Journal of Universal Computer Science, vol. 6, no. 1 (2000), 39-59
submitted: 20/10/99, accepted: 6/11/99, appeared: 28/1/00 © Springer Pub. Co.

A π-calculus Machine[1]

Gabriel Ciobanu
(Department of Computer Science, A.I. Cuza University,
Iasi, Romania,
Email: gabriel@infoiasi.ro.)

Mihai Rotaru
(Institute of Theoretical Computer Science, Romanian Academy,
Iasi, Romania,
Email: mrotaru@iit.tuiasi.ro.)

Abstract: In this paper we investigate the π-calculus guards, proposing a formalism which use exclusively machine tradition concepts: state, resource, transition. The reduction mechanism is similar to the token-game of Petri nets. We provide a multiset semantics for the π-calculus by using this formalism. Moreover, our machines have a graphical representation which emphasizes their structure. As a consequence, we give a new improved graphical representation for the asynchronous π-calculus.

Key Words: abstract machine, concurrent processes, π-calculus, multiset semantics, nets, graphical representation of processes.

Category: F.3.1.

1 Introduction

There are various formalisms for concurrent distributed computations. Among them, there exist two important classes coming from two different classical approaches of computability. On one hand, there are machines such as Petri nets, communication automata, ... where the central concepts, emerged originally from Turing machines, are state and transition. On the other hand, there are process algebras such as CSP , CCS , π-calculus, ... where the central concepts, emerged originally from λ-calculus, are term and redex. Process algebras have nice theoretical properties determined mainly by compositionality. Machines are more effective and closer to implementation. These two traditional research lines are strongly connected. It is well-known that λ-calculus and Turing machines have the same computation power. In this sense, Turing machines correspond to the λ-calculus. The π-calculus [Mil93] is a process algebra (modulo an equational theory) coming along the tradition started by λ-calculus. It is an extension of CCS. While for CCS there exist corresponding machines (see for instance [Tau89]), there is no machine widely accepted for π-calculus. Two semantics [Eng93, BG95] for the π-calculus use Petri nets as a corresponding machine. However, these approaches did not pay a particular attention to input guarding, an important interaction operator of the π-calculus. In fact, this operator is interpreted by a similar term-like abstract construction. In this way, the corresponding Petri nets lose some features of the automata tradition. We investigate

[1] C. S. Calude and G. Ştefănescu (eds.). *Automata, Logic, and Computability. Special issue dedicated to Professor Sergiu Rudeanu Festschrift.*

deeply the π-calculus guards, proposing a formalism which use exclusively machine tradition concepts: state, resource, transition. The reduction mechanism is similar to the token-game of Petri nets. We provide a semantics of the π-calculus by using this formalism. Moreover, our machines possess a graphical representation which emphasizes their structure. As a consequence, we get a new improved graphical representation for the π-calculus.
Since identical machines correspond to structural congruent π-processes, our machines could be thought as normal forms for the π-processes with respect to the equational theory imposed by structural congruence.
Consequently, concerning interaction, we remark that machines are more effective than processes. In order to identify a redex within a π-process, we sometimes have to transform the π-process according to the structural congruence axioms. We have not such a need for our machines, where all possible reductions are available at any time. Such aspect of effectiveness is suitable and desirable for implementation. Our machines provide also a modular description of the π-processes, well visualized by their graphical representations; this can be quite useful for programming.

The paper is self-contained. Section 2 provides preliminary notions. Section 3 is devoted to R-machines and its graphics. In section 4 we define abstract R-machines. P-machines are introduced in section 5 as an algebra of R-machines. To aid their presentation, we use the graphics of R-machines already introduced in section 3. Section 4 defines abstract R-machines, and we introduce then our P-machines. In section 5 we give a characterization theorem for (quasi-finite) P-machines. Section 6 introduces abstract P-machines. Dynamics of these machines is described in section 7, together with some properties. Section 8 is devoted to the expressive power of our machines; we give a translation from π-calculus to P-machines together with some results related to this translation. The most important result asserts a faithful operational correspondence. This section ends with an example. Section 9 presents conclusions and comparisons with related works.

2 Preliminaries

We use in this paper an extension of natural numbers set $\mathbf{N}_\omega = \mathbf{N} \cup \{\omega\}$, where ω is a special element which is not in $\mathbf{N}$. We denote by $\mathbf{N}^\star$ the set $\mathbf{N} \setminus \{0\}$. The usual order relation $<$ and operations $+$ and $-$ over $\mathbf{N}$ are extended to $\mathbf{N}_\omega$ such that for every $n \in \mathbf{N}$, $\omega + n = n + \omega = \omega + \omega = \omega$, $\omega - n = \omega$, $n - \omega = \omega - \omega = 0$, and $n < \omega$. We denote by $\leq$ the reflexive closure of $<$. The addition operation $+$ over $\mathbf{N}_\omega$ is associative, and thus it can be extended to finite and infinite sums. If such a sum has a term ω, then the whole sum is ω. If an infinite sum of terms from $\mathbf{N}$ diverges, then its value is defined to be ω.

A *multiset* over a set X is a function $r : X \to \mathbf{N}_\omega$. The set X is the support of the multiset r. If $x \in X$, then $r(x)$ is the multiplicity of x in the multiset r. The set of all multisets over X is denoted by $\mathcal{M}(X)$. We shall use two particular members of $\mathcal{M}(X)$: *empty* multiset $\emptyset_X$ given by $\emptyset_X(x) = 0$ for every $x \in X$ and *singleton* multiset $\{\!|x|\!\}_X$, determined by x in X, and given by $\{\!|x|\!\}_X(x) = 1$, and $\{\!|x|\!\}_X(y) = 0$ for every $y \in X$, $y \neq x$. We shall write $\emptyset$ and $\{\!|a|\!\}$ instead of $\emptyset_X$ and $\{\!|a|\!\}_X$ whenever the support X is clear from the context.

We extend some usual set operations to multisets and discuss a few properties of them. Let $r \in \mathcal{M}(X)$ and $r_i \in \mathcal{M}(X_i)$, $i = 1, 2$. Then

$\in_m$ $x \in_m r$ iff $x \in X \ \wedge \ r(x) > 0$.
$=_m$ $r_1 =_m r_2$ iff $x \in_m r_1$ implies $x \in_m r_2 \ \wedge \ r_1(x) = r_2(x)$ and
$x \in_m r_2$ implies $x \in_m r_1 \ \wedge \ r_1(x) = r_2(x)$ for every x.
$\uplus$ $r_1 \uplus r_2 \in \mathcal{M}(X_1 \cup X_2)$ is defined by $(r_1 \uplus r_2)(x) = s_1(x) + s_2(x)$
for $x \in X_1 \cup X_2$ and $s_i \in \mathcal{M}(X_1 \cup X_2)$ such that $s_i =_m r_i, i = 1, 2$.
$-$ $r_1 - r_2 \in \mathcal{M}(X_1)$ is defined by $(r_1 - r_2)(x) = max\{r_1(x) - s(x), 0\}$
for $x \in X_1$ and $s \in \mathcal{M}(X_1 \cup X_2)$ such that $s =_m r_2$.

Remark. (i) If $r, s \in \mathcal{M}(X)$ and $r =_m s$, then $r = s$. (ii) Given a set Y and $r \in \mathcal{M}(X)$, there exists an unique $s \in \mathcal{M}(X \cup Y)$ such that $r =_m s$.

The second assertion in the previous remark ensures that $\uplus$ and $-$ are well defined. To be precise, the multisets s always exist and their uniqueness shows that these operations do not depend on the choice of them. Moreover, the multiset equality $=_m$ can be considered not only as a predicate to compare two multisets, but also as an assignment operator. Thus, if we have a multiset $r \in \mathcal{M}(X)$ and consider $s \in \mathcal{M}(X \cup Y)$, then by writing $s =_m r$ the multiset s is uniquely determined by r.

Given a function $f : X \to Y$, we can extend it to pairs and multisets. We define $f^\times : X \times X \to Y \times Y$ by $f^\times(x_1, x_2) = (f(x_1), f(x_2))$ for any $x_1, x_2 \in X$. If Z is a countable subset of X and $r \in \mathcal{M}(Z)$, then r is either the empty multiset, or a singleton multiset, or it is a multiset union of singleton multisets (multiset union can be extended to countable sets of multisets). Consequently, the image of the multiset r by the function f is the multiset $f(r) \in \mathcal{M}(f(Z))$ defined by $f(\emptyset_Z) = \emptyset_{f(Z)}$, $f(\{\!|z|\!\}_Z) = \{\!|f(z)|\!\}_{f(Z)}$, and $f(\uplus_{k \in K} r_k) = \uplus_{k \in K} f(r_k)$ for some countable set K.

Remark. We consider a function $f : X \to Y$, two countable subsets Z and Z' of X, and two multisets r and r' over Z and Z'. (1) If $r =_m r'$, then $f(r) =_m f(r')$. (2) If $z \in_m r$, then $f(z) \in_m f(r)$. Moreover, if k and l are the multiplicities of z and $f(z)$ in r and $f(r)$, then $k \leq l$.

3 Resource Machines

To define our resource machines, we assume given two uncountable infinite mutually disjoint sets N and T of *nonterminals* and *terminals*. We shall use $x, y, z \ldots$ to range over N and $a, b, c \ldots$ to range over T. Let $R = N \cup T$ be their union, ranged over by $\alpha, \beta, \gamma \ldots$ We shall write $\tilde{x}$ for an enumeration of nonterminals $x_1, x_2, \ldots,$ $\tilde{a}$ for an enumeration of terminals $a_1, a_2, \ldots$, and $\tilde{\alpha}$ for an enumeration of terminals or nonterminals $\alpha_1, \alpha_2, \ldots$ Given such an enumeration, its components are distinct, and the enumerated set is countable. For the sake of simplicity, we keep the same notation to denote the enumerated set.

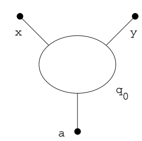
Definition 1. Let I be a subset of $\mathbf{N}$ containing 0. A *R-machine* indexed by I is a structure $A = (q, r, \to)_I$ where

- $q = (q_i)_{i \in I}$ is a family of countable sets $q_i \subset R$ called *states*

– $r = (r_i)_{i \in I}$ is a family of *resources* $r_i \in \mathcal{M}(q_i \times q_i)$ for q_i
– $\rightarrow = (\rightarrow_{ij})_{i,j \in I}$ is a family of *transitions* $\rightarrow_{ij} \subset q_i \times q_j$ from q_i to q_j .

To every R-machine, a directed graph called state graph is associated. Given a R-machine $A = (q, r, \rightarrow)_I$, q_0 is called the *initial* state of A. By $I^\star$ is denoted the set $I \setminus \{0\}$. The associated *state graph* of A is $H_A = (I, E)$, where $E = \{ (i, j) \in I \times I \mid \rightarrow_{ij} \neq \emptyset \}$. A is called *quasi-finite* if every path in H_A has a finite length, $|q_i|, |\rightarrow_{ij}|$ are finite, and $r_i(\alpha, \beta) < \omega$, $\forall \alpha, \beta \in q_i$, $\forall i, j \in I$. To simplify the notation, whenever f is a family of (multi)sets, we use the same f to denote the (multiset) union of the component (multi)sets.

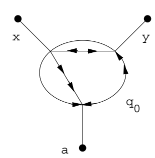
The graphical representation of a R-machine is given by a pseudo-graph. The nodes of the pseudo-graph are labeled injectively by elements of R. They are connected by directed edges and by hyperedges. Finally, the nodes connected by a hyperedge serve as the vertex set for some directed multigraph. To be precise, given a R-machine $A = (q, r, \rightarrow)_I$, the elements of the set q represent the labeled nodes and the elements of the family q represent the hyperedges of the pseudo-graph; e.g. $q_0 = \{x, a, y\}$ is the hyperedge represented by an oval labeled with q_0 and connected through tentacles to the nodes labeled by x, a, and y of the pseudo-graph, as in the following figure:



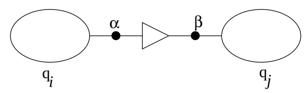
The elements of the family r are directed multigraphs. In particular, r_i is a directed multigraph having the vertex set q_i. This multigraph is represented inside the hyperedge labeled by q_i; e.g. for the above q_0 and

$r_0(x, a) = r_0(a, a) = r_0(y, y) = 2$
$r_0(x, y) = r_0(y, x) = 1$
$r_0(\alpha, \beta) = 0$, otherwise

we have the following graphical representation:



The elements of the set $\rightarrow$ are the directed edges of the pseudo-graph; e.g. $(\alpha, \beta) \in \rightarrow_{ij}$, denoted by $\alpha \rightarrow_{ij} \beta$, is the directed edge from the node labeled by α to the node labeled by β of the pseudo-graph, represented as in the figure:



4 Abstract Resource Machines

We don't want to distinguish between two R-machines which differ by some irrelevant information. We shall identify them by a notion of isomorphism. We shall call *abstract machine* such an isomorphism class.

Definition 2. Let $A = (q, r, \rightarrow)_I$ be a R-machine. For every $i \in I$ we define a set $\mathbf{useful}(q_i)$ of the useful elements in q_i as the smallest set closed to the following rules

- if $(\alpha, \beta) \in_m r_i$, then $\alpha, \beta \in \mathbf{useful}(q_i)$
- if $\alpha \rightarrow_{ij} \beta$, then $\alpha \in \mathbf{useful}(q_i)$
- if $\alpha \rightarrow_{ki} \beta$, then $\beta \in \mathbf{useful}(q_i)$.

Definition 3. Let $A = (q, r, \rightarrow)_I$ be an R-machine. We define the R-machine $\mathbf{useful}(A) = (q', r', \rightarrow')_I$ by

- $q'_i = \mathbf{useful}(q_i), \ i \in I$
- $r'_i = r_i|_{\mathbf{useful}(q_i) \times \mathbf{useful}(q_i)}, \ i \in I$
- $\rightarrow'_{ij} = \rightarrow_{ij}, \ i, j \in I$.

Definition 4. Given A and B two R-machines, and their corresponding R-machines $\mathbf{useful}(A) = (q^A, r^A, \rightarrow^A)_I$ and $\mathbf{useful}(B) = (q^B, r^B, \rightarrow^B)_J$, we say that A and B are isomorphic if there exist two bijective mappings $\phi : q^A \rightarrow q^B$ satisfying $\phi(x) = x$ whenever $x \in N$, and $\sigma : I \rightarrow J$ satisfying $\sigma(0) = 0$ which fulfill the following conditions

- $\phi^\times(r_i^A) = r^B_{\sigma(i)}, \ i \in I$
- $\phi^\times(\rightarrow^A_{ij}) = \rightarrow^B_{\sigma(i)\sigma(j)}, \ i, j \in I$.

We use the notation $(\phi, \sigma) : A \cong B$, or simply $A \cong B$.

We can remark that the conditions within the definition of weak isomorphism refer to resources and transitions. These conditions are enough, they ensure a suitable correspondence for states.

Proposition 5. *Let A and B be two R-machines, and their corresponding R-machines* $\mathbf{useful}(A) = (q^A, r^A, \rightarrow^A)_I$ *and* $\mathbf{useful}(B) = (q^B, r^B, \rightarrow^B)_J$*. If $(\phi, \sigma) : A \cong B$, then $\phi(q_i^A) = q^B_{\sigma(i)}, \ \forall i \in I$.*

The previous defined isomorphism is an equivalence. Modulo this equivalence, we can rename the states of an R-machine, remove elements which are not useful, and rename the elements of T which appear within the R-machine.

Abstract machines have almost the same graphical representation as R-machines. To be precise, the graphical form of an abstract machine corresponding to a R-machine $A = (q, r, \rightarrow)_I$, i.e. its isomorphism class, is obtained starting from the graphical representation of A. For every hyperedge q_i, a tentacle which leads to a vertex bearing a label that is not in $\mathbf{useful}(q_i)$ is removed. The reached vertex is also removed, excepting the case when it is used by another hyperedge. All the labels from T together with the labels of hyperedges are removed.

5 P-Machines

When we describe the graphical representation of an abstract machine, we did not explain how we identify the initial state, after we remove the labels of the hyperedges. In fact, we are working with a subset $\mathbf{PM}^\omega$ of the R-machines where states form a directed rooted tree; the initial state is the root of this tree, and it is easily to be identified.

We define now the endomorphisms over R-machines.

Definition 6. Given a R-machine $A = (q, r, \rightarrow)_I$ and a function $f : R \rightarrow R$, the image of A by f is a R-machine $fA = (q', r', \rightarrow')_I$ defined by

- $q'_i = f(q_i)$, $i \in I$
- $r'_i = f^\times(r_i)$, $i \in I$
- $\rightarrow'_{ij} = f^\times(\rightarrow_{ij})$, $i, j \in I$.

Substitutions are particular endomorphisms.

Definition 7. Let $\tilde{\alpha}$ and $\tilde{\beta}$ be two enumerations from R such that $|\tilde{\alpha}| = |\tilde{\beta}|$. A substitution $\{\tilde{\alpha}/\tilde{\beta}\} : R \rightarrow R$ is given by $\{\tilde{\alpha}/\tilde{\beta}\}(\gamma) =$ if $(\gamma = \beta_k)$, then α_k else γ.

We present some simple properties of substitution in the next lemma.

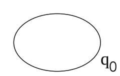
Lemma 8. *Let $A = (q, r, \rightarrow)_I$ be a R-machine. Then*

1. $\{\tilde{\alpha}/\tilde{\beta}\}A = A$, *if* $\tilde{\beta} \cap q = \emptyset$
2. $\{\tilde{\alpha}/\tilde{\alpha}\}A = A$
3. $\{\tilde{\alpha}/\tilde{\beta}\{\tilde{\beta}/\tilde{\alpha'}\}A = \{\tilde{\alpha}/\tilde{\alpha'}\}A$, *if* $\tilde{\beta} \cap q = \emptyset$
4. $\{\tilde{\alpha}/\tilde{\beta}\}\{\tilde{\alpha'}/\tilde{\beta'}\}A = \{\tilde{\alpha'}/\tilde{\beta'}\}\{\tilde{\alpha}/\tilde{\beta}\}A$, *if* $\tilde{\beta} \cap \tilde{\beta'} = \tilde{\alpha} \cap \tilde{\beta'} = \tilde{\alpha'} \cap \tilde{\beta} = \emptyset$.

Definition 9. The set $\mathbf{PM}^\omega$ of P-machines is defined inductively by starting from two simple machines

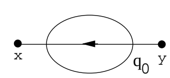
nil : $\mathbf{PM}^\omega$ contains the R-machine $nil \stackrel{def}{=} (q, r, \rightarrow)_I$ where $I^\star = \emptyset$ and

- $q_0 = \emptyset$
- $r_0 = \emptyset$
- $\rightarrow_{00} = \emptyset$.



basic : If $x, y \in N$, then $\mathbf{PM}^\omega$ contains the R-machine $1(x, y) \stackrel{def}{=} (q, r, \rightarrow)_I$ where $I^\star = \emptyset$ and

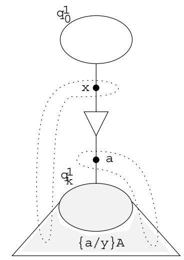
- $q_0 = \{x, y\}$
- $r_0 = \{\!|\ (y, x)\ |\!\}$
- $\rightarrow_{00} = \emptyset$.



and using the following operations:

prefixing : Let $x, y \in N$, $A = (q, r, \rightarrow)_I \in \mathbf{PM}^\omega$, $a \in T \setminus q$, and $k \in \mathbf{N} \setminus I$. If we note $\{a/y\}A = (q', r', \rightarrow')_I$, then $\mathbf{PM}^\omega$ contains the R-machine $(x, y)_{a,k}\, A \stackrel{def}{=} (q^1, r^1, \rightarrow^1)_K$ where $K = I \cup \{k\}$ and

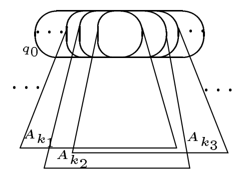
- $q_i^1 = q_i'$, $i \in I^\star$
  $q_k^1 = q_0' \cup \{a\}$
  $q_0^1 = \{x\}$
- $r_i^1 = r_i'$, $i \in I^\star$
  $r_k^1 =_m r_0'$
  $r_0^1 = \emptyset$
- $\rightarrow_{ij}^1 = \rightarrow_{ij}'$, $i, j \in I^\star$
  $\rightarrow_{kj}^1 = \rightarrow_{0j}'$, $j \in I^\star$
  $\rightarrow_{0k}^1 = \{(x, a)\}$
  $\rightarrow_{ij}^1 = \emptyset$, otherwise.



restriction : If $A = (q, r, \rightarrow)_I \in \mathbf{PM}^\omega$, $\tilde{x} \subset N$, and $\tilde{a} \subset T \setminus q$ such that $|\tilde{x}| = |\tilde{a}|$, then $\mathbf{PM}^\omega$ contains the R-machine $(\tilde{x})_{\tilde{a}} A \stackrel{def}{=} \{\tilde{a}/\tilde{x}\}A$.

parallel composition : Let K be a countable set. If $A_k = (q^k, r^k, \rightarrow^k)_{I_k} \in \mathbf{PM}^\omega$, $k \in K$ such that each of the families $(T \cap q^k)_{k \in K}$ and $(I_k^\star)_{k \in K}$ has mutually disjoint sets, then $\mathbf{PM}^\omega$ contains the R-machine $\otimes_{k \in K} A_k \stackrel{def}{=} (q, r, \rightarrow)_I$ where $I = \cup_{k \in K} I_k$ and

- $q_i = q_i^k$, $i \in I_k^\star$, $k \in K$
  $q_0 = \cup_{k \in K} q_0^k$
- $r_i = r_i^k$, $i \in I_k^\star$, $k \in K$
  $r_0 =_m \uplus_{k \in K} r_0^k$
- $\rightarrow_{ij} = \rightarrow_{ij}^k$, $i, j \in I_k$, $k \in K$
  $\rightarrow_{ij} = \emptyset$, otherwise.



In $\mathbf{PM}^\omega$ we consider the subset $\mathbf{PM}$ of the quasi-finite R-machines.

Lemma 10. *Given a R-machine A, if $A \in \mathbf{PM}^\omega$, then* **useful**$(A) = A$.

Lemma 11. *Let $A, B \in \mathbf{PM}^\omega$ such that $A \cong B$. If $A \in \mathbf{PM}$, then $B \in \mathbf{PM}$.*

Lemma 12. *If all constructions involved are valid, then*

1. $\{\tilde{\alpha}/\tilde{\beta}\}nil = nil$
2. $\{\tilde{\alpha}/\tilde{\beta}\}1(x, y) = 1(\{\tilde{\alpha}/\tilde{\beta}\}x, \{\tilde{\alpha}/\tilde{\beta}\}y)$
3. $\{\tilde{\alpha}/\tilde{\beta}\}(x, y)_{a,k} A = (\{\tilde{\alpha}/\tilde{\beta}\}x, y)_{a,k} \{\tilde{\alpha}/\tilde{\beta}\}A$, *if* $y, a \notin \tilde{\alpha} \cup \tilde{\beta}$
4. $\{\tilde{\alpha}/\tilde{\beta}\}(\tilde{x})_{\tilde{a}} A = (\tilde{x})_{\tilde{a}} \{\tilde{\alpha}/\tilde{\beta}\}A$, *if* $\tilde{\alpha} \cap \tilde{x} = \tilde{\beta} \cap \tilde{x} = \tilde{\beta} \cap \tilde{a} = \emptyset$
5. $\{\tilde{\alpha}/\tilde{\beta}\} \otimes_{k \in K} A_k = \otimes_{k \in K} \{\tilde{\alpha}/\tilde{\beta}\} A_k$.

Lemma 13. *Given a R-machine $A = (q, r, \rightarrow)$, whenever the constructions involved are valid, we have*

1. $(x, y)_{a,k} A = (x, y')_{a,k} \{y'/y\}A$, *if* $y' \in N \setminus q$
2. $(\tilde{x})_{\tilde{a}} A = (\tilde{x'})_{\tilde{a}} \{\tilde{x'}/\tilde{x}\}A$, *if* $\tilde{x'} \subset N \setminus (q \cup \tilde{x})$.

Proposition 14. *If $A \in \mathbf{PM}^\omega$, then $\{\tilde{x}/\tilde{z}\}A \in \mathbf{PM}^\omega$ for any two enumerations $\tilde{x}, \tilde{z}$ from N such that $|\tilde{x}| = |\tilde{z}|$.*

The following result gives a characterization for **PM**. The proof of the theorem is included into an appendix.

Theorem 15. *Let $A = (q, r, \rightarrow)_I$ be a quasi-finite R-machine. $A \in \mathbf{PM}^\omega$ if and only if for any $i, j \in I$, A satisfies the following conditions*

1. $\mid \rightarrow_{ij} \mid \leq 1$
2. *its state graph H_A is a directed rooted tree having* 0 *as root*
3. $\mathbf{useful}(q_i) = q_i$
4. *if $\alpha \rightarrow_{ij} \beta$, then*
   (a) $\beta \in T$ *and*
   (b) *whenever $\beta \in q_k$ $(k \neq j)$, then there exists a jk-path in H_A.*

6 Abstract P-Machines

We denote by $\mathbf{APM}^\omega$ the set of all abstract P-machines, and by **APM** the set of all abstract quasi-finite P-machines. For every abstract P-machine $\mathcal{A}$, we associate the set $\lambda(\mathcal{A})$ of the nonterminals, and the number $\kappa(\mathcal{A})$ of the shared terminals of any P-machine $A \in \mathcal{A}$.

Definition 16. Function $\lambda : \mathbf{APM}^\omega \rightarrow 2^N$ is defined by $\lambda(\mathcal{A}) = q^A \cap N$, where $A \in \mathbf{PM}^\omega \cap \mathcal{A}$.

Definition 17. If $\mathcal{A} \in \mathbf{APM}^\omega$ and $A \in \mathbf{PM}^\omega \cap \mathcal{A}$, then we define

$$\kappa(\mathcal{A}) = \mid \bigcup_{\substack{i,j \in I_A \\ i \neq j}} q_i^A \cap q_j^A \cap T \mid$$

Definition 18. We define the following operations upon $\mathbf{APM}^\omega$:

1. $(x, y)\mathcal{A} \stackrel{def}{=} [\,(x, y)_{a,k} A\,]_{\cong}, \;\; A \in \mathcal{A}$;
2. $(\tilde{x})\mathcal{A} \stackrel{def}{=} [\,(\tilde{x})_{\tilde{a}} A\,]_{\cong}, \;\; A \in \mathcal{A}$;
3. $\otimes_{k \in K} \mathcal{A}_k \stackrel{def}{=} [\,\otimes_{k \in K} A_k\,]_{\cong}, \;\; A_k \in \mathcal{A}_k, \; k \in K$.

Lemma 19.

1. *If $A \cong B$, then $(x, y)_{a,k} A \cong (x, y)_{b,l} B$.*
2. *If $A \cong B$, then $(\tilde{x})_{\tilde{a}} A \cong (\tilde{x})_{\tilde{b}} B$.*
3. *If $A_k \cong B_k$ for every $k \in K$, then $\otimes_{k \in K} A_k \cong \otimes_{k \in K} B_k$.*

This lemma is not enough to prove that the previous introduced operations are well-defined. Next proposition comes to show that this is true.

Proposition 20. *Let $x, y \in N$ and let $\tilde{x}$ be an enumeration in N. If $\mathcal{A}, \mathcal{A}_k \in$ $\mathbf{APM}^\omega$ where k belongs to the (countable) set K, then*

1. $(x, y)\mathcal{A} \in \mathbf{APM}^\omega$ *;*
2. $(\tilde{x})\mathcal{A} \in \mathbf{APM}^\omega$ *;*
3. $\otimes_{k \in K} \mathcal{A}_k \in \mathbf{APM}^\omega$ *.*

The following lemma is useful in the proof of the previous proposition, but it suggests also that the requirement that some sets are uncountable is not essential.

Lemma 21. *Let $B \in \mathbf{PM}^\omega$ and let $S \subset T$, $L \subset \mathbf{N}^\star$ be two infinite countable sets. There exits $A \in \mathbf{PM}^\omega$ such that $q^A \cap T \subset S$, $I_A^\star \subset L$, and $A \cong B$.*

Definition 22. If $x_1, x_2 \in N$, then the function $\{x_1/x_2\} : \mathbf{APM}^\omega \to \mathbf{APM}^\omega$ is defined by $\{x_1/x_2\}\mathcal{A} = [\,\{x_1/x_2\}A\,]_\cong$, where $A \in \mathbf{PM}^\omega \cap \mathcal{A}$.

Lemma 23. *Let $A, B \in \mathbf{PM}^\omega$. If $A \cong B$, then $\{x_1/x_2\}A \cong \{x_1/x_2\}B$.*

Some results given for P-machines are valid for abstract P-machines too. We use the notations : $\mathbf{nil} = [\,nil\,]_\cong$ and $\mathbf{1}(x, y) = [\,1(x, y)\,]_\cong$.

Lemma 24.

1. $\{x_1/x_2\}\mathbf{nil} = \mathbf{nil}$;
2. $\{x_1/x_2\}\mathbf{1}(x, y) = \mathbf{1}(\{x_1/x_2\}x, \{x_1/x_2\}y)$;
3. $\{x_1/x_2\}(x, y)\mathcal{A} = (\{x_1/x_2\}x, y)\{x_1/x_2\}\mathcal{A}$, *if* $y \notin \{x_1, x_2\}$;
4. $\{x_1/x_2\}(x)\mathcal{A} = (x)\{x_1/x_2\}\mathcal{A}$, *if* $x \notin \{x_1, x_2\}$;
5. $\{x_1/x_2\} \otimes_{k \in K} \mathcal{A}_k = \otimes_{k \in K} \{x_1/x_2\}\mathcal{A}_k$.

Lemma 25.

1. $(x, y)\mathcal{A} = (x, y')\{y'/y\}\mathcal{A}$, *if* $y' \notin \lambda(\mathcal{A})$;
2. $(x)\mathcal{A} = (x')\{x'/x\}\mathcal{A}$, *if* $x' \notin \lambda(\mathcal{A})$.

Lemma 26. *If $\mathcal{A} \in \mathbf{APM}$ and $A \in \mathbf{PM}^\omega \cap \mathcal{A}$, then $A \in \mathbf{PM}$.*

7 Dynamics

Until now we talked about the static part of our machines. We proceed with our approach, and we present the dynamics. The static part of our machines uses concepts as those of state, transition or resource derived from the formalisms that belong to the tradition of machines. The dynamics of our machines also uses a *token-game* mechanism similar to that of Petri nets. However, our machines have a flexible structure.

First we focus our attention on the helpful relation $\mapsto \subset \mathbf{PM} \times \mathbf{APM}$.

Definition 27. Let $A = (q^A, r^A, \to^A)_{I_A} \in \mathbf{PM}$. If we have $(\alpha_1, \gamma) \in_m r_0$ and $\gamma \to_{0k} \alpha_2$, then $A \mapsto [\,\{\alpha_1/\alpha_2\}B\,]_\cong$ where $B = (q^B, r^B, \to^B)_{I_B}$ is given by $I_B = I_A \setminus \{k\}$ and

- $q_j^B = q_j^A$, $j \in I_B^\star$,
  $q_0^B = q_0^A \cup q_k^A$;
- $r_j^B = r_j^A$, $j \in I_B^\star$,
  $r_0^B =_m (r_0^A - \{\!|(\alpha_1, \beta)|\!\}) \ \uplus\ r_k^A$;
- $\to_{ij}^B = \to_{ij}^A$, $i, j \in I_B^\star$,
  $\to_{0j}^B = \to_{0j}^A \cup \to_{kj}^A$, $j \in I_B^\star$,
  $\to_{ij}^B = \emptyset$, otherwise.

The multiset equality $r_0^B =_m (r_0^A - \{\!|(\alpha_1,\beta)|\!\}) \uplus r_k^A$ is the core part of the reduction mechanism. We can remark some similarity with the token-game of Petri nets. But what makes the difference is that in our machines the tokens are consumed. Resource $(\alpha_1,\beta) \in_m r_0$ – which corresponds to a token in Petri nets – is *consumed*, and not sent. Moreover, our machines have a flexible structure, and not a fixed one as that of Petri nets. After the resource of the initial state is consumed, this state, q_0, is *enriched* (by fusion) with the resources of the state q_k determined by the transition $\beta \rightarrow_{0k} \alpha_2$. We call $\langle \alpha_1, \gamma, \alpha_2; k\rangle$ a *reduction tuple* of A. We shall use $\rho, \pi \dots$ to range over reduction tuples. If ρ is a reduction tuple of A, then we write $A \mapsto^{\rho} \mathcal{A}$ for the reduction of A by ρ.

Proposition 28. $\mapsto$ *is well-defined.*

The proof of this proposition is mainly based on theorem 15. We extend now $\mapsto$ to a relation $\Rightarrow$ upon **APM**.

Definition 29. We define $\mathcal{A} \Rightarrow \mathcal{B}$ if $A \mapsto \mathcal{B}$ for some $A \in \mathcal{A}$.

The next lemmas can be proved by analyzing the corresponding reduction tuples.

Lemma 30. *Let* $A, B \in \mathbf{PM}$. *If* $A \cong B$ *and* $A \mapsto \mathcal{A}$, *then* $B \mapsto \mathcal{A}$.

Lemma 31.

1. *If* $\mathcal{A}_1 \Rightarrow \mathcal{A}_2$, *then* $(\tilde{x})\mathcal{A}_1 \Rightarrow (\tilde{x})\mathcal{A}_2$.
2. *If* $(\tilde{x})\mathcal{A}_1 \Rightarrow \mathcal{A}$, *then there exists* $\mathcal{A}_2$ *such that* $\mathcal{A}_1 \Rightarrow \mathcal{A}_2$ *and* $\mathcal{A} = (\tilde{x})\mathcal{A}_2$.

Lemma 32.

1. $\mathbf{1}(x,z) \otimes (x,y)\mathcal{A} \Rightarrow \{z/y\}\mathcal{A}$.
2. *If* $\mathcal{A}_1 \Rightarrow \mathcal{A}_2$, *then* $\mathcal{A}_1 \otimes \mathcal{A} \Rightarrow \mathcal{A}_2 \otimes \mathcal{A}$ *and* $\mathcal{A} \otimes \mathcal{A}_1 \Rightarrow \mathcal{A} \otimes \mathcal{A}_2$.
3. *If* $\mathcal{A} \otimes \mathcal{B} \Rightarrow \mathcal{C}$, *then one of the following possibilities holds*

1a)	$\mathcal{A} \Rightarrow \mathcal{A}'$ $\mathcal{C} = \mathcal{A}' \otimes \mathcal{B}$	1b)	$\mathcal{B} \Rightarrow \mathcal{B}'$ $\mathcal{C} = \mathcal{A} \otimes \mathcal{B}'$
2a)	$\mathcal{A} = \mathcal{A}_1 \otimes \mathbf{1}(x,z)$ $\mathcal{B} = (\tilde{u})((x,v)\mathcal{B}_1 \otimes \mathcal{B}_2)$ $\mathcal{C} = \mathcal{A}_1 \otimes (\tilde{u})(\{z/v\}\mathcal{B}_1 \otimes \mathcal{B}_2)$	2b)	$\mathcal{A} = (\tilde{u})((x,v)\mathcal{A}_1 \otimes \mathcal{A}_2)$ $\mathcal{B} = \mathcal{B}_1 \otimes \mathbf{1}(x,z)$ $\mathcal{C} = \mathcal{B}_1 \otimes (\tilde{u})(\{z/v\}\mathcal{A}_1 \otimes \mathcal{A}_2)$
3a)	$\mathcal{A} = (w)(\mathcal{A}_1 \otimes \mathbf{1}(x,w))$ $\mathcal{B} = (\tilde{u})((x,v)\mathcal{B}_1 \otimes \mathcal{B}_2)$ $\mathcal{C} = (w)(\mathcal{A}_1 \otimes (\tilde{u})(\{z/v\}\mathcal{B}_1 \otimes \mathcal{B}_2))$ where $\tilde{u}vw$ is an sequence in $N \setminus \lambda(\mathcal{A} \otimes \mathcal{B})$ such that $\lvert\tilde{u}\rvert \leq \kappa(\mathcal{B})$	3b)	$\mathcal{A} = (\tilde{u})((x,v)\mathcal{A}_1 \otimes \mathcal{A}_2)$ $\mathcal{B} = (w)(\mathcal{B}_1 \otimes \mathbf{1}(x,w))$ $\mathcal{C} = (w)(\mathcal{B}_1 \otimes (\tilde{u})(\{z/v\}\mathcal{A}_1 \otimes \mathcal{A}_2))$ where $\tilde{u}vw$ is a sequence in $N \setminus \lambda(\mathcal{A} \otimes \mathcal{B})$ such that $\lvert\tilde{u}\rvert \leq \kappa(\mathcal{A})$

8 From π-calculus to P-machines

To show the expressiveness of our machines we compare them with π-calculus. This is a calculus for interaction which has the full power of Turing machines. We show that π-calculus can be simulated with our machines. Consequently, everything the Turing machines can do, our machines can also do. A natural question is the following : can our machines do more ? The well-known Turing-Church thesis seems to give us a negative answer. Nevertheless, in a recent paper

[Weg98] it is stated that Turing machines can not simulate interaction. In this way, a machine with a reduction mechanism based on interaction – and this is our case – can do something that a Turing machine can not do. This is not an argument to infirm the recalled thesis, but just one more step towards a better understanding of the power of interaction.

8.1 π-calculus

First we introduce the formal π-calculus framework. We use the asynchronous version of the π-calculus [HT91, Bou92, ACS96], i.e. a fragment of the π-calculus with a particular form of replication, where there is no output prefixing and nondeterministic sum. Therefore we don't use the output guards $\overline{x}\langle z\rangle.P$, but only the output messages $\overline{x}\langle z\rangle$; an output message denotes the emission of a name z along a channel x. It is well-known that asynchronous π-calculus can simulate full π-calculus.

Let $\mathcal{X} \subset N$ be a infinite countable set of *names*. The elements of $\mathcal{X}$ are denoted by $x, y, z \ldots$ The terms of this formalism are called processes. The set of processes is denoted by $\mathcal{P}$, and processes are denoted by $P, Q, R \ldots$.

Definition 33. The *processes* are defined over the set $\mathcal{X}$ of names by the following grammar

$$P \ ::= \ 0 \quad | \quad \overline{x}\langle z\rangle \quad | \quad x(y).P \quad | \quad !x(y).P \quad | \quad \nu x P \quad | \quad P \mid Q$$

0 is the empty process. An input guard $x(y).P$ denotes the reception of an arbitrary name z along the channel x, and afterwards behaving as $\{z/y\}P$. A replicated input guard $!x(y).P$ denotes a process that allows to generate arbitrary instances of the form $\{z/y\}P$ in parallel, by repeatedly receiving names z along channel the x. The informal meaning of the restriction $\nu x P$ is that x is local in P. $P \mid Q$ represents the parallel composition of P and Q.
In $x(y).P$, the name y binds free occurrences of y in P, and in $\nu x P$, the name x binds free occurrences of x in P. We denote by $fn(P)$ the set of the names with free occurrences in P, and by $=_\alpha$ the standard α-conversion.

Over the set of processes it is defined a structural congruence relation; this relation provides a static semantics of some formal constructions.

Definition 34. The relation $\equiv \subset \mathcal{P} \times \mathcal{P}$ is called *structural congruence*, and it is defined as the smallest congruence over processes which satisfies

- $P \equiv Q$ if $P =_\alpha Q$
- $P \mid 0 \equiv P$, $P \mid Q \equiv Q \mid P$, $(P \mid Q) \mid R \equiv P \mid (Q \mid R)$, $!P \equiv P \mid !P$
- $\nu x 0 \equiv 0$, $\nu x \nu y P \equiv \nu y \nu x P$, $\nu x(P \mid Q) \equiv \nu x P \mid Q$ if $x \notin fn(Q)$.

The structural congruence deals with the aspects related to the structure of the processes. Dynamics is defined by a reduction relation.

Definition 35. The *reduction* relation over processes is defined as the smallest relation $\rightarrow \subset \mathcal{P} \times \mathcal{P}$ satisfying the following rules

(com) $\overline{x}\langle z\rangle \mid x(y).P \rightarrow \{z/y\}P$
(par) $P \rightarrow Q$ implies $P \mid R \rightarrow Q \mid R$
(res) $P \rightarrow Q$ implies $(\nu x)P \rightarrow (\nu x)Q$
(str) $P \equiv P',\ P' \rightarrow Q'$, and $Q' \equiv Q$ implies $P \rightarrow Q$.

8.2 The encoding of π-calculus by P-machines

Definition 36. The function of interpretation $\mathcal{I} : \mathcal{P} \to \mathbf{APM}$ is defined by

- $\mathcal{I}(0) = \mathbf{nil}$
- $\mathcal{I}(\overline{x}\langle z \rangle) = \mathbf{1}(x, z)$
- $\mathcal{I}(x(y).P) = (x, y)\mathcal{I}(P)$
- $\mathcal{I}(!P) = \otimes_{n \geq 0} \mathcal{I}(P)$
- $\mathcal{I}(\nu x P) = (x)\, \mathcal{I}(P)$
- $\mathcal{I}(P_1 \mid P_2) = \mathcal{I}(P_1) \otimes \mathcal{I}(P_2)$.

Proposition 37. *$\mathcal{I}$ is well-defined.*

Proof. The proof is mainly based on proposition 3 and lemma 11.

Proposition 38. $fn(P) = \lambda(\,\mathcal{I}(P)\,)$.

Proof. A simple induction on P.

Proposition 39. $\mathcal{I}(\,\{x_1/x_2\}P\,) = \{x_1/x_2\}\mathcal{I}(P)$.

Proof. By induction on P, by using lemmas 9, 10, and the previous proposition.

Proposition 40. *If $P_1 \equiv P_2$, then $\mathcal{I}(P_1) = \mathcal{I}(P_2)$.*

Proof. By induction on the derivation of $\equiv$. It is based mainly on lemma 10, and the previous proposition.

Lemma 41. $|\lambda(\,\mathcal{I}(P)\,)|$ *and* $\kappa(\,\mathcal{I}(P)\,)$ *are finite.*

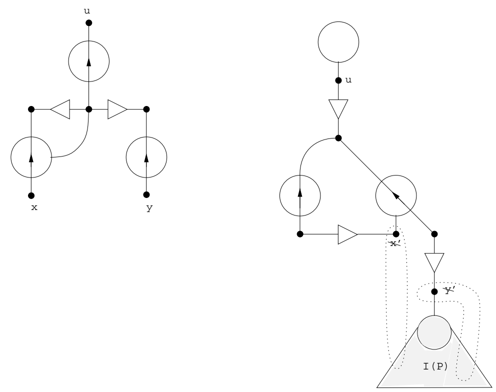
Theorem 42.

1. *If $P_1 \to P_2$, then $\mathcal{I}(P_1) \Rightarrow \mathcal{I}(P_2)$.*
2. *If $\mathcal{I}(P_1) \Rightarrow \mathcal{A}$, then there exists P_2 such that $P_1 \to P_2$ and $\mathcal{A} = \mathcal{I}(P_2)$.*

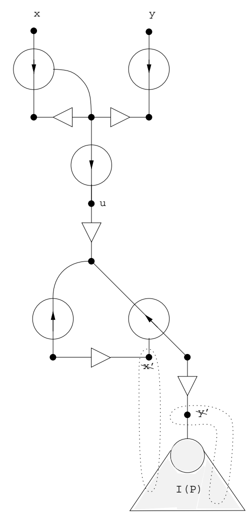
Proof. Using lemmas 31,32, and the previous propositions. The first part follows by induction on the derivation of $\to$. The second, by induction on P_1. Note that for the second part, previous lemma plays an important role.

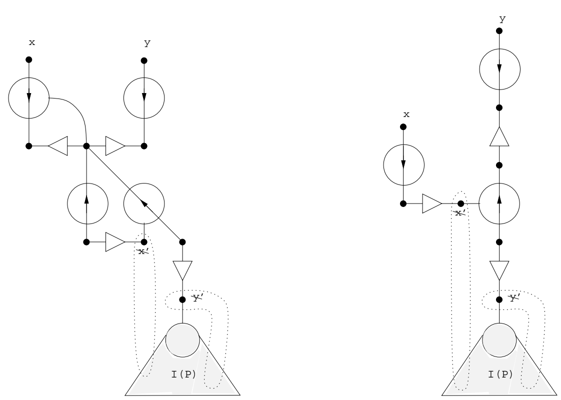
8.3 An example

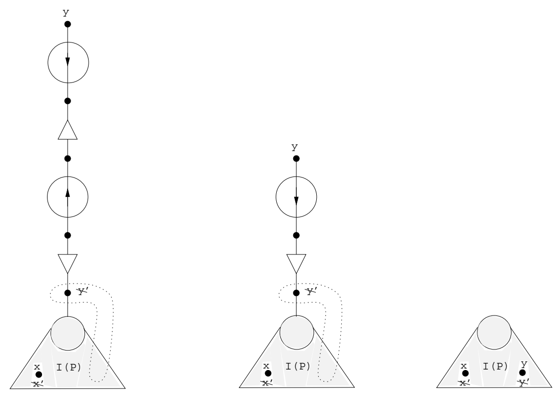
In this section we use our machines to model a problem of synchronization: a process P is waiting for a channel x, and then for a channel y along a public channel u to stock them in x', and respectively y'. We deliberately choose this problem that Milner also uses to explain his graphical form of the π-calculus given by the π-nets [Mil94]. In this way, we could compare Milner's graphical form of the π-calculus with ours (obtained via the interpretation). A solution proposed by Honda and Tokoro is to consider two π-processes, a sender $S = \nu w(\overline{u}\langle w \rangle \mid w(v_1).(\overline{v}_1\langle x \rangle \mid w(v_2).\overline{v}_2\langle y \rangle))$, and a receiver $R = u(w).\, \nu v_1(\overline{w}\langle v_1 \rangle \mid v_1(x').\, \nu v_2(\overline{w}\langle v_2 \rangle \mid v_2(y').P))$.
We translate the processes S (figure a), R (figure b), and $S \mid R$ into machines by using their interpretation. Afterwards, we show how our machines dynamically work by analyzing the reduction sequence of the machine which corresponds to the process $S \mid R$.



a) Sender b) Receiver







9 Conclusion and Related Works

We introduce in this paper a new formalism for concurrency, namely a π-calculus machine with joined resources. We translate the π-processes into these machines, giving at the same time a refined multiset semantics for the π-calculus. This approach asked a deep investigation of the input guards. We give a graphical description for our π-calculus machines. Taking into account that we give a semantics of the π-calculus by using these machines, we get a new graphical representation of the π-calculus. This new graphical representation improves the faithful π-nets we already presented in [CR98].

The description of concurrent processes has been approached from different points of view. In particular, machine-like formalisms and various process algebras. These different descriptions can be seen as expressing complementary views of concurrent processes. Machines are used to describe processes with all details of their operational behaviour. Process algebras are considered as abstract specifications or concurrent programming languages, and they are based on compositionality. By our machines, we describe the details of the operational behaviour, but preserving also the compositionality and the expressive power of the π-calculus. Using the semantics given by Engelfriet or by Busi and Gorrieri, someone cannot reach the details of the operational behaviour, and the corresponding machines for the π-processes are not so effective as ours. Our formalism is in fact close to that described by Engelfriet in [Eng93]; our semantics conservatively extends Engelfriet's approach.

On the other hand, our translation from π-calculus to P-machine is somehow similar to the translation from action calculi to molecular forms presented in [Mil96].

References

[ACS96] Amadio, R., Castellani, I., Sangiorgi, D.: "On Bisimulation for the Asynchronous π-calculus"; TCS (Theoretical Computer Science) 195 (1998), 291-324.

[Bou92] Boudol, G.: "Asynchrony and the π-calculus"; INRIA Technical Report 1702, Sophia-Antipolis, France (1992).

[BG95] Busi, N., Gorrieri, R.: "A Petri Net Semantics for π-calculus"; Proc. CONCUR'95, Springer-Verlag (1995).

[CR98] Ciobanu, G., Rotaru, M.: "Faithful π-nets. A Graphical Representation of the Asynchronous π-calculus."; ENTCS (Electronic Notes of Theoretical Computer Science), 18 (1998), 22 p, http://www.elsevier.nl/locate/entcs/volume18.html.

[Eng93] Engelfriet, J.: "A Multiset Semantics for the π-calculus with Replication"; TCS (Theoretical Computer Science) 153 (1996), 65-94.

[HT91] Honda, K., Tokoro, M.: "An Object Calculus for Asynchronous Communication"; Proc. ECOOP'91, LNCS 512, Springer-Verlag (1991), 133-147.

[Mil93] Milner, R.: "The Polyadic π-calculus: a Tutorial"; in Bauer, F.L., Schwichtenberg, W.H. (Eds.): Logic and Algebra of Specification, Springer-Verlag (1993), 203-246.

[Mil94] Milner, R.: "π-nets: a Graphical Form of π-calculus"; Proc. ESOP'94, LNCS 788, Springer-Verlag (1994), 26-42.

[Mil96] Milner, R.: "Calculi for Interaction"; Acta Informatica, 33, 8 (1996), 707-737.

[Old91] Olderog, E.-R.: "Nets, Terms and Formulas"; Cambridge University Press (1991).

[Tau89] Taubner, D.: "Finite Representation of CCS and TCSP Programs by Automata and Petri Nets"; LNCS 369, Springer-Verlag (1989).

[Weg98] Wegner, P.: "Interactive Foundations of Computing"; TCS (Theoretical Computer Science), 192 (1998), 315-351.

A Proof of theorem 15

In the proof of the theorem we use the following lemma:

Lemma 43. *If $A = (q, r, \rightarrow)_I$ is a quasi-finite R-machine, where $I^\star = \emptyset$, and the following requirements*

1. $\rightarrow_{00} = \emptyset$
2. $\mathbf{useful}(q_0) = q_0$
3. $q_0 \subset N$

hold, then $A \in \mathbf{PM}^\omega$.

Proof. By induction on $|q_0|$.

If $|q_0| = 0$, then $q_0 = \emptyset$ and $r_0 = \emptyset$; thus A is $nil \in \mathbf{PM}^\omega$.

If $|q_0| > 0$, then let $x \in q_0$. By keeping in mind the graphical form of A, we consider the labels of those vertices linked by (multi)edges to the vertex labeled by x, i.e. the set

$$\Delta_x = \{y \in q_0 \mid (x, y) \in_m r_0 \ \vee \ (y, x) \in_m r_0\}$$

We also consider the labels of the vertices linked only by (multi)edges to the vertex labeled by x, i.e. the set

$$\nabla_x = \{y \in \Delta_x \mid \forall z \in q_0 \setminus \{x\},\ r_0(y, z) = r_0(z, y) = 0\}.$$

We now build the following two R-machines:

$A_x = (q^x, r^x \rightarrow)_I$, where

- $q_0^x = \Delta_x \cup \{x\}$;
- $r_0^x(x_1, x_2) = r_0(x_1, x_2),\ \ (x_1, x_2) \in (q_0^x \times \{x\}) \cup (\{x\} \times q_0^x)$,
  $r_0^x(x_1, x_2) = 0$, otherwise

and

$\overline{A_x} = (\overline{q^x}, \overline{r^x}, \rightarrow)_I$, where

- $\overline{q_0^x} = q_0 \setminus (\nabla_x \cup \{x\})$;
- $\overline{r_0^x}(x_1, x_2) = r_0(x_1, x_2),\ \ x_1, x_2 \in \overline{q_0^x}$.

First, we show that $A_x \in \mathbf{PM}^\omega$. To this end we take the following machines of $\mathbf{PM}^\omega$ (parallel compositions of 0 machines is considered nil) :

$$B_x = \overbrace{1(x, x) \otimes \ldots \otimes 1(x, x)}^{r_0(x,x) \text{ times}};$$
$$B_y = \underbrace{1(x, y) \otimes \ldots \otimes 1(x, y)}_{r_0(x,y) \text{ times}} \otimes \underbrace{1(y, x) \otimes \ldots \otimes 1(y, x)}_{r_0(y,x) \text{ times}}, \quad \text{where } y \in \Delta_x \setminus \{x\}.$$

It is easy to show the following decomposition $A_x = \otimes_{z \in q_0^x} B_z$. It follows that $A_x \in \mathbf{PM}^\omega$.

Second, we prove by induction that $\overline{A_x} \in \mathbf{PM}^\omega$. To this end, it is easy to see that $|\overline{q_0^x}| < |q_0|$ and that $\overline{A_x}$ satisfies the conditions 1 and 3. It remains to show that $\overline{A_x}$ satisfies the condition 2. If we suppose it does not, then $\exists x_0 \in \overline{q_0^x}$ such that $\forall y \in \overline{q_0^x}$ we have $\overline{r_0^x}(x_0, y) = \overline{r_0^x}(y, x_0) = 0$. By definition of $\overline{r_0^x}$, it follows that $r_0(x_0, y) = r_0(y, x_0) = 0$, $\forall y \in \overline{q_0^x}$. Since $x_0 \in$ **useful**(q_0), $\exists y_0 \in q_0$ such that $r_0(x_0, y_0) > 0 \vee r_0(y_0, x_0) > 0$. Thus $y_0 \in \nabla_x \cup \{x\}$. If $y_0 \in \nabla_x$, then $\forall z \in q_0 \setminus \{x\}$, $r_0(z, y_0) = r_0(y_0, z) = 0$; considering $z = x_0$ we get a contradiction. Consequently, $y_0 = x$. This implies $x_0 \in \Delta_x$. We show moreover that $x_0 \in \nabla_x$, and in this way we also get a contradiction. To this end, let $y \in q_0 \setminus \{x\}$. If $y \notin \nabla_x$, then $y \in \overline{q_0^x}$ and so $r_0(x_0, y) = r_0(y, x_0) = 0$. If $y \in \nabla_x$, then $\forall z \in q_0 \setminus \{x\}$, $r_0(z, y) = r_0(y, z) = 0$; considering $z = x_0$ we obtain $x_0 \in \nabla_x$, contradiction.

We prove that $A = A_x \otimes \overline{A_x}$. Parallel composition is valid, and let $A_x \otimes \overline{A_x} = (q', r', \rightarrow)_I$. Immediately, $q_0 = q'_0$. It is enough to verify $r_0 =_m r'_0$. To this end we take the multisets $k_1, k_2 \in \mathcal{M}(q_0 \times q_0)$ uniquely determined by $k_1 =_m r_0^x$ and $k_2 =_m \overline{r_0^x}$. To show the above multiset equality it suffices to prove that

$$\forall x_1, x_2 \in q_0, \qquad r_0(x_1, x_2) = k_1(x_1, x_2) + k_2(x_1, x_2) \qquad (\star)$$

Let $x_1, x_2 \in q_0$. We proceed by cases.

a) If $x_1 = x_2 = x$, then $x_1, x_2 \notin \overline{q_0^x}$, and so $k_2(x_1, x_2) = 0$. On the other hand, since $x_1, x_2 \in q_0^x$, we have $k_1(x_1, x_2) = r_0^x(x, x) = r_0(x_1, x_2)$, thus $(\star)$.
b) If $x_1 = x$ and $x_2 \neq x$, then $x_1 \notin \overline{q_0^x}$, and so $k_2(x_1, x_2) = 0$. We proceed by sub-cases:

– if $x_2 \in \Delta_x$, then $x_2 \in q_0^x$, and so $k_1(x_1, x_2) = r_0^x(x, x_2) = r_0(x_1, x_2)$, thus $(\star)$;

– if $x_2 \notin \Delta_x$, then $k_1(x_1, x_2) = 0$ because $x_2 \notin q_0^x$. On the other hand, since $x_2 \notin \Delta_x$, we have $r_0(x, x_2) = 0$, i.e. $r_0(x_1, x_2) = 0$, thus $(\star)$.

c) If $x_1 \neq x$ and $x_2 = x$; similar to b).

d) If $x_1 \neq x$ and $x_2 \neq x$, then $r_0^x(x_1, x_2) = 0$ whenever $x_1, x_2 \in q_0^x$. Then $k_1(x_1, x_2) = 0$. We proceed by sub-cases:

– if $x_1, x_2 \notin \nabla_x$, then $x_1, x_2 \in \overline{q_0^x}$, and so $k_2(x_1, x_2) = \overline{r^x}_0(x_1, x_2) = r_0(x_1, x_2)$, thus $(\star)$;

– if some $x_i \in \nabla_x$, then $x_i \notin \overline{q_0^x}$, and this implies $k_2(x_1, x_2) = 0$. On the other hand, since $x_i \in \nabla_x$, we have $\forall z \in q_0 \setminus \{x\}$, $r_0(z, x_i) = r_0(x_i, z) = 0$. If we take z as being the another x_j, we obtain $r_0(x_1, x_2) = 0$, so $(\star)$.

We have $A = A_x \otimes \overline{A_x}$, where $A_x, \overline{A_x} \in \mathbf{PM}^\omega$. Therefore $A \in \mathbf{PM}^\omega$.

We are ready now to prove theorem 15:

Proof. $'' \Rightarrow''$ By induction on the structure of A given by definition 9.

$'' \Leftarrow''$ *a)* It will be enough to prove this implication when $q_0 \subset N$.

Indeed, let $\tilde{a}$ be an enumeration of the set $q_0 \cap T$. N is large enough to allow an enumeration $\tilde{y}$ from $N \setminus q$ with $|\tilde{a}| = |\tilde{y}|$.

By lemma 8(2,3), we have $A = \{\tilde{a}/\tilde{a}\}A = \{\tilde{a}/\tilde{y}\}\{\tilde{y}/\tilde{a}\}A$. Let $B = \{\tilde{y}/\tilde{a}\}A = (q', r', \rightarrow')_I$. It is easy to show that the R-machine B is quasi-finite and satisfies conditions 1–4 of the hypothesis. The only difficult part is related to the condition 4. To overcome it, note that the substitution $\{\tilde{y}/\tilde{a}\}$ does not modify the targets of transitions in A. If we suppose that the substitution modify them, then there exists a transition $\alpha \rightarrow_{ij} c$ in A with $c \in \tilde{a}$. Since A satisfies condition $4(b)$ and

$c \in q_0$, it follows that there exists a $j0$-path in the tree H_A which has 0 as root. Thus, we get a contradiction.
B has the required particular form, namely $q'_0 \subset N$. If the implication is true for these machines, we get $B \in \mathbf{PM}^\omega$. We have $\tilde{a} \cap q' = \emptyset$, and $A = (\tilde{y})_{\tilde{a}} B$. So $A \in \mathbf{PM}^\omega$.

b) Moreover, it will be enough to prove this implication when $q_0 \subset N$ and $r_0 = \phi$. By *a)* we can consider $q_0 \subset N$. By keeping in mind the graphical representation of A, we take the labels of those vertices which are linked by tentacle to the hyperedge labeled by q_0 and also are the sources of some transitions, i.e. the set

$$\Delta = \{x \in q_0 \mid x \rightarrow_{0j} \beta\}.$$

We consider also its subset

$$\nabla = \{x \in \Delta \mid \forall y \in q_0,\ r_0(x,y) = r_0(y,x) = 0\}.$$

We build now the following two R-machines:

$\overline{A} = (\overline{q}, \overline{r}, \rightarrow)_I$, where

- $\overline{q}_i = q_i,\ \ i \in I^\star$,
  $\overline{q}_0 = \Delta$;
- $\overline{r}_i = r_i,\ \ i \in I^\star$,
  $\overline{r}_0 = \emptyset$

and

$\underline{A} = (\underline{q}, \underline{r}, \rightarrow^1)_{I_1}$, where $I_1^\star = \emptyset$, and also

- $\underline{q}_0 = q_0 \setminus \nabla$;
- $\underline{r}_0 =_m r_0$;
- $\rightarrow^1_{00} = \emptyset$.

First, by using lemma 43, we show that $\underline{A} \in \mathbf{PM}^\omega$. To this end we have only to verify the equality $\mathbf{useful}(\underline{q}_0) = \underline{q}_0$. This is equivalent to show that $\forall x \in \underline{q}_0$, $\exists y_0 \in \underline{q}_0$ such that $(x, y_0) \in_m \underline{r}_0 \vee (y_0, x) \in_m \underline{r}_0$. Let $x_0 \in \underline{q}_0$. Since $x_0 \in \mathbf{useful}(q_0)$, we have $\exists y_0 \in q_0$ such that $(x_0, y_0) \in_m r_0 \vee (y_0, x_0) \in_m r_0$. Using the multiset equality $r_0 =_m \underline{r}_0$, we obtain $(x_0, y_0) \in_m \underline{r}_0 \vee (y_0, x_0) \in_m \underline{r}_0$.

Second, it is easy to show that the R-machine $\overline{A}$ is quasi-finite and it satisfies the conditions 1–4. Moreover, $\overline{A}$ has the required particular form, namely $\overline{q}_0 \subset N$ and $\overline{r}_0 = \emptyset$. If the implication is true for these machines, we get $\overline{A} \in \mathbf{PM}^\omega$. Let $\underline{A} \otimes \overline{A} = (q', r', \rightarrow)_I$ be a parallel composition. For every $i \in I^\star$, we have $q_i = q'_i$ and $r_i = r'_i$. Immediately, $q_0 = q'_0$ and $r_0 =_m r'_0$. Then $A = \underline{A} \otimes \overline{A}$. Therefore $A \in \mathbf{PM}^\omega$.

c) By induction on h_A, the depth of the tree $H_A = (I, E)$. By *b)*, we assume $q_0 \subset N$ and $r_0 = \emptyset$.

If $h_A = 0$, then H_A is the empty tree having only the root $0 \in I$ and no edges. Then $I^\star = \emptyset$. Trivially, $\rightarrow_{00} = \emptyset$. Since $r_0 = \emptyset$, we get $\mathbf{useful}(q_0) = \emptyset$. A verifies condition 3, and so $q_0 = \emptyset$. We obtain that A is $nil \in \mathbf{PM}^\omega$.

If $h_A > 0$, then we proceed again by induction on $|q_0|$.

i) If $|q_0| = 1$, then let $q_0 = \{x\}$. We take the (nonempty) set $K = \{k \in I \mid (0,k) \in E\}$. We consider an arbitrary k in K. Since A verifies conditions 1 and 4(a), there exists a unique $a_k \in q_k \cap T$ such that $x \rightarrow_{0k} a_k$. We take the set $I_k \subset I$ containing 0 given by $I^\star_k = \{j \in I \mid$ there exists a kj-path in $H_A\}$. We consider a set $\nabla_k \subset q_k$. If $(a_k, \alpha) \in_m r_k$, $(\alpha, a_k) \in_m r_k$ or $a_k \rightarrow_{kj} \alpha$ for some α, then we set ∇_k to $\emptyset$, otherwise we set it to $\{a_k\}$.

We build now a R-machine $A_k = (q^k, r^k, \rightarrow^k)_{I_k}$, where

- $q^k_i = q_i,\ i \in I^\star_k$,
  $q^k_0 = q_k \setminus \nabla_k$;
- $r^k_i = r_i,\ i \in I^\star_k$,
  $r^k_0 =_m r_k$;
- $\rightarrow^k_{ij} = \rightarrow_{ij},\ i, j \in I^\star_k$,
  $\rightarrow^k_{0j} = \rightarrow_{kj},\ j \in I^\star_k$,
  $\rightarrow^k_{ij} = \emptyset$, otherwise .

One can prove that A_k is quasi-finite, satisfies the conditions 1–4, and $h_{A_k} < h_A$. By induction, we obtain $A_k \in \mathbf{PM}^\omega$.

We can choose $y \in N \setminus q$. Since $q^k \subset q$, we have $y \notin q^k$. If B_k is the machine $\{y/a_k\}A_k$, then one can prove that B_k is quasi-finite, satisfies conditions 1–4, and $h_{B_k} = h_{A_k}$. The only difficult part is related to the condition 4. To overcome it, note that the substitution $\{y/a_k\}$ does not modify the targets of the transitions of A_k. Suppose this substitution modifies them. Then there exists $\alpha \rightarrow^k_{ij} a_k$ a transition in A_k. By definition of $\rightarrow^k$, we obtain $\alpha \rightarrow_{ij} a_k$, where $j \in I^\star_k$. Since A satisfies condition $4(b)$ and $a_k \in q_k$, there exists a jk-path in the tree H_A. Then this tree has a cycle. Contradiction. Consequently, by induction we get $B_k \in \mathbf{PM}^\omega$.

The machine $C_k = (x, y)_{a_k,k} B_k$ is well-defined, then $C_k \in \mathbf{PM}^\omega$. Let $\tilde{c}$ be an enumeration of the set

$$\bigcup_{\substack{k,l \in K \\ k \neq l}} (q^k \cap q^l \cap T)$$

We can choose an enumeration $\tilde{z}$ from $N \setminus q$, $y \notin \tilde{z}$ with $|\tilde{c}| = |\tilde{z}|$. We show by contradiction that $a_k \notin \tilde{c}$. Suppose there exists $l \in K$ $(l \neq k)$ such that $a_k \in q^l$. Then $a_k \in q_j$, where $j \in I^\star_l \cup \{l\}$. Since A satisfies condition $4(b)$ and $a_k \in q_j$, then there exist two $0j$-path which are distinct in the tree H_A. Contradiction. One can prove similarly by induction that $\{\tilde{z}/\tilde{c}\}B_k \in \mathbf{PM}^\omega$. By lemma 12(3), we get $\{\tilde{z}/\tilde{c}\}C_k = (x, y)_{a_k,k}\{\tilde{z}/\tilde{c}\}B_k$. If D_k is the machine $\{\tilde{z}/\tilde{c}\}C_k$, then $D_k \in \mathbf{PM}^\omega$.

The machine $B = \otimes_{k \in K} D_k$ is well-defined, then $B \in \mathbf{PM}^\omega$. One can prove that $A = (\tilde{z})_{\tilde{c}} B$. Therefore $A \in \mathbf{PM}^\omega$.

ii) If $|q_0| > 1$, then there exists $x \in q_0$. Consider the set $K = \{k \in I \mid (0, k) \in E\}$, and also its subsets $K_1 = \{k \in K \mid x \rightarrow_{0k} a\}$ and $K_2 = K \setminus K_1$.

We build now the following two R-machines

$A_x = (q^x, r^x, \rightarrow^1)_{I_1}$, where $I^\star_1 = \{j \in I \mid H_A$ has a kj-path, $k \in K_1\}$, and also

- $q_i^x = q_i,\ i \in I_1^\star$,
  $q_0^x = \{x\}$;
- $r_i^x = r_i,\ i \in I_1^\star$,
  $r_0^x = \emptyset$;
- $\rightarrow_{ij}^1 = \rightarrow_{ij},\ i, j \in I_1$

and

$\overline{A_x} = (\overline{q^x}, \overline{r^x}, \rightarrow^2)_{I_2}$, where $I_2^\star = \{j \in I \mid H_A$ has a kj-path, $k \in K_2\}$, and also

- $\overline{q_i^x} = q_i,\ i \in I_2^\star$,
  $\overline{q_0^x} = q_0 \setminus \{x\}$;
- $\overline{r_i^x} = r_i,\ i \in I_2^\star$,
  $\overline{r_0^x} = \emptyset$;
- $\rightarrow_{ij}^2 = \rightarrow_{ij},\ i, j \in I_2$.

By induction, it follows that $A_x, \overline{A_x} \in \mathbf{PM}^\omega$. Let $\tilde{c}$ be an enumeration of the set $q^x \cap \overline{q^x} \cap T$. We can choose $\tilde{z}$ an enumeration from $N \setminus q$ with $|\tilde{c}| = |\tilde{z}|$. As above, one can prove that the following machines $B_x = \{\tilde{z}/\tilde{c}\}A_x$ and $\overline{B_x} = \{\tilde{z}/\tilde{c}\}\overline{A_x}$ are from $\mathbf{PM}^\omega$. $B = B_x \otimes \overline{B_x}$ is well-defined and thus $B \in \mathbf{PM}^\omega$. One can prove that $A = (\tilde{z})_{\tilde{c}}B$. Therefore $A \in \mathbf{PM}^\omega$.

B Proofs of the reduction lemmas

B.1 Sketch of proof for lemma 30

Proof. Let $A = (q^A, r^A, \rightarrow^A)_{I_A}$ and $B = (q^B, r^B, \rightarrow^B)_{I_B}$. There exist two bijective mappings $\phi : q^A \rightarrow q^B$ and $\sigma : I_A \rightarrow I_B$ with $\sigma(0) = 0$ s.t. $(\phi, \sigma) : A \cong B$. Suppose we have the reduction $A \mapsto^\rho \mathcal{A}$ by some $\rho = \langle \alpha_1, \gamma, \alpha_2; k\rangle$. One can prove that $\pi = \langle \phi(\alpha_1), \phi(\gamma), \phi(\alpha_2); \sigma(k)\rangle$ is a reduction tuple of B and $B \mapsto^\pi \mathcal{A}$.

B.2 Sketch of proof for lemma 31

Proof. 1. There exists $A \in \mathcal{A}_1$ s.t. $(\tilde{x})\mathcal{A}_1 = [(\tilde{x})_{\tilde{a}}A]_\cong$. We have the reduction $A \mapsto^\rho \mathcal{A}_2$ by a $\rho = \langle \alpha_1, \gamma, \alpha_2; k\rangle$. One can prove that $\pi = \langle \{\tilde{a}/\tilde{x}\}\alpha_1, \{\tilde{a}/\tilde{x}\}\gamma, \alpha_2; k\rangle$ is a reduction tuple of $(\tilde{x})_{\tilde{a}}A$, and $(\tilde{x})_{\tilde{a}}A \mapsto^\pi \mathcal{A}_2$. Thus $(\tilde{x})\mathcal{A}_1 \Rightarrow (\tilde{x})\mathcal{A}_2$.

2. There exists $A \in \mathcal{A}_1$ s.t. $(\tilde{x})\mathcal{A}_1 = [(\tilde{x})_{\tilde{a}}A]_\cong$. Let $(\tilde{x})_{\tilde{a}}A \mapsto^\pi \mathcal{A}$. This means that $\pi = \langle \{\tilde{a}/\tilde{x}\}\alpha_1, \{\tilde{a}/\tilde{x}\}\gamma, \alpha_2; k\rangle$, where $\rho = \langle \alpha_1, \gamma, \alpha_2; k\rangle$ is a reduction tuple of A. If we define $\mathcal{A}_2$ by $A \mapsto^\rho \mathcal{A}_2$, then $\mathcal{A}_1 \Rightarrow \mathcal{A}_2$. As in 1, we obtain $(\tilde{x})_{\tilde{a}}A \mapsto^\pi (\tilde{x})\mathcal{A}_2$. Thus $\mathcal{A} = (\tilde{x})\mathcal{A}_2$.

B.3 Sketch of proof for lemma 32

Proof.

1. An easy verification by using the definitions.

2. There exist $A_1 \in \mathcal{A}_1$ and $A \in \mathcal{A}$ s.t. $\mathcal{A}_1 \otimes \mathcal{A} = [A_1 \otimes A]_\cong$. We have a reduction $A_1 \mapsto^\rho \mathcal{A}_2$. One can prove that ρ is a reduction tuple of $A_1 \otimes A$ and $A_1 \otimes A \mapsto^\rho \mathcal{A}_2 \otimes \mathcal{A}$. Thus $\mathcal{A}_1 \otimes \mathcal{A} \Rightarrow \mathcal{A}_2 \otimes \mathcal{A}$. The other part follows similarly.

3. There exist $A \in \mathcal{A}$ and $B \in \mathcal{B}$ s.t. $\mathcal{A} \otimes \mathcal{B} = [\, A \otimes B\,]_{\cong}$. We have the reduction $A \otimes B \mapsto^{\rho} \mathcal{C}$ by a reduction tuple $\rho = \langle \alpha_1, \gamma, \alpha_2; k \rangle$. Let $A = (q^A, r^A, \rightarrow^A)_{I_A}$ and $B = (q^B, r^B, \rightarrow^B)_{I_B}$. There are two cases:

• ρ is not coming from the parallel composition. Then either i) ρ is a reduction tuple of A or ii) ρ is a reduction tuple of B. We analyse i). If we define $\mathcal{A}'$ by $A \mapsto^{\rho} \mathcal{A}'$, then $\mathcal{A} \Rightarrow \mathcal{A}'$. As in 2, we have $A \otimes B \mapsto^{\rho} \mathcal{A}' \otimes \mathcal{B}$. Thus $\mathcal{C} = \mathcal{A}' \otimes \mathcal{B}$, i.e. $1a$). Similarly, ii) leads to $1b$).

• ρ is coming from the parallel composition. Then either i) $(\alpha_1, \gamma) \in_m r_0^A$ and $\gamma \rightarrow_{0k}^B \alpha_2$ or ii) $(\alpha_1, \gamma) \in_m r_0^B$ and $\gamma \rightarrow_{0k}^A \alpha_2$. We analyse i). $\gamma \notin T$ because the sets $q^A \cap T$ and $q^B \cap T$ are disjoint. Let $x = \gamma$ and $I_k = \{k\} \cup \{\, i \in I_B \mid \exists\ ki\text{-path in } H_B\}$. Let $\tilde{c}$ be an enumeration of the set

$$\bigcup_{\substack{i \in I_k \\ j \in I_B \setminus I_k}} q_i^B \cap q_j^B \cap T \subseteq \bigcup_{\substack{i,j \in I_B \\ i \neq j}} q_i^B \cap q_j^B \cap T$$

Immediately, $|\tilde{c}| \leq \kappa(\mathcal{B})$. There exists an enumeration $\tilde{u}$ from $N \setminus \lambda(\mathcal{A} \otimes \mathcal{B})$ with $|\tilde{c}| = |\tilde{u}|$. Then $B = \{\tilde{c}/\tilde{c}\}B = \{\tilde{c}/\tilde{u}\}\{\tilde{u}/\tilde{c}\}B$. As in the proof of theorem 15, we find a decomposition $\{\tilde{u}/\tilde{c}\}B = (x, v)_{\alpha_2, k} B_1 \otimes B_2$ (take B_1 the substructure of $\{\tilde{u}/\tilde{c}\}B$ induced by I_k and considering the state k as the initial one, and B_2 the remaining part), where v is chosen from $N \setminus \lambda(\mathcal{A} \otimes \mathcal{B})$ and $v \notin \tilde{u}$. Let $\mathcal{B}_i = [B_i]_{\cong},\ i = 1, 2$. We proceed by sub-cases.

⋆ If $\alpha_1 \in N$, then let $z = \alpha_1$. We have a decomposition $A = A_1 \otimes 1(x, z)$. Let $\mathcal{A}_1 = [A_1]_{\cong}$. Then $A \otimes B = A_1 \otimes (\tilde{u})_{\tilde{c}}(\,(\,1(x,z) \otimes (x,y)_{\alpha_2,k} B_1\,) \otimes B_2\,)$. As in 1, we have $1(x,z) \otimes (x,v)_{\alpha_2,k} B_1 \mapsto^{\rho} \{z/v\}\mathcal{B}_1$. As in 2, we have $(\,1(x,z) \otimes (x,y)_{\alpha_2,k} B_1\,) \otimes B_2 \mapsto^{\rho} \{z/v\}\mathcal{B}_1 \otimes \mathcal{B}_2$. As in lemma 31, we have $(\tilde{u})_{\tilde{c}}(\,(\,1(x,z) \otimes (x,v)_{\alpha_2,k} B_1\,) \otimes B_2\,) \mapsto^{\pi} (\tilde{u})(\{z/v\}\mathcal{B}_1 \otimes \mathcal{B}_2)$ where $\pi = \langle \{\tilde{c}/\tilde{u}\}\alpha_1, \{\tilde{c}/\tilde{u}\}\gamma, \alpha_2; k \rangle = \rho$. Finally as in 2, we have $A \otimes B \mapsto^{\rho} \mathcal{A}_1 \otimes (\tilde{u})(\{z/v\}\mathcal{B}_1 \otimes \mathcal{B}_2)$. So $\mathcal{C} = \mathcal{A}_1 \otimes (\tilde{u})(\{z/v\}\mathcal{B}_1 \otimes \mathcal{B}_2)$. Thus we obtain 2a).

⋆ If $\alpha_1 \in T$, then we consider $w \in N \setminus \lambda(\mathcal{A} \otimes \mathcal{B})$, $w \neq v$, and $w \notin \tilde{u}$. Then $A = \{\alpha_1/\alpha_1\}A = \{\alpha_1/w\}\{w/\alpha_1\}A$. There exists a decomposition $\{w/\alpha_1\}A = A_1 \otimes 1(x, w)$. Then $A = (w)_{\alpha_1}(A_1 \otimes 1(x, w)\,)$. Similar to the previous case, we obtain 3a). Similarly, ii) leads to 2b), and respectively 3b).