^*)$. Further, $Z^* = \tau(bY^* + aZ^* + \tau Z^*)$ as derived already; hence $\tau Y^* = \tau Z^*$. □

REMARK 7.10. It is not hard to see that the unique solvability of systems of recursion equations

as in the preceding theorem is preserved when parameters $\underline{Q} = \underline{Q}_1,...,\underline{Q}_n$ are admitted in the RHS's of the equations $X_i = T_i(X)$, i = 1,...,n. In fact, the $\underline{Q}$ are new names for elements Q in $\mathbb{R}^\rho(+,\cdot,u)/\hat{=}_{r\tau}$. To see this, note that one can eliminate the names $\underline{Q}$ in favour of a larger system of equations, as follows: first choose representatives $q_i \in \mathbb{R}^\rho(+,\cdot,u)$ of the Q_i; by Corollary 1.3.13 these may be supposed τ-cycle free. Then write down the canonical LR-expressions denoting these q_i; these are τ-cycle free too. Next, use the definitions of the q_i to extend the original system of equations. This extended system, which is still τ-cycle free, has then a unique solution vector, containing the desired solution vector.

EXAMPLE 7.11. Let $\underline{Q}_1,\underline{Q}_2$ denote $Q_1,Q_2 \in \mathbb{R}^\rho(+,\cdot,u)/\hat{=}_{r\tau}$.
Then $\{X = \tau(aY + \underline{Q}_1),\ Y = \tau(bX + \underline{Q}_2)\}$ has a unique solution in $\mathbb{R}^\rho(+,\cdot,u)/\hat{=}_{r\tau}$.

7.12. KOOMEN'S FAIR ABSTRACTION RULE. A proof rule which is convenient in computations is Koomen's Fair Abstraction Rule (KFAR). It was used by C.J. Koomen of Philips Research in a formula-manipulation system based on CCS [9], and defined explicitly in [1] (see also [5] where it served to give an algebraic verification of a simple version of the Alternating Bit Protocol). In the name KFAR, the adjective 'fair' refers to the bias that rτ-bisimulation has towards a fair execution of τ-paths in the sense that, after finitely many executions of a τ-cycle, an alternative not on the τ-cycle will be chosen if possible. (This bias was already discussed in MILNER [9].) The formal version of the rule KFAR (which is in fact parametrised by $k \geq 1$) is:

$$\mathrm{KFAR}_k \qquad \frac{\forall n \in \mathbb{Z}_k \quad x_n = i_n \cdot x_{n+1} + y_n \qquad (i_n \in I)}{\tau_I(x_n) = \tau \cdot \tau_I(\Sigma_{m \in \mathbb{Z}_k} y_m)}$$

Here the subscripts n,n+1 are elements of $\mathbb{Z}_k = \{0,1,...,k\text{-}1\}$ in which addition is modulo k. The x_n,x_{n+1},y_n,y_m are meta-variables ranging over the process algebra under consideration, in our case: $\mathbb{R}^\rho(+,\cdot,u)/\hat{=}_{r\tau}$. They should not be confused with formal variables X, ... which appear in LR-expressions. The i_n are elements of A; we always require $I \subseteq A$, so i_n cannot be τ. This is essential, as we will show. We conceive the i_n (or more generally, the elements of I) as *internal* steps (but *not*, as τ is, invisible or silent) which can be abstracted to yield τ-steps. We will explain the rule by some examples.

(i) KFAR_1: $\dfrac{x = ix}{\tau_{\{i\}}(x) = \tau}$

(ii) KFAR_1: $\dfrac{x = a + ix}{\tau_{\{i\}}(x) = \tau a}$

(iii) KFAR_3: If $x = iy + p$, $y = jz + q$, $z = kx + r$, then $\tau_{\{i,j,k\}}(x) = \tau_{\{i,j,k\}}(y) = \tau_{\{i,j,k\}}(z) = \tau(\tau_{\{i,j,k\}}(p) + \tau_{\{i,j,k\}}(q) + \tau_{\{i,j,k\}}(r))$.

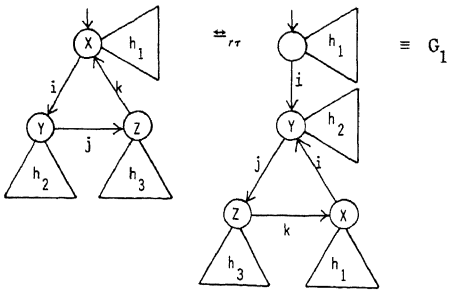
Before proving that KFAR is valid in the model $\mathbb{R}^\rho(+,\cdot,u)/\hat{=}_{r\tau}$, and hence can be added

consistently to the proof system $BPA_{\tau LR}$, we remark that it is essential that the $i_0,i_1,\ldots,i_{k-1}$-"cycle" appearing in the hypothesis of KFAR is *not* a cycle of τ-steps. Indeed, the 'version' of KFAR: "If $\forall n \in Z_k\ x_n = \tau x_{n+1} + y_n$, then $x_n = \tau(\Sigma y_m)$" would simply be false: from $x = a + \tau x$ it does not follow that $x = \tau a$, as we already remarked. This is an important reason for the introduction of τ_I and the distinction of internal steps from τ-steps.

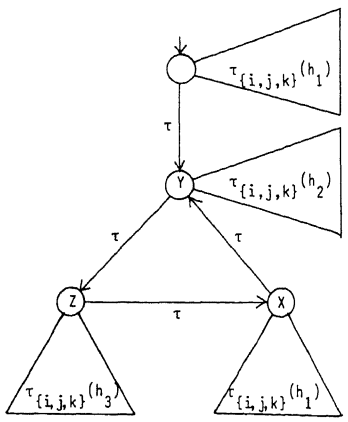
THEOREM 7.12.1. $\mathbb{R}^{\rho}(+,\cdot,u)/\triangleq_{r\tau} \models$ KFAR.

PROOF. We give the proof for $KFAR_3$ and use Example (iii) above. So suppose $g_1 \triangleq_{r\tau} ig_2 + h_1$, $g_2 \triangleq_{r\tau} jg_3 + h_2$, $g_3 \triangleq_{r\tau} kg_1 + h_3$ where $g_1/\triangleq_{r\tau} = x$, $g_2/\triangleq_{r\tau} = y$ and $g_3/\triangleq_{r\tau} = z$. We may suppose by Corollary 1.3.13 that h_1,h_2h_3 are τ-cycle free.

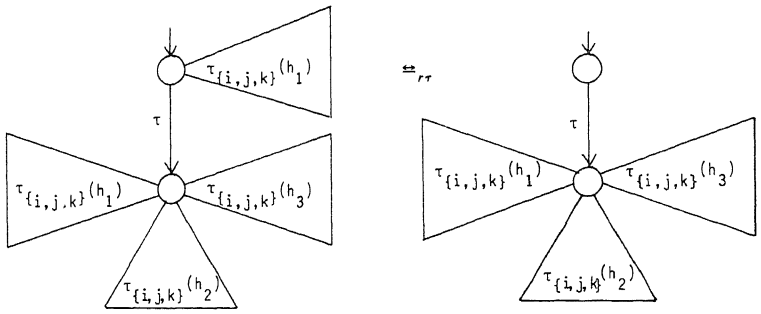
Now by Remark 7.10 the present system of equations has a unique solution (as it is τ-cycle free) modulo $\triangleq_{r\tau}$, namely g_1,g_2,g_3. Hence $g_1 \triangleq_{r\tau}$ the root-unwinding of



Therefore $\tau_{\{i,j,k\}}(g_1) \triangleq_{r\tau} \tau_{\{i,j,k\}}(G_1) =$



which is by (the proof of) Corollary 1.3.13 $r\tau$-bisimilar to



So

$$g_1 \leftrightarrows_{r\tau} \tau(\tau_{\{i,j,k\}}(h_1) + \tau_{\{i,j,k\}}(h_2) + \tau_{\{i,j,k\}}(h_3)),$$

and hence

$$x = \tau(\tau_{\{i,j,k\}}(p) + \tau_{\{i,j,k\}}(q) + \tau_{\{i,j,k\}}(r)),$$

which is the consequence of $KFAR_3$ we wanted. □

REMARK 7.12.2. Note that by the same proof we obtain a slightly stronger version of KFAR (as valid in the domain of regular processes), namely one in which some *but not all* of the i_n ($n \in \mathbb{Z}$) may be τ. Thus we may, e.g., conclude from $x = iy + p$ ($i \in I \subseteq A$), $y = \tau x + q$ that $\tau_I(x) = \tau_I(y) = \tau \cdot \tau_I(p + q)$.

8. $PA_{\tau LR}$: a proof system for regular processes with τ-steps and free merge

It is not hard to extend the proof system $BPA_{\tau LR}$ with axioms for the interleaving (or free merge) operator $\|$. (Here 'free' denotes the absence of communication.) Let $PA_{\tau LR}$ be $BPA_{\tau LR}$ plus the axioms in the following table, where $a \in A_\tau$:

$x \| y = x \mathbin{\lfloor\!\lfloor} y + y \mathbin{\lfloor\!\lfloor} x$	M1
$(ax) \mathbin{\lfloor\!\lfloor} y = a(x \| y)$	M2
$a \mathbin{\lfloor\!\lfloor} y = ay$	M3
$(x + y) \mathbin{\lfloor\!\lfloor} z = x \mathbin{\lfloor\!\lfloor} z + y \mathbin{\lfloor\!\lfloor} z$	M4

Table 10

Warning: LR-expressions are defined as before; $\|$ and the auxiliary operator $\mathbin{\lfloor\!\lfloor}$ (left-merge) may *not* occur in the bodies of LR-expressions, since otherwise we would leave the realm of regular processes. (E.g. the process uniquely defined by $X = a(b \| X)$ is not regular.)

The semantics of $PA_{\tau LR}$ is the obvious extension of the semantics of $BPA_{\tau LR}$, using the

operations $\|$, $\lfloor\!\lfloor$ on graphs which were defined in Section 4.

THEOREM 8.1. *Let* T,S *be closed* $PA_{\tau LR}$*-terms. Then:* $PA_{\tau LR} \vdash T = S \Leftrightarrow [T] = [S]$.

PROOF. Compared to the completeness proof of $BPA_{\tau LR}$ the only extra fact to be proved is that the merge of two expressions T,S can be eliminated in a provable way. Here the main task is to show this for LR-expressions; this we will do now.

Let $\underline{X}_1 \equiv \langle X_1 \mid \{X_i = T_i(\mathbf{X}) \mid i = 1,..,n\}\rangle$ and $\underline{Y}_1 \equiv \langle Y_1 \mid \{Y_1 = S_j(\mathbf{Y}) \mid j = 1,...,m\}\rangle$. Let $D_1,...,D_k$ be the derived subterms of $\underline{X}_1$ (see the definition of 'derived subterm' in the proof of Theorem 5.4.1) and let $E_1,...,E_{k'}$ be the derived subterms of $\underline{Y}_1$. Let $\underline{D}_i$ (i = 1,...,k) be D_i where the formal variables $X_1,...,X_n$ are replaced by $\underline{X}_1,...,\underline{X}_n$; likewise $\underline{E}_j$ (j = 1,...,k') is defined. Further, abbreviate

$$\underline{U}_{ij} \equiv \underline{D}_i \| \underline{E}_j$$
$$\underline{V}_{ij} \equiv \underline{D}_i \lfloor\!\lfloor \underline{E}_j$$
$$\underline{W}_{ij} \equiv \underline{E}_j \lfloor\!\lfloor \underline{D}_i.$$

Now, using the axioms M1-4 and the recursion equations in $\underline{X}_1$, $\underline{Y}_1$ as rewrite rules from left to right, a simple computation shows that the set of expressions $\{\underline{U}_{ij},\underline{V}_{ij},\underline{W}_{ij},\underline{D}_i,\underline{E}_j \mid i = 1,...,k;\ j = 1,...,k'\}$ can be expressed "guardedly in itself". That is:

$$\underline{U}_{ij} = P_{ij}(\underline{\mathbf{U}},\underline{\mathbf{V}},\underline{\mathbf{W}},\underline{\mathbf{D}},\underline{\mathbf{E}})$$
$$\underline{V}_{ij} = Q_{ij}(\underline{\mathbf{U}},\underline{\mathbf{V}},\underline{\mathbf{W}},\underline{\mathbf{D}},\underline{\mathbf{E}})$$
$$\underline{W}_{ij} = R_{ij}(\underline{\mathbf{U}},\underline{\mathbf{V}},\underline{\mathbf{W}},\underline{\mathbf{D}},\underline{\mathbf{E}})$$
$$\underline{D}_i = F_i(\underline{\mathbf{D}},\underline{\mathbf{E}})$$
$$\underline{E}_j = G_j(\underline{\mathbf{D}},\underline{\mathbf{E}})$$

for some guarded terms $P_{ij}(\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{D}',\mathbf{E}')$, $Q_{ij}(\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{D}',\mathbf{E}')$, $R_{ij}(\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{D}',\mathbf{E}')$, $F_i(\mathbf{D}',\mathbf{E}')$, $G_j(\mathbf{D}',\mathbf{E}')$ not containing $\|$, $\lfloor\!\lfloor$. Here the $\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{D}',\mathbf{E}'$ are (strings of) formal variables. Hence

$$\underline{U}_{11} \equiv \underline{X}_1 \| \underline{Y}_1 =$$
$$\langle U_{11} \mid \{U_{ij} = P_{ij}(\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{D}',\mathbf{E}'),\ V_{ij} = Q_{ij}(\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{D}',\mathbf{E}'),\ W_{ij} =$$
$$R_{ij}(\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{D}',\mathbf{E}'),\ D_i' = F_i(\mathbf{D}',\mathbf{E}'),\ E_j' = G_j(\mathbf{D}',\mathbf{E}') \mid i = 1,...,k;\ j = 1,...,k'\}\rangle$$

and likewise for $\underline{X}_1 \lfloor\!\lfloor \underline{Y}_1$ and $\underline{Y}_1 \lfloor\!\lfloor \underline{X}_1$. So the operators $\|$, $\lfloor\!\lfloor$ are eliminated. (If the LR-expressions $\underline{X}_1,\underline{Y}_1$ contain τ-guarded recursion equations, the same procedure is followed after the detour via τ_I as demonstrated in Example 6.2.1.1.) □

EXAMPLE 8.2. Consider $\langle X \mid X = a(X + b)\rangle \| \langle Y \mid Y = cY\rangle$. The derived subterms of $\underline{X}$ are X, X + b and of $\underline{Y}$ only Y. Now

$$\underline{X} \| \underline{Y} = \underline{X} \mathbin{\mathbb{L}} \underline{Y} + \underline{Y} \mathbin{\mathbb{L}} \underline{X} = a(\underline{X} + b) \mathbin{\mathbb{L}} \underline{Y} + c\underline{Y} \mathbin{\mathbb{L}} \underline{X} = a((\underline{X} + b) \| \underline{Y}) + c(\underline{Y} \| \underline{X})$$

$$(\underline{X} + b) \| \underline{Y} = (\underline{X} + b) \mathbin{\mathbb{L}} \underline{Y} + \underline{Y} \mathbin{\mathbb{L}} (\underline{X} + b) = \underline{X} \mathbin{\mathbb{L}} \underline{Y} + b \mathbin{\mathbb{L}} \underline{Y} + \underline{Y} \mathbin{\mathbb{L}} (\underline{X} + b) =$$
$$a(\underline{X} + b) \mathbin{\mathbb{L}} Y + b\underline{Y} + c\underline{Y} \mathbin{\mathbb{L}} (\underline{X} + b) = a((\underline{X} + b) \| \underline{Y}) + b\underline{Y} + c(\underline{Y} \| (\underline{X} + b)).$$

Hence $\underline{X} \| \underline{Y} = \langle U \mid U = aV + cU, V = aV + bY + cV, Y = cY \rangle$.

References

[1] J.C.M. BAETEN, J.A. BERGSTRA, J.W. KLOP (1985). *On the consistency of Koomen's Fair Abstraction Rule*, CWI Report CS-R8511, Amsterdam. To be published in TCS.

[2] J.A. BERGSTRA, J.W. KLOP (1983). *An abstraction mechanism for process algebra*, Report IW 231/83, Centre for Mathematics and Computer Science, Amsterdam 1983.

[3] J.A. BERGSTRA, J.W. KLOP (1984). *Process algebra for synchronous communication*, Information and Control 60 (1/3), 109-137.

[4] J.A. BERGSTRA, J.W. KLOP (1985). *Algebra of communicating processes with abstraction*, Theor. Comp. Sci. 37 (1), 77-121.

[5] J.A. BERGSTRA, J.W. KLOP (1986). *Process Algebra: Specification and Verification in Bisimulation Semantics*, Proceedings of the CWI Symposium Mathematics and Computer Science 1986 (eds. M. Hazewinkel, J.K. Lenstra, L.G.L.T. Meertens), CWI Monographs 4, 61-94, North-Holland, Amsterdam 1986.

[6] S.D. BROOKES (1983). *On the relationship of CCS and CSP*, Proc. 10th ICALP (ed. J. Díaz), Barcelona 1983, Springer LNCS 154, 83-96.

[7] M. HENNESSY, R. MILNER (1985). *Algebraic laws for nondeterminism and concurrency*, JACM 32, 137-161.

[8] C.P.J. KOYMANS, J.C. MULDER (1986). *A modular approach to protocol verification using process algebra*, Logic Group Preprint Series Nr.6, Dept. of Philosophy, State University of Utrecht.

[9] R. MILNER (1980). *A Calculus of Communicating Systems*, Springer LNCS 92.

[10] R. MILNER (1984). *A complete inference system for a class of regular behaviours*, Journal of Computer and Systems Sciences, Vol.28, Nr.3, 439-466.

[11] R. MILNER (1984). *Lectures on a Calculus for Communicating Systems*, Working Material for the Summer School Control Flow and Data Flow, Munich, July 1984.

[12] R. MILNER (1985). *A complete axiomatisation for observational congruence of finite-state behaviours*, manuscript, Univ. of Edinburgh.

[13] C.A.R. HOARE (1984). *Notes on Communicating Sequential Processes*, Working Material for the Summer School Control Flow and Data Flow, Munich, July 1984.

[14] D.M.R. PARK (1981). *Concurrency and automata on infinite sequences*, Proc. 5th GI Conference, Springer LNCS 104.

[15] F.W. VAANDRAGER (1986). *Verification of two communication protocols by means of process algebra*, CWI Report CS-R8608, Amsterdam.