Petri nets and algebraic calculi of processes

Gérard Boudol, Gérard Roucairol, Robert de Simone

HAL Id: inria-00076146
https://inria.hal.science/inria-00076146

Submitted on 24 May 2006

Rapports de Recherche

PETRI NETS
AND ALGEBRAIC CALCULI
OF PROCESSES

Gérard BOUDOL
Gérard ROUCAIROL
Robert de SIMONE

PETRI NETS AND ALGEBRAIC CALCULI OF PROCESSES

Gérard Boudol(*)
Gérard Roucairol(**)
Robert de Simone(*)

(*) INRIA, Sophia Antipolis, 06565-Valbonne, France
(**)BULL, Centre de recherches, 68 Route de Versailles,
78430-Louveciennes, France

Abstract

We show that, as transition systems, Petri nets may be expressed by terms of a calculus of processes which is a variant of Milner's SCCS. We then prove that the class of labelled nets forms a subcalculus, thus an algebra, with *juxtaposition, adding condition* and *labelling* as primitive operations.Finally we introduce *rational machines* which express explicit synchronizations on nets.

Résumé

Nous montrons que les réseaux de Petri, considérés comme des systèmes de transitions, peuvent être décrits par des termes d'un calcul de processus; le calcul que nous utilisons est une variante du calcul SCCS de Milner. La classe des réseaux étiquetés forme un sous-calcul, donc une algèbre, dont les opérations primitives sont la *juxtaposition*, l' *ajout de condition* et l' *étiquetage*. Enfin nous introduisons les *machines rationnelles* qui permettent d'exprimer des synchronisations explicites sur les réseaux.

1 - INTRODUCTION

In the last few years several mathematical models of concurrent and communicating systems did emerge. One of the best known is that of Petri nets, which gave rise to a considerable amount of theoretical developments (see for example [15,4,5]). In some sense a Petri net is a dynamic pictorial description for a non deterministic asynchronous concurrent system. This graphical aspect involves a slight defect : it is not clear how to recognize "subgames" in the picture, which is given as a whole, and reset the components. In other words, we may need some syntax to build nets. Previous efforts in this direction may be found in [6,10].

We aim here at proposing such a syntax. We use as framework a more recent abstract model, that of Milner's algebraic calculi of processes ([12,14]). This model consists first in a free algebra of terms : primitive operations are parallel composition, synchronization or desynchronization mechanisms, and so on : whence the algebraic character. Second there are actions labelled transition relations between terms. These relations are structurally defined, that is the behaviour of a compound term is built from the behaviour of its components by means of rules. Therefore each term is endowed with a labelled transition system as its operational semantics. Finally an equivalence relation on terms is introduced. It is a congruence for the operators of the algebra and respects the transitions abilities. Thus we have equational laws, whence a calculus, and each term denotes a *process:* a transition system on equivalence classes. The calculus provides this way a syntax for some universe of processes.

Milner already showed that the static (pictorial) aspect of nets can be modeled in a calculus, that of (free) flow algebra. Concerning dynamics, one of his most fruitful ideas was to realize that the set of actions should be thought of as an algebraic structure too [13,14]. Specifically, actions will form an abelian monoïd : in order to perform simultaneously two actions one performs their product. This formalizes the idea of global actions of a system. Moreover some actions may have an inverse. This handles a rudimentary form of communication (handshake).

We set a relationship from Petri's to Milner's mathematical models of processes. Roughly speaking, we translate a Petri net into a term of a calculus ; this translation is semantically sound in the sense that the labelled transition system on markings generated by a net and the operational semantics of its target term are the same. A converse study can be found in [9], where the authors intend to provide a Petri Net semantics to CCS. The calculus we will use, called MEIJE (cf [2,3]), lies on an "asynchronous" (not in the technical sense of asynchrony developped in [14]) parallel composition, together with some synchronization primitives. We shall present it briefly in the second section. In fact, this calculus is just an equivalent formulation of the "finitary" version of Milner's synchronous calculus (see [3,13]). We show in the third section that, as far as they denote transition systems, Petri nets may always be exactly expressed as terms of this calculus. The crucial tool here rests in the monoid of actions which formalizes notions of simultaneity and communication. In the fourth section we present the labelled nets as an algebra, build with the following primitive operations : juxtaposition, adding conditions and labelling, from a simple set of generators : clocks. For instance this syntax allows us to describe things like "the addition of a new place to a net", and so on...

In the last technical section we introduce a generalization of Petri nets that we have called rational parallel place machines ([3]). These machines are designed for the expression of synchronization constraints on parallel processes depicted by Petri nets. We discuss the semantical aspects of our propositions in the conclusion.

2 - THE MEIJE CALCULUS

2.1. Actions

First we describe the monoïd of actions which is the basis of our calculus. This monoïd $\mathbb{M}$ is the two fold product of :

(1) the free abelian monoïd generated by some given countable set A of *atomic actions*. Thus, loosely speaking, this monoïd is that of instantaneous events, which do not interact with each other. The product $a.b$ means a and b happening together, or the global action composed of a and b. Here from A we get multisets of atomic actions, with a multiplicative notation. For instance the multiset $\{a,b,b,a,a,c\}$ of atomic actions a,b,c is denoted a^3b^2c.

(2) The free abelian group generated by some given countable set S of *signals*. Each of the element of S, say s, is a synchronization or communication action, endowed with an inverse $\bar{s}$; the simultaneous occurrence $s.\bar{s}$ of these actions establishes a communication, or a handshake, which is a private act, only showing the unit action 1.

The product of $a \in \mathbb{M}$ and $b \in \mathbb{M}$ will be denoted by $a.b$ and the unit by 1. If B is a subset of $A \cup S$ we shall denote by $\mathbb{M}\backslash B$ the substructure of $\mathbb{M}$ generated by $(A \cup S)-B$.

2.2 Syntax of terms : the Meije operators

Now let X be a countable set of variables, which will serve as identifiers in order to define recursive processes. The syntax for the terms of the MEIJE calculus ([2,3]) is given by the following :

(i) $\mathbb{0}$ is a term *(inaction)*, and any variable is a term ;

(ii)   guard or *action* : if $a \in \mathbb{M}$ and p is a term, then $a :p$ is a term (a before p) ;

(iii) *morphism* : if φ is an endomorphism of $\mathbb{M}$ and p is a term, then $<\varphi>p$ is a term (p renamed by φ) ;

(iv) *restriction* : if $\alpha \in S$ and p is a term then $p \backslash \alpha$ is a term (p restricted on signal $\overset{\bullet}{\alpha}$) ;

(v) *recursive definition* : if $x_1, \ldots, x_n$ are variables and $p_1, \ldots, p_n$ are terms, then for $1 \le i \le n$ (x_i **where** $x_1 = p_1, \ldots, x_n = p_n$) is a term.

These are, with slight lexical variations (but the same semantics, see below), among CCS's or SCCS's primitives.

This is not the case of the following :

(vi) *asynchronous parallel composition* : if p and q are terms, then ($p \parallel q$) is a term ;

(vii) *ticking* : if $\alpha \in S$ and p is a term, then $\alpha * p$ is a term.

Free and bound occurrences of variables are defined in the usual way, and we denote by

$$[q_1/x_1, \ldots, q_n/x_n]p$$

the term obtained by substituting the q_i's to the x_i's at their free occurrences in p (renaming bound variables of p if necessary).

2.3 Semantics : the transition rules

The transitions relations $\overset{a}{\to}$ (where $a \in \mathbb{M}$) between terms are the least ones satisfying the following rules :

(R1) $\quad a : p \overset{a}{\to} p$

(R2) $\quad$ if $p \overset{a}{\to} p'$ then $<\varphi>p \overset{\varphi(a)}{\longrightarrow} <\varphi>p'$

(R3) $\quad$ if $p \overset{a}{\to} p'$ and $a \in \mathbb{M} \backslash \alpha$ (that is neither α nor $\overline{\alpha}$ appears as an irreducible factor of a) then $p \backslash \alpha \overset{a}{\to} p' \backslash \alpha$

(R4) $\quad$ if, for $1 \le j \le n$, $q_j = (x_j$ **where** $x_1 = p_1, \ldots, x_n = p_n)$ and $[q_1/x_1, \ldots, q_n/x_n]p_i \overset{a}{\to} p$ then $q_i \overset{a}{\to} p$.

There are three rules defining the semantics of the parallel composition, according to the idea that the components are independant :

(R5) $\quad$ if $p \overset{a}{\to} p'$ then ($p \parallel q$) $\overset{a}{\to}$ ($p' \parallel q$)

(R6) $\quad$ if $p \overset{a}{\to} p'$ and $q \overset{b}{\to} q'$ then ($p \parallel q$) $\overset{a b}{\longrightarrow}$ ($p' \parallel q'$)

(R7) $\quad$ if $q \overset{b}{\to} q'$ then ($p \parallel q$) $\overset{b}{\to}$ ($p \parallel q'$)

Here either both or only one of two components set in parallel are assume to work at any time. It is the notion of asynchrony we shall adopt throughout.

Finally for the synchronization primitive of MEIJE :

(R8) $\quad$ if $p \overset{a}{\to} p'$ then $\alpha * p \overset{a.\alpha}{\longrightarrow} \alpha * p'$ (signal α adds itself to every transition of p from now on).

Thus $t \overset{a}{\to} t'$ *iff* this transition has a proof following these rules.

Examples

a - the transition graph of the term $p = ((\alpha * (a : b : 0) \parallel \alpha * (c : 0)) \parallel \overline{a} : 0) \backslash \alpha$ is

$$p \overset{a}{\nearrow} ((\alpha * (b : 0) \parallel \alpha * (c : 0)) \parallel 0) \backslash \alpha$$
$$\overset{c}{\searrow} ((\alpha * (a : b : 0) \parallel \alpha * 0) \parallel 0) \backslash \alpha$$

b - With each action a we associate a *clock* on a

$$h_a =_{def} (x \text{ where } x = a : x)$$

and it is easily seen that $h_a \to h_a$ (and moreover $h_a \to p$ implies $b = a$ and $p = h_a$).

This example already shows the most basic way to have a loop in our transition systems. Since recourse to no operator is limited inside recursive definitions, dynamic evolution of a term's topology (its $\|$, $\backslash$ and $a *$ operators) is feasible, and bring the calculus much of its power, see [18]. The two following examples, extended clocks, illustrate this :

c - $\qquad p =_{def} (x \text{ where } x = 1 : (a * x))$

Here the number of a actions recalls a current date :

$$p \xrightarrow{1} a * p \xrightarrow{a} a * a * p \xrightarrow{a^2} \cdots p_n \xrightarrow{a^n} p_{n+1}$$

where $p_0 = p$ and $p_{n+1} = a * p_n$.

d - $\qquad p =_{def} (x \text{ where } x = (a : 0 \| x))$

Here $a : 0$ is an "elementary impulse", and the term p may always execute an arbitrary number of them :

$$\forall \in \mathbb{N}, n > 0 \; \exists p_n \quad p \xrightarrow{a_n} p_n$$

where $p_0 = p$ and $p_{n+1} = (0 \| p_n)$. And in fact p_n and p have the same behaviour, see the following section.

For those two last examples the reader is urged to try and show, in the previous deduction system, the adequacy of the terms in the face of their claimed behaviour.

2.4 Semantics : equivalences of terms

Let $A_{(A,S)}$ be the set of closed terms, that is without free variables : these are the *agents* of the calculus, which denote processes, as we shall just see. Let $\sim$ be the coarsest equivalence relation between agents which satisfy the property of commutation (or compatibility) with transitions :

$$p \sim q \text{ and } p \xrightarrow{a} p' \Rightarrow \exists q' \; q \xrightarrow{a} q' \text{ and } p' \sim q'$$

It is a congruence relation over the algebra of agents, called the *strong congruence* (see [14,3]). The set $P_{(A,S)} = A_{(A,S)}/\sim$ of equivalence classes is the set of (MEIJE) *processes*. In this set we still have transition relations :

$$\{ p \} \xrightarrow{a} \{ p' \} \; iff \; \exists q : p \xrightarrow{a} q \text{ and } q \sim p'$$

(where $\{ p \}$ denotes the equivalence class of the agent p).

Equational laws where verified somewhere else ([2]), among which :

(L1) $(p \| q) \sim (q \| p)$ (*commutativity*)

(L2) $(p \| (q \| r)) \sim ((p \| q) \| r)$ (*associativity*)

which allow one to write $(p_1 \| \cdots \| p_n)$

(L3) $(0 \| p) \sim p$ (*unit*)

And, for $\alpha, \beta \in S$:

(L4) $(p \backslash \alpha) \backslash \alpha \sim p \backslash \alpha$

(L5) $(p \backslash \alpha) \backslash \beta \sim (p \backslash \beta) \backslash \alpha$

Thus for $\{\alpha_1, \ldots, \alpha_n\} \subseteq S$ we may write $p \backslash \{\alpha_1, \ldots, \alpha_n\}$ for $(\ldots (p \backslash \alpha_1) \backslash \cdots \backslash \alpha_n)$.

Another law will be of use : if, as in (R4), we let for $1 \leq i \leq n$ $q_i = (x_i \text{ where } x_1 = p_1, \dots, x_n = p_n)$ then

(L6) $q_i \sim [q_1/x_1, \dots, q_n/x_n]p_i$ (fixpoint).

While agents denote processes, the expressions of the calculus, with free variables, denote functions on processes. If t is such an expression, with free variables $x_1, \dots, x_n$ we define

$$\hat{t}(\|p_1\|, \dots, \|p_n\|) = \| [p_1/x_1, \dots, p_n/x_n]t \|$$

for agents $p_1, \dots, p_n$.

For example, the *interleaving* operator, specified on $P_{(A,S)}$ by the rules :

(1)   if $p \overset{a}{\to} p'$ then $(p \mid q) \overset{a}{\to} (p' \mid q)$

(2)   if $q \overset{b}{\to} q'$ then $(p \mid q) \overset{b}{\to} (p \mid q')$

is realized through the MEIJE expression

$$(x \mid y) = (\alpha^* x \parallel \alpha^* y \parallel h_{\bar{a}}) \setminus \alpha$$

Some other examples of such *derived operators* (for which $\sim$ is still a congruence) are given in [2,3]. We get a *subcalculus* by taking some of these derived operators as being primitive.

3 - EXPRESSING PETRI NETS IN MEIJE

3.1. Petri Nets

Let us first briefly fix our notations (we assume here familiarity with Petri nets, see [15,4,5]). We deal with Petri nets allowing multiple arcs between transitions and places. Thus a net τ is a structure $(P, T, Pre, Post)$ where

- $P = \{p_1, \dots, p_K\}$ is the finite non-empty set of *places*

- $T = \{t_1, \dots, t_N\}$ is the finite non-empty set of *transitions* (with some ambiguity in the use of this word)

- $Pre : P \times T \to \mathbb{N}$ ($\mathbb{N}$ is the set of non-negative integers)

and $Post : T \times P \to \mathbb{N}$ are the numerical functions setting the preconditions and postconditions of the firing of transitions.

A *marking* μ on the net τ is a map from P into $\mathbb{N}$: $\mu(p)$ is the number of *tokens* in the place p for this marking (and we note $\mu(p_i) = \mu_i$). A transition t is *enabled* by the marking μ, a property noted $(\tau, \mu)[t >$ *iff*

$$\forall p \in P \quad \mu(p) \geq Pre(p, t)$$

Under this condition, by *firing* t we get the marking μ' on τ such that

$$\forall p \in P \quad \mu'(p) = \mu(p) - Pre(p, t) + Post(t, p)$$

We write that $(\tau, \mu)[t > (\tau, \mu')$, which is the transition relation between states of the net.

One may extend this relation in order to define the *simultaneous* firing of transitions. Given a non-empty subset U of T, we say that U is enabled by the marking μ *iff*

$$\forall p \in P \quad \mu(p) \geq \sum_{t \in U} Pre(p, t)$$

and we denote again this fact by $(\tau, \mu)[U >$. Then, by firing simultaneously (or : in parallel) the transitions of U, we get the marking μ' :

$$\forall p \in P \quad \mu'(p) = \mu(p) + \sum_{t \in U} [Post(t, p) - Pre(p, t)]$$

The extended transition relation is again denoted

$$(\tau,\mu)\,[U>(\tau,\mu')]$$

(This is the definition of [17] which seems to correspond to Petri's original idea about concurrency).

3.2. The Translation

Now we want to find for each marked Petri net a MEIJE agent which, as far as the two denote processes, is "isomorphic" to this net. The idea is to consider places and transitions of the net as processes, and then to express them by agents and find an appropriate communication structure between them.

3.2.1. From places to Bags

A place of a net is no more than a *bag*, into which we can put -and there after remove- tokens. Let us denote for a moment by a and b respectively the atomic actions of putting and removing a token. Then a bag is a process, the behaviour of which is the following :

one can, simultaneously

- put (performing the action a) as many tokens as one wishes

- remove (performing b) no more tokens than there is already in the bag.

Thus, if we denote by $\Omega^{(k)}$ a bag initially containing k tokens, its specification is :

$$\forall\, m \in \mathbb{N}\, \forall\, n \in \mathbb{N}\, n \le k \,,\, m+n > 0\,,\, \Omega^{(k)} \xrightarrow{a^m b^n} \Omega^{(k+m-n)}$$

Now let $\omega^{(k)}$ be the sequence of MEIJE agents defined by

$$\omega^{(o)} = (x \text{ where } x = (a:b:\mathbb{O} \parallel x))$$
$$\omega^{(k+1)} = (b:\mathbb{O} \parallel \omega^{(k)})$$

If we regard $a:b:\mathbb{O}$ as an "elementary cell" (we could also have taken a cell $(x \text{ where } x = a:b:x)$.) $\omega^{(o)}$ is just a potentially infinite juxtaposition of such cells.

We shall next show that $\omega^{(k)}$ meets the $\Omega^{(k)}$ specification requirement for a bag.

lemma 1

$$\forall\, k \in \mathbb{N}\quad \omega^{(k)} \sim (a:b:\mathbb{O} \parallel \omega^{(k)})$$

proof : we proceed by induction on k

- if $k=0$, then we apply the law $(L\,6)$ and immediately get the result

$$-(a:b:\mathbb{O} \parallel \omega^{(k+1)}) = (a:b:\mathbb{O} \parallel (b:\mathbb{O} \parallel \omega^{(k)}))\ (definition)$$

$$\sim (b:\mathbb{O} \parallel (a:b:\mathbb{O} \parallel \omega^{(k)}))\ (by\ L\,1, L\,2)$$

$$\sim (b:\mathbb{O} \parallel \omega^{(k)})\ (induction\ hypothesis)$$

$$= \omega^{(k+1)}\ (definition)\quad\blacksquare$$

lemma 2

$$\{\omega^{(o)}\} \xrightarrow{c} \{p\}\ \ iff\ \exists\, k \in \mathbb{N}\,,\, k>0\ \ c=a^k\ and\ \{p\} = \{\omega^{(k)}\}$$

proof :

(1) by induction on k ($k>0$) we prove that $\omega^{(o)} \xrightarrow{a^k} \omega^{(k)}$

    (1.1) - We have (rule $R\,1$) $a:b:\mathbb{O} \xrightarrow{a} b:\mathbb{O}$

    whence (rule $R\,5$) $(a:b:\mathbb{O} \parallel \omega^{(o)}) \xrightarrow{a} (b:\mathbb{O} \parallel \omega^{(o)}) = \omega^{(1)}$

Thus (rule R 4) $\omega^{(o)} \xrightarrow{a} \omega^{(1)}$

(1.2) - By induction hypothesis $\omega^{(o)} \xrightarrow{a^k} \omega^{(k)}$. Since $a : b : \mathbb{0} \xrightarrow{a} b : \mathbb{0}$ we have (rule R6)

$$(a : b : \mathbb{0} \| \omega^{(o)}) \xrightarrow{a^{k+1}} (b : \mathbb{0} \| \omega^{(k)}) = \omega^{(k+1)}$$

Thus (rule R 4) $\omega^{(o)} \xrightarrow{a^{k+1}} \omega^{(k+1)}$

(2) We prove the converse by induction on the definition of the transition relations.

If $\{ \omega^{(o)} \} \xrightarrow{c} \{ p \}$ then (by definition of transitions in $P_{(A,S)}$ and rule R 4)

$$\exists q \sim p \quad (a : b : \mathbb{0} \| \omega^{(o)}) \xrightarrow{c} q$$

then

(2.1) - $c = a$ and $q = (b : \mathbb{0} \| \omega^{(o)}) = \omega^{(1)}$ (rules R 1 and R 5)

or

(2.2) - (rule R 7) $\exists q' : \omega^{(o)} \xrightarrow{c} q'$ and $q = (a : b : \mathbb{0} \| q')$

then by induction hypothesis

$$\exists k \in \mathbb{N}, k > 0 \ c = a^k \text{ and } q' \sim \omega^{(k)}$$

Thus $q \sim (a : b : \mathbb{0} \| \omega^{(k)})$ since $\sim$ is a congruence and by lemma 1 $q \sim \omega^{(k)}$

or

(2.3) - (rules R 6 and R 1)

$$\exists c' \ \exists q' : \omega^{(o)} \xrightarrow{c'} q', \ c = a.c' \text{ and } q = (b : \mathbb{0} \| q')$$

by induction hypothesis

$$\exists k > 0 \ c' = a^k \text{ and } q' \sim \omega^{(k)}$$

thus $c = a^{k+1}$ and ($\sim$ congruence) $q \sim (b : \mathbb{0} \| \omega^{(k)}) = \omega^{(k+1)}$ ∎

We get the immediate corollary :

lemma 3

$$\forall k \in \mathbb{N} \ \{ \omega^{(k)} \} \xrightarrow{c} \{ p \} \ \text{iff} \ \exists n, m \in \mathbb{N}, n \le k, m + n > 0$$

and $c = a^m b^n, p \sim \omega^{(k+m-n)}$

(proof : by induction on k and case analysis, obvious).

This proves that we did correctly model the concept of bag by the MEIJE agents $\omega^{(k)}$, that is :

$$\{ \omega^{(k)} \} = \Omega^{(k)}$$

In the synchronous calculus of Milner ([13,14]) we should have written

$$\omega^{(k)} = fix_k \ \{ x_j = \sum_{n \le j} ((b^n : \mathbb{1}) \times \sum_{m \in \mathbb{N}} a^m : x_{j+m-n}) \ / \ j \in \mathbb{N} \}$$

where $\mathbb{1} = h_1 = (x \text{ where } x = 1 : x)$.

3.2.2. From Transitions to Clocks

The second step consists in building agents for transitions. The semantics of the firing of a transition t is that, simultaneously :

- one remove $Pre(p, t)$ tokens from the input places from t
- one performs t
- one put $Post(t, p)$ tokens in the output places of t.

In order to provide this, we assume that places and transitions communicate by means of signals. Technically we suppose with the notations of (3.1), that:

- the set A of atomic actions includes $\{t_1, \ldots, t_N\}$
- the set S of signals includes $\{\alpha_i, \beta_i \, / \, 1 \leq i \leq K\}$

Then performing $\overline{\alpha_i}$ means putting a token in the place p_i, while p_i performs the action α_i, that is receives this token. When p_i performs $\overline{\beta_i}$, a token is sent from p_i, while performing β_i means receiving this token. Thus each place p_i, with some initial content (marking) is now a bag among

$$\Theta_i^{(o)} = (x \text{ where } x = (\alpha_i : \overline{\beta_i} : \mathbb{0} \parallel x))$$

$$\Theta_i^{(k+1)} = (\overline{\beta_i} : \mathbb{0} \parallel \Theta_i^{(k)})$$

$$(1 \leq i \leq K)$$

If we denote $n_{i,j} = Pre(p_i, t_j)$ and $m_{i,j} = Post(t_j, p_i)$ to each transition t_j correspond the instantaneous action

$$\tau_j = \beta_1^{n_{1,j}} \cdots \beta_K^{n_{K,j}} \, t_j \, \overline{\alpha_1}^{m_{1,j}} \cdots \overline{\alpha_K}^{m_{K,j}}$$

And the process associated with this transition, which can repeatedly perform this action, is therefore correctly modeled by the clock h_{τ_j} :

$$h_{\tau_j} \xrightarrow{\tau_j} h_{\tau_j}$$

3.2.3. From Marked Nets to Meije Expressions

Finally for each Petri net r (with the notations of (3.1)) and marking μ on r we build the MEIJE agent

$$\rho_{(r,\mu)} =_{def} ((h_{\tau_1} \mid \ldots \mid h_{\tau_N}) \parallel (\Theta_1^{(\mu_1)} \parallel \cdots \parallel \Theta_K^{(\mu_K)})) \setminus \{\alpha_i, \beta_i / 1 \leq i \leq K\}$$

Here we set an interleaving structure (which, strictly speaking, ought to be expanded in its MEIJE code) on the transitions, according to the fact that only one at a time may fire. Restriction on the signals means that tokens flow between places and transitions, so none can be lost.

Next task is to prove that the translation $\rho(r, \mu)$ of the marked net (r, μ) meets its behavioural specification : we have to show that the process which is the semantics of $\rho(r, \mu)$ is isomorphic to the transition system on markings determined by (r, μ).

lemma 4

$$\langle \rho_{(r,\mu)} \rangle \xrightarrow{c} \langle p \rangle \quad iff \; \exists \, j \; (1 \leq j \leq N) \; such \; that$$

$1 - \forall \, i \; (1 \leq i \leq K) \; \mu_i \geq n_{i,j}$ and

$2 - c = t_j$ and

$3 - p \sim \rho_{(r,\mu')} \; where \; \forall_i (1 \leq i \leq K) \; \mu'_i = \mu_i + m_{i,j} - n_{i,j}$

proof

(1) Let us assume that $\rho_{(r,\mu)} \xrightarrow{c} p$. Since for all i $(1 \leq i \leq K)$ and $k \in \mathbb{N}$ we have

$$\Theta_i^{(k)} \xrightarrow{c'} q \; implies \; c' \notin \mathbb{M} \setminus \{\alpha_i, \beta_i / 1 \leq i \leq K\}$$

we deduce (using rule $R\,3$) that

(1.1) - either (rule $R\,5$) $(h_{\tau_1} \mid \ldots \mid h_{\tau_N}) \xrightarrow{c} q$ for some q such that

$$p = (q \parallel (\Theta_1^{(\mu_1)} \parallel \cdots \parallel \Theta_K^{(\mu_K)})) \setminus \{\alpha_i, \beta_i / 1 \leq i \leq K\}$$

In this case we have, by the definition of the interleaving operator (and the behaviour of the clocks) :

$$\exists\, j\ (1 \leq j \leq N)\quad c = \tau_j \text{ and } q \sim (h_{\tau_1} | \dots | h_{\tau_N})$$

Moreover $c \in \mathbb{M} \setminus \{\alpha_i, \beta_i / 1 \leq i \leq K\}$ (rule $R\,3$) thus for all $i\,(1 \leq i \leq K)$ $n_{i,j} = 0 = m_{i,j}$ and we get the desired conclusion in this case.

(1.2) - either (rule $R\,6$)

$$(h_{\tau_1} | \dots | h_{\tau_N}) \xrightarrow{c'} q \text{ and}$$

$$(\theta_1^{(\mu_1)} \| \dots \| \theta_K^{(\mu_K)}) \xrightarrow{c''} q'$$

and $c = c'c''$ and $p = (q \| q') \setminus \{\alpha_i, \beta_i / 1 \leq i \leq K\}$. Thus $\exists\, j\ (1 \leq j \leq N)\ c' = \tau_j$ and $q \sim (h_{\tau_1} | \dots | h_{\tau_N})$

Since $c \in \mathbb{M} \setminus \{\alpha_i, \beta_i / 1 \leq i \leq K\}$ we must have

$$c'' = \overline{\beta_1}^{n_{1,j}} \dots \overline{\beta_K}^{n_{K,j}} \alpha_1^{m_{1,j}} \dots \alpha_K^{m_{K,j}} \text{ whence } c = t_j$$

Thus for all i $(1 \leq i \leq K)$ such that $n_{i,j} + m_{i,j} \neq 0$ we have :

$$\exists\, i : \theta_i^{(\mu_i)} \xrightarrow{\overline{\beta_i}^{n_{i,j}} \alpha_i^{m_{i,j}}} q_i$$

and $q' = (q_1 \| \dots \| q_K)$ if we let $q_i = \theta_i^{(\mu_i)}$ if $n_{i,j} + m_{i,j} = 0$.

By lemma 3 $n_{i,j} \leq \mu_i$ and if we let $\mu'_i = \mu_i + m_{i,j} - n_{i,j}$ we have $q_i \sim \theta_i^{(\mu'_i)}$.

Finally, $p \sim \rho_{(r,\mu')}$ as we meant to prove.

(2) Conversely, assuming that for some j $(1 \leq i \leq N)$ we have $\forall\, i$ $(1 \leq i \leq K)$ $\mu_i \geq n_{i,j}$, it is easy, by means of lemma 3 and rules defining the transition relations, to show that

$$\langle\!\langle \rho_{(r,\mu)} \rangle\!\rangle \xrightarrow{t_j} \langle\!\langle \rho_{(r,\mu')} \rangle\!\rangle$$

where $\mu'_i = \mu_i - n_{i,j} + m_{i,j}$ $\blacksquare$

Our goal is now fully achieved :

proposition 1

For any net r, marking μ and transition t :

$$(r,\mu)[t > (r,\mu')\ \textit{iff}\ \langle\!\langle \rho_{(r,\mu)} \rangle\!\rangle \xrightarrow{t} \langle\!\langle \rho_{(r,\mu')} \rangle\!\rangle \quad \blacksquare$$

We model the parallel behaviour of a Petri nets in a similar way, by defining

$$\overline{\rho}_{(r,\mu)} =_{def} (((h_{\tau_1} \| \dots \| h_{\tau_N}) \| (\theta_1^{(\mu_1)} \| \dots \| \theta_K^{(\mu_K)})) \setminus \{\alpha_i, \beta_i / 1 \leq i \leq K\}$$

and we get the analogous property :

proposition 2

For all non empty set $U = \{t_{j_1}, \dots, t_{j_n}\}$ of transitions of r

$$(r,\mu)[U > (r,\mu')\ \textit{iff}\ \langle\!\langle \overline{\rho}_{(r,\mu)} \rangle\!\rangle \xrightarrow{t_{j_1} \dots t_{j_n}} \langle\!\langle \overline{\rho}_{(r,\mu')} \rangle\!\rangle$$

This last expression seems more convenient if we think about Petri nets as modelling parallel systems. And we get a flow expression in the sense of [11] which depicts the graphical aspect of the net -taking marked places' and transitions' agents as primitive nodes.

We should also emphasize that these results are actually more accurate : they are true for the semantics of agents given by an equational congruence. We find among the equational laws we use the "laws of flow" of Milner [11].

The translation is rather straightforward : each element of the net translated first, then everything stuck together. Thus in the following example (taken from [16]) we shall anticipate our next aim, the modular construction of Petri net expressions. The so called multiple producer/multiple consumer problem (n producers, k consumers) is modeled by the Petri net given in [16] :

producers   consumers
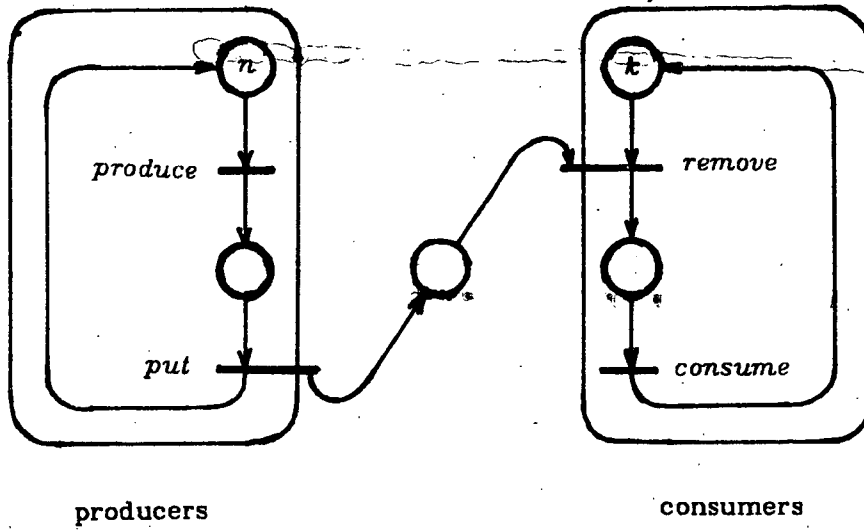
ι   Ignoring for a while the intermediate place which models the buffer, we translate the marked nets "producers" and "consumers" in :

$$Prod =_{def} ((h_{\tau_1} \parallel h_{\tau_2}) \parallel (\Theta_1^{\{n\}} \parallel \Theta_2^{\{0\}})) \setminus \{\alpha_1, \beta_1, \alpha_2, \beta_2\}$$

with $\tau_1 = \beta_1 . produce . \overline{\alpha}_2$ and $\tau_2 = \beta_2 . put . \overline{\alpha}_1$

$$Cons =_{def} ((h_{\tau'_1} \parallel h_{\tau'_2}) \parallel (\Theta_1^{\{k\}} \parallel \Theta_2^{\{0\}})) \setminus \{\alpha_1, \beta_1, \alpha_2, \beta_2\}$$

with $\tau'_1 = \beta_1 . remove . \overline{\alpha}_2$ and $\tau'_2 = \beta_2 . consume . \overline{\alpha}_1$

In both terms one recognizes the two transitions and the two places.

The next construction shows how one may build the whole system of producers and consumers commuticating through a buffer by adding a new place. The only thing to do is to say that in fact a "put" (resp. "remove") action have to put (resp. remove) a token in (from) the new place. This is done by means of renamings of actions :

$$System =_{def} (((<\overline{\alpha}_1 . put / put > Prod \parallel < \beta_1 . remove / remove > Cons) \parallel \Theta^{\{0\}}) \setminus \{\alpha_1, \beta_1\}$$

We could also have given the expression which is the translation, by $\overline{p}$, of the whole Petri net. However the structure of the problem is less visible in this last expression.

4 - THE ALGEBRA OF LABELLED NETS

So far we have only used a restricted syntax (a subcalculus) from MEIJE to express the behaviour of Petri nets. But even in this restricted syntax (inaction, action, parallel composition, restriction and recursive definition) we could write agents which do not represent Petri nets (see [2,3]). In this section we face the question of whether there is (inside MEIJE, that is by means of derived operators) a syntax which describes exactly the processes determined by Petri nets. A further would be that these new operators correspond to appealing easy structural constructions on Petri nets.

We do this for a slight extension of these nets, to be precise for *labelled nets*. The idea behind this is that the set of transitions merely is a set of (spatial) occurrences of actions. Thus a labelled and marked net is

- as in (3.1) a net $\tau = (P, T, Pre, Post)$ with $T \subseteq A$

- a marking μ *on* τ

- a labelling, that is a mapping $\lambda : T \to A$.

For a transition $t \in T$, $\lambda(t)$ is the actual action performed when t fires. Thus the transition relation is modified in an obvious way :

$$(\tau, \mu, \lambda) \, [V > (\tau, \mu', \lambda) \quad iff \quad \exists \, U \subseteq T \quad (\tau, \mu) \, [U > (\tau, \mu')$$

and V is the *multiset* $\lambda(U)$.

A labelling uniquely determines an endomorphism on $\mathbb{M}$ such that $\lambda(a) = a$ for $a \in (S \cup A) - T$ (and we do not distinguish in between both in notation). A labelled marked net is modeled, in the same sense than before, by a MEIJE agent

$$< \lambda > \overline{\rho}_{(\tau, \mu)}$$

Let L_A be the set of such nets' expressions in MEIJE. We may thus define the set of processes determined by labelled marked nets by

$$R_A = L_A \, / \sim$$

This is the universe of processes which we want to present as an algebraic calculus.

In the sequel we will note α_1 and β_1 by α and β, and a marked "typical place" will be an agent from the sequence $\{\Theta^{(k)} \, / \, k \in \mathbb{N}\}$ where $\Theta^{(k)} = \Theta_1^{(k)}$.

There exists an operation on marked nets which consists in "adding a condition", that is which adds a new marked place and extends the *Pre* and *Post* functions in a specified manner. Thus such an operation is fully determined by

- an integer $k \in \mathbb{N}$ which is the marking of the new place p

- a map $f : A \to \mathbb{N} \times \mathbb{N}$ which for each transition t gives

$$(Pre(p, t), Post(t, p)) = f(t).$$

We have seen an example of this construction in the producers/consumers problem : the new place added to the system $(Prod \, || \, Cons)$ was marked by 0 and subjected to the conditions

$$f(put) = (0, 1)$$

$$f(remove) = (1, 0)$$

(and $f(t) = (0, 0)$ for any other transition).

We do not formulate the precise definition, but merely indicate the MEIJE expression of the operation associated with such a "marked condition" (k, f) :

$$\pi_{(k, f)}(x) =_{def} (< \psi > x \, || \, \Theta^{(k)}) \setminus \alpha, \beta$$

where ψ is the morphism determined by

$$\forall\, a \in A \quad \psi(a) = \beta^n\, a\, \overline{\alpha}^m \quad \text{if } f(a) = (n,m)$$

$$(\text{and } \psi(s) = s \text{ for } s \in S)$$

Here we simply call such a derived operator a *constraint*.

In the producers/consumers example we could have built the whole system using this construction :

$$System = \pi_{(0,f)}\,(Prod \parallel Cons)$$

with the condition f given above.

proposition 3

Let N_A be the least set of MEIJE agents containing the set $H_A = \{h_a \;/\; a \in A\}$ of clocks (on A) and closed under parallel composition, labellings and constraints. Then $R_A = N_A \;/\sim$.

proof (outline)

(1) In one direction we have to prove that each labelled marked net expression $< \lambda >\overline{\rho}_{(r,\mu)}$ may be translated, up to strong equivalence, into a term of the algebra N_A. It suffices to show that $\overline{\rho}_{(r,\mu)}$ is equivalent to a term build on clocks using parallel composition and constraints. And here we simply follow the idea that, in order to build a marked net one

  - sets together some transitions,
  - then adds one by one places and appropriate arcs.

We use here a slight extension (concerning morphisms) of Milner's laws of flow (which are valid in MEIJE, see [2]) together with :

  - for all morphism φ $\quad < \varphi >h_a \sim h_{\varphi(a)}$
  - if φ is the morphism given by $\varphi(\alpha_i) = \alpha$ and

$$\varphi(\beta_i) = \beta \quad then \quad < \varphi > \Theta_i^{(k)} \sim \Theta^{(k)}$$

We sketch the proof that, with the notations of (3.3)

$$\overline{\rho}_{(r,\mu)} = [(h_{\tau_1} \parallel \cdots \parallel h_{\tau_N}) \parallel (\Theta_1^{(\mu_1)} \parallel \cdots \parallel \Theta_K^{(\mu_K)})] \setminus \{\alpha_i,\beta_i \;/\; a \leq i \leq K\}$$

we have

$$\overline{\rho}_{(r,\mu)} \sim \pi^{(K)}(...(\pi^{(1)}(h_{t_1} \parallel \cdots \parallel h_{t_N}))...)$$

where $\pi^{(i)} = \pi_{(\mu_i, f_i)}$, $f_i(t_j) = (n_{i,j}, m_{i,j})$

By induction on K :

(1.1) - If $K = 1$, we let

$$\psi(t_j) = \beta^{n_{1,j}}\, t_j\, \overline{\alpha}^{m_{1,j}}$$

for $1 \leq j \leq N$. Then,

$$(h_{\tau_1} \parallel \cdots \parallel h_{\tau_N}) \sim (< \psi >h_{t_1} \parallel \cdots \parallel < \psi >h_{t_N})$$

$$\sim < \psi >(h_{t_1} \parallel \cdots \parallel h_{t_N})$$

(1.2) - At the induction step, where $K = l+1$, we let

  - for $1 \leq j \leq N$

$$\psi(t_j) = \beta_K^{n_{K,j}}\, t_j\, \overline{\alpha}_K^{m_{K,j}} \quad \text{and}$$

$$\tau'_j = \beta_1^{n_{1,j}} \cdots \beta_l^{n_{l,j}}\, t_j\, \overline{\alpha}_1^{m_{1,j}} \cdots \overline{\alpha}_l^{m_{l,j}}$$

  (thus $\psi(\tau'_j) = \tau_j$)

$$- \overline{p}_{(r',\mu')} = [(h_{T'_1} \| \cdots \| h_{T'_N}) \| (\Theta_1^{(\mu_1)} \| \cdots \| \Theta_l^{(\mu_l)})] \setminus \{\alpha_i, \beta_i \ / \ 1 \leq i \leq l\}$$

We have

$$\overline{p}_{(r,\mu)} \sim [(< \psi > h_{T'_1} \| \cdots \| < \psi > h_{T'_N}) \| (\Theta_1^{(\mu_1)} \| \cdots \| \Theta_K^{(\mu_K)})] \setminus \{\alpha_i, \beta_i \ / \ 1 \leq i \leq K\}$$

$$\sim [(< \psi > (h_{T'_1} \| \cdots \| h_{T'_N})] \| (\Theta_1^{(\mu_1)} \| \cdots \| \Theta_K^{(\mu_K)})] \setminus \{\alpha_i, \beta_i \ / \ 1 \leq i \leq K\}$$

$$\sim (< \psi > \overline{p}_{(r',\mu')} \| \Theta_K^{(\mu_K)}) \setminus \alpha_K, \beta_K$$

$$\sim \pi^{(K)}(\overline{p}_{(r',\mu')})$$

(the last two steps might have to be detailed).

And we apply the induction hypothesis

(2) Now in the converse direction we have to prove that each clock is equivalent to a (labelled marked) net expression and then that the operations preserve this property. Here again, apart for the generators, we essentially use laws of flow.

(2.1) - Intuitively a clock h_a is nothing but the net
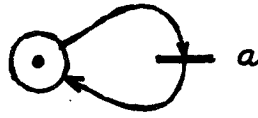
And in fact one can prove (by induction on the definition of transitions) that adding such an implicit condition has no effect :

if $f : A \to \mathbb{N} \times \mathbb{N}$ is such that, for some $a \in A$

$$f(t) = \begin{cases} (1,1) & \text{if } t = a \\ (0,0) & \text{otherwise} \end{cases}$$

then for all net expression $p \qquad \pi_{(1,f)}(p) \sim p$

(2.2) - The parallel composition of net expressions represents the operation of *juxtaposition* (and here appears the labelling). Thus the proof that it preserves the property of being equivalent to a term of L_A follows the idea that to juxtapose two labelled nets, one

- renames (by injective labellings) their sets of transitions, in order to disjoin them ;

- puts together these two new disjoint unlabelled nets, obtaining a net ;

- resets the labelling (composing the original one with the inverse renamings).

We omit the technical details.

(2.3) - Applying a constraint to a labelled net we have

$$\pi_{(k,f)}(< \lambda > \overline{p}_{(r,\mu)}) \sim < \lambda > (\pi_{(k,g)} (\overline{p}_{(r,\mu)}))$$

$$\text{where } g = f \circ \lambda', \ \lambda'(t) = \begin{cases} \lambda(t) & \text{if } t \in T \\ t & \text{otherwise} \end{cases}$$

And as in the point (1) :

$$\exists (r'',\mu'') : \pi_{(k,g)} (\overline{p}_{(r,\mu)}) \sim \overline{p}_{(r'',\mu'')}$$

(2.4) - The case of labellings is trivial since

$$< \psi > (< \varphi > p) \sim < \psi \circ \varphi > p \qquad \blacksquare$$

We may rediscover the interleaved transitions behaviour of labelled Petri nets since

if $f : A \to \mathbb{N} \times \mathbb{N}$ is such that

$$\forall a \in A \quad f(a) = (1,1) \ then \ \pi_{(1,f)}(< \lambda > \overline{p}_{(r,\mu)}) \sim < \lambda > p_{(r,\mu)}$$

A remark about the proposed syntax for labelled nets : we could improve the readability of structural descriptions by allowing some naming facilities. For instance the producers/consumers problem could be written

$$System = (\pi_{(0,f)} (Prod \parallel Cons) \text{ where}$$
$$Prod = \pi_{(n,f_1)} (\pi_{(0,f_2)} (h_{produce} \parallel h_{put})) ,$$
$$Cons = \pi_{(k,g_1)} (\pi_{(0,g_2)} (h_{remove} \parallel h_{consume})))$$

with

$$\begin{cases} f(put) = (0,1) \\ f(remove) = (1,0) \end{cases}$$
$$\begin{cases} f_1(produce) = (1,0) \\ f_1(put) = (0,1) \end{cases}$$

and so on ...

Obviously much more work remains to be done if one claims a convenient concrete syntax.

5 - RATIONAL PARALLEL PLACE MACHINES

Various extensions or restrictions of Petri nets have been proposed in order to increase either the modeling convenience or the efficiency of their analysis (see [16]). These modifications mainly affect the state space and the firing rule. We could model in Meije various kinds of modified nets, for instance nets with bounded or coloured places, or nets where places are fifo files, and so on. We focus in this section on a particular extension. Here the generalization concerns the "transition space".

So far we did not exploit much Milner's idea about the action monoid. It was an essential tool in the modeling of the internal mechanism of a Petri net, but the notion of global action hardly shows in the external behaviour. However this notion allows one to take into account the synchonism of some systems (a similar idea can be found in [1]). Let us consider the following typical example : the matter is to model the behaviour of a "box" which transmit synchronous and asynchronous inputs. This box may be depicted as :
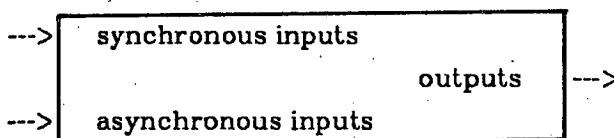
The following requirements must be fulfilled :

1/ the input/output behaviour is cyclic on a two phases pace : synchronous input followed by output. But furthermore :

2/ at any time the box may accept and store an asynchronous input

3/ a synchronous input is either full or empty. The box can output an asynchronous stored input only immediately after an empty synchronous input.

Let us name the atomic actions (or events) :

i_f : a full synchronous input
i_e : an empty synchronous input
i_a : an asynchronous input
o_f, o_e and o_a are the corresponding outputs.

To model the states of the box we use three places p_1, p_2, p_3 :

p_1 : input/output phase interlacing. Whenever this place is marked (always by at most one token) a synchronous input may and must take place ; otherwise it is an output that must occurs.

p_2 : marked iff the previous synchronous input was full.

p_3 : a counter of still handling asynchronous inputs.

We shall denote by (i,j,k) a marking which sets i,j and k tokens into p_1, p_2 and p_3 respectively. The initial state of the box is $(1,0,0)$. Then the following set Θ of transitions provides a model of the above requirements :

For all $n \in \mathbb{N}$ (the number of previous asynchronous inputs)

For $m \leq 1$ (in presence or absence of an asynchronous input)

$$(1,0,n) \xrightarrow{\;i_f \cdot i_a^m\;} (0,1,n+m) \;\Big\}$$
$$(1,0,n) \xrightarrow{\;i_e \cdot i_a^m\;} (0,0,n+m) \;\Big\}\quad \text{receiving phase}$$

$$(0,1,n) \xrightarrow{\;o_f \cdot i_a^m\;} (1,0,n+m) \qquad \text{(a full synchronous input to send)}$$

$$(0,0,n+1) \xrightarrow{\;o_a \cdot i_a^m\;} (1,0,n+m) \qquad \begin{array}{l}\text{(a stored asynchronous input is sent :}\\ \text{the previous synchronous input was empty)}\end{array}$$

$$(0,0,0) \xrightarrow{\;o_e \cdot i_a^m\;} (1,0,m) \qquad \text{(nothing to send)}$$

We shall show this to be an example of a certain kind of transition systems : those determined by Boudol's *rational parallel place machines* (cf [3]). As for Petri nets, the states space in such a machine is the set of markings on places, that is $\mathbb{N}^P$ if we call P the set of places. Let us remark that this space $\mathbb{N}^P$ is an *abelian monoid* for vectors' addition :

If $P = \{p_1, ..., p_K\}$ then the sum of two markings

$\mu = (n_1, ..., n_K)$ and $\mu' = (m_1, ..., m_K)$ is

$\mu + \mu' = (n_1 + m_1, \ldots, n_K + m_K)$.

Therefore the transition space -the cartesian product $\mathbb{N}^P \times \mathbb{M} \times \mathbb{N}^P$ where $\mathbb{M}$ is the abelian action monoid- is an abelian monoid too. We shall take a multiplicative notation :

The product of two transitions $t_1 = (\mu_1, a, \mu'_1)$ and $t_2 = (\mu_2, b, \mu'_2)$ is

$$t_1 . t_2 = (\mu_1 + \mu_2, a.b, \mu'_1 + \mu'_2)$$

(sum of markings, product of actions. This is the synchronous product of [14]).

The neutral element is

$$\mathbb{1} = ((0, ..., 0), 1, (0, ..., 0))$$

There is a well-known remarkable family of subsets of a given monoid, namely the family of *rational subsets*. Let us recall the definition (cf [7,8]) :

the family of rational subsets of a multiplicative monoid $(X, \bullet, 1)$ is the least family which includes the singletons (denoted x, for $x \in X$) and the empty set, here denoted 0, and closed under the *rational operations* which are :

- union $\quad U + V = \{x \ / \ x \in U \text{ or } x \in V\}$

- product $\quad U . V = \{x.y \ / \ x \in U \ \& \ y \in V\}$

- star, or generated submonoid $U^* = \bigcup_{n \in \mathbb{N}} U^n$

where $U^0 = \{\mathbb{1}\}$ and $U^{n+1} = U . U^n$. As usual, we shall denote $U^+ = U . U^*$. Thus the rational

subsets of X are exactly those denoted by regular expressions. When X is a product monoid, as it is the case of $\mathbb{N}^P \times \mathbb{M} \times \mathbb{N}^P$, a rational subset is often called a *rational relation*.

Definition : a Rational Parallel Place Machine is a structure

$\mathcal{Q} = (P, \mathbb{M}, \theta)$ where

- $P = \{p_1, ..., p_K\}$ is a finite set of places
- $\mathbb{M}$ is an abelian action monoid
- The set $\theta \subseteq \mathbb{N}^P \times \mathbb{M} \times \mathbb{N}^P$ of transitions is a **rational** relation.

Such a machine $\mathcal{Q}$ leads to a transition system, which is simply

$$\Xi(\mathcal{Q}) = (\mathbb{N}^P, \mathbb{M}, \theta)$$

Example : a bag is such an object produced by a one place machine. For if we consider the transitions

$\varepsilon = (1) \xrightarrow{1} (1)$ (an ineffective transition)

$t_1 = (0) \xrightarrow{a} (1)$ (putting a token, cf. 3.2.1)

$t_2 = (1) \xrightarrow{b} (0)$ (removing a token)

and if we let

$$\theta = \varepsilon^* . (t_1 + t_2)^+$$

then we have

$$\theta = \{(k) \xrightarrow{a^n b^m} (k + n - m) \,/\, n + m > 0, \, m \leq k\}$$

It is well-known (see [8]) that a rational relation such as θ is always a rational subset of a *finitely generated* submonoid. A finite set $U \subseteq \mathbb{N}^P \times \mathbb{M} \times \mathbb{N}^P$ of generating transitions such that $\theta \subseteq U^*$ may be represented by a labelled Petri net ; for a transition $t = ((n_1, ..., n_K), a, (m_1, ..., m_K))$ is an element of a labelled net such that $\lambda(t) = a$ and $Pre(p_i, t) = n_i$, $Post(t, p_j) = m_j$.

For instance the generating transitions ε, t_1, t_2 in the previous bag's example can be drawn :
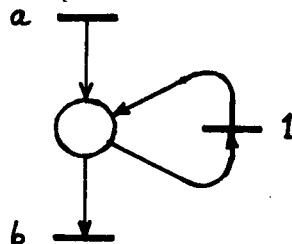
What is not shown by this representation is the firing rule ; this is expressed by the rational relation on transitions, which indicates the synchronizations. It must be seen that the behaviour of a rational machine $\mathcal{Q}$ given by $\Xi(\mathcal{Q})$ is essentially synchronous : a transition $(\mu, a, \mu') \in \theta$ can only be fired from the marking μ, where there is *exactly* the required number of tokens. Differently stated, in a transition **all** the tokens **must** be removed. However it is possible to *desynchronize* such a machine :

Let $\mathcal{Q} = (P, \mathbb{M}, \theta)$ be a rational parallel place machine. For each place $p_j \in P$ let ε_j be the "ineffective" transition which picks a token in p_j, performs 1 and puts the token back in p_j :

$$\varepsilon_j = (p_j, 1, p_j)$$

where

$$\rho_j(p) = \begin{cases} 0 & \text{if } p \neq p_j \\ 1 & \text{otherwise} \end{cases}$$

Then the *asynchronous* transition system determined by $\mathcal{Q}$ is

$$\Delta(\mathcal{Q}) = (\mathbb{N}^P, \mathbb{M}, \{\varepsilon_j / p_j \in P\}^* . \theta)$$

(obviously $\Delta(\mathcal{Q})$ is also $\Xi(\mathcal{Q}')$ for a "desynchronized" rational parallel place machine).

In this system $\Delta(\mathcal{Q})$ a transition of θ may be fired as soon as there are enough tokens in the marking, for

$$(\mu, a, \mu') \in \{\varepsilon_j / p_j \in P\}^* . \theta <=> \exists \rho, \rho' \ (\rho, a, \rho') \in \theta, \rho \leq \mu$$

$$\text{and } \rho' = \mu - \rho + \mu'$$

Now it is easily seen that the rational parallel place machines generalize Petri nets. Indeed, if one keeps with the notations of 3.1 where $r = (P, T, Pre, Post)$,

for each $t_i \in T$: let $\hat{t_i} = (\mu_i, t_i, \mu'_i)$, where $\mu_i(p) = Pre(p, t_i)$ and $\mu'_i(p) = Post(t_i, p)$.

Then the transition system determined by the net r is exactly the *asynchronous* transition system determined by the rational machine

$$(P, \mathbb{M}, (\hat{t_1} + \cdots + \hat{t_N}))$$

(where $\mathbb{M}$ is the abelian action monoid generated by T).

Let us return to our motivating example : the behaviour of the box mixing synchronous and asynchronous inputs is that of a rational machine, with an "asynchronous place" p_3. The generating transitions are

$$I_f = (1,0,0) \xrightarrow{i_f} (0,1,0)$$

$$I_e = (1,0,0) \xrightarrow{i_e} (0,0,0)$$

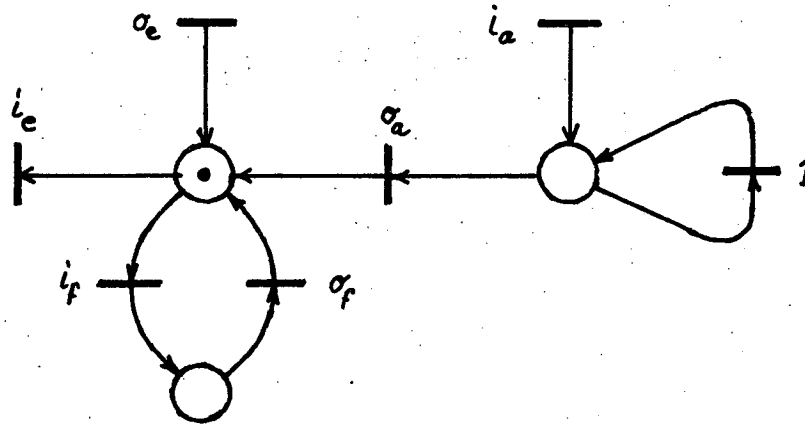$$I_a = (0,0,0) \xrightarrow{i_a} (0,0,1)$$

$$O_f = (0,1,0) \xrightarrow{o_e} (1,0,0)$$

$$O_e = (0,0,0) \xrightarrow{o_e} (1,0,0)$$

$$O_a = (0,0,1) \xrightarrow{o_a} (1,0,0)$$

$$\varepsilon = (0,0,1) \xrightarrow{1} (0,0,1)$$
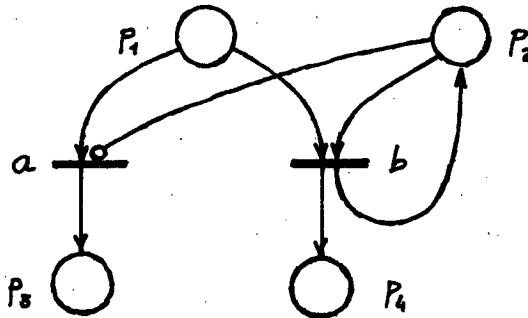
This may be drawn as :

Then the required set Θ of transition is described by a rational expression :

$$\Theta = (I_a + \mathbf{1})\,(\varepsilon^*(I_f + I_e + O_f + O_a) + O_e)$$

(we recall that $\mathbf{1} = ((0,...,0),\,1,\,(0,...,0))$ is the neutral transition).

Finally one may remark that Petri nets with inhibitor arcs are special cases of rational machines. For instance the basic "zero testing" situation (cf. [16]) is depicted by

Now let

$$t_1 = (1,0,0,0) \xrightarrow{a} (0,0,1,0) \quad \text{and}$$

$$t_2 = (1,1,0,0) \xrightarrow{b} (0,1,0,1)$$

and $\varepsilon_1,\ \varepsilon_2,\ \varepsilon_3,\ \varepsilon_4$ be the "ineffective" transitions, as before. Then the rational expression

$$(\varepsilon_1 + \varepsilon_3 + \varepsilon_4)^*\,t_1 + (\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4)^*\,t_2$$

describes the following transition system :

$$\Theta = \{(n+1,0,m,k) \xrightarrow{a} (n,0,m+1,k)\ ,(n+1,m+1,k,h) \xrightarrow{b} (n,m+1,k,h+1) \,/\,n,\,m,\,k,\,h \in \mathbb{N}\}$$

6 - CONCLUDING REMARKS

In order to build a true calculus of nets, it remains to find expressions of other interesting operations preserving (or extending) the proposed algebra and, perhaps more important, to discover algebraic properties of these operators. Each equational law may be seen as the validation of transformations or simplifications. It also remains to see to what extent a structural algebraic description can be combined with analysis techniques. Obviously we cannot derive any benefit from rational machines with regard to questions about the state space. Nevertheless rational transition systems are certainly not all effective systems.

We would like to conclude with a brief discussion on semantic options. The informal postulate underlying the strong congruence is that nothing can be said about a "state" (of a process) unless it results from the observation of the performed actions. Thus one may disagree with the fact that our semantics is right for Petri nets. For instance the k-boundedness property is not preserved. Nevertheless, this question requires more careful examination : what is the exact meaning of our strong congruence, that is how can we describe our "states" of a net (obviously all that concern transitions is preserved) ? Moreover we could have proposed a stronger equivalence : we may find a set E of equational laws (among them the laws of flow, together with laws concerning the dynamics of generators and primitive operators) so that we may present, with the same proofs, the algebra of labelled nets as $L_A /=_E$ (where $=_E$ is the least congruence containing the instances of the equalities of E). Here again, to what extent do we get a right semantics for Petri nets ? Certainly such an equational congruence preserves some structural aspects. On the other hand, observational equivalences or bisimulations (see [12,14,3]), which allow one to forget about some transitions, may be better suited for some purposes. Thus the equivalence appears as a parameter of an algebraic calculus of processes, one which may be chosen according to the intended semantics.

We leave all these semantical questions for future research, which may bring forth a new point of view on Petri nets and modelisation.

REFERENCES

[1] C. André, P. Armand & F. Boeri : "Synchronic relations and applications in parallel computations", Digital Process 5 (1979), 99-113

[2] D. Austry, G. Boudol : "Algèbre de processus et synchronisation", Theoret. Comput. Sci. 30 (1984)

[3] G. Boudol : "Notes on algebraic calculi of process", Advanced Course on Logics and Models for Verification and Specification of Concurrent Systems, La Colle-sur-Loup (1984), INRIA Res. Rep. to appear

[4] W. Brauer (editor) : "Advanced course on general net theory of processes and systems", Hamburg 1979. Lecture Notes in Comput. Sci. 84 (Springer, Berlin, 1980)

[5] G.W. Brams : "Réseaux de Petri : théorie et pratique", (Masson, Paris, 1983)

[6] R.H. Campbell, P.E. Lauer : "Formal semantics for a class of high-level primitives for coordinating concurrent processes". Acta Informatica 5 (1975), 247-332

[7] S. Eilenberg & M.P. Schutzenberger : "Rational sets in Commutative monoids", Journal of Algebra 13 (1969), 173-191

[8] S. Eilenberg : "Automata, Languages and Machines" Vol.A, Academic Press (1974)

[9] U. Goltz, A. Mycroft : "On the relationship of CCS and Petri Nets", ICALP 84, Lecture Notes in Comput. Sci. 172 (Springer, Berlin, 1984), 196-208

[10] V.E. Kotov : "An algebra for parallelism based on Petri nets", MFCS 79, Lecture Notes in Comput. Sci. 84 (Springer, Berlin, 1979), 39-55

[11] R. Milner : "Flowgraphs and flow algebras", JACM 26 (1979), 794-818

[12] R. Milner : "A calculus of communicating systems", Lecture Notes in Comput. Sci., 92 (Springer, Berlin, 1980)

[13] R. Milner : "On relating synchrony and asynchrony", Tech. Rept. CSR-75-80, Comput. Sci. Dept., Edinburgh Univ. 1980

[14] R. Milner : "Calculi for synchrony and asynchrony", Theoret. Comput. Sci. 25 (1983), 267-310

[15] J.L. Peterson : "Petri nets", Comput. Surveys 9 (1977), 223-252

[16] J.L. Peterson : "Petri net theory and the modeling of systems". Prentice Hall (1981)

[17] G. Rozenberg, R. Verraedt : "Subset languages of Petri nets", Part I, Theoret. Comput. Sci. 26 (1983), 301-326

[18] R. de Simone : "Higher-Level synchronizing devices in Meije-SCCS", INRIA Res. Rep. n° 360 (1985). To appear in TCS