TOWARDS A GENERAL AND POWERFUL COMPUTATION OF FLOWS FOR PARAMETRIZED COLOURED NETS

S. HADDAD **J.M. COUVREUR**
Universite Paris VI -C.N.R.S. MASI et C3 4 Place Jussieu, 75252 Paris Cedex 05

Abstract

This paper extends the computation of flows for parametrized regular nets to new categories of parametrized coloured nets. These kinds of nets can model associations between objects of the same class, bounded files, virtual rings, counters gestion. The algorithms presented here fulfill the main criteria of a good computation for high level nets: the intensive representation of flows, the independance from cardinality domains and the computation of a generative family of flows. New ideas have enabled to obtain these results such as transformation of coloured domains and computation in a quotient polynomial ring. Various significative examples are given with the interpretation of the computed flows. We are then close to a general algorithm for a very large category of coloured nets and certainly large enough to model a wide range of applications.

INTRODUCTION

In many models of parallelism, there are only two methods of validation: analysis of the reachability graph or verification of properties from axioms and inference rules. The first method cannot be used on real systems since the size of the graph (even when finite) is generally too big while the drawbacks of the second method are:
-a property may be true even if it cannot be proved in the formal system,
-a property may be verified but not computed.

An important advantage of Petri nets [Bra83] over the other models of parallelism is the existence of alternative and constructive methods of proofs such as the flows computation [Mem83] or the reductions [Ber 83]. So, as abbreviations of Petri nets -coloured nets [Jen82] and predicate transition nets [Gen81], [Lau85] - were introduced in order to model complex systems, many researchers have tried to extend the main results of the Petri net theory and in particular the flows calculus.

In contrast to the other methods [Jen81], [Vau84], [All84] and [Sil85] which were only partial solutions to the general problem we have focused on finding a general solution for some categories of colored nets. Then we have pointed out in [Had87b] hidden ideas not only about the flows computation but more generally about the specification and the validation by high level nets. We recall in this paper these ideas.

First of all the definition of coloured nets which gives no restriction about the colour domains and the colour functions is too large to enable any other proof method than the unfolding of the coloured net into an ordinary Petri net. However, the modelling of applications by means of coloured nets uses only particular colour domains and colour functions (identity, sum, projection, successor,...). Each of these domains and functions verifies regular properties. Thus our first idea is to define a category of coloured nets large enough to model realistic problems and regular enough to obtain theoretical results about it.

At first such a category may appear as a new subclass of coloured nets but it is much more than that. Indeed all the parallel systems have parameters (number of process or ressources) and in a modelling by an ordinary coloured net, these parameters must be fixed while in a regular net they remain parameters of the net. Therefore our second main idea is that the parametrization of the net enables its global validation if one builds new proof algorithms which take these parameters into account.

Based on these two general ideas, we have defined and parametrized a regular category of coloured nets and we have studied this model obtaining three important proof methods (symbolic accessibility graph, flows computation and reduction theory). More specifically the principles about the flows computation [Had86] , [Had87a] were the following:
-study the algebraic structure of the flows space,
-starting from this study, decompose this space into subspaces of particular flows,
-build nets deduced from the original net, on which the particular flows may be computed,
-perform computations in the polynomial ring in order to handle the parametrization.

Our previous algorithm completely fulfilled the criteria of a good computation: intensive representation of flows ("high-level" flows), computation independant from the cardinality of colour domains, and computation of generative family of flows. But in our opinion, there are two important limitations of regular nets. The first one is that objects of same class cannot be associated and the second one is the lack of successor functions.

So the main purpose of this paper is therefore to give algorithms to compute flows on these two kinds of nets - which we will call associative nets and ordered nets -but we also want to emphasize that:
- These new models are **not arbitrary models** since many of the examples presented here were introduced by other authors.
- Our new algorithms are **not a simple extension** of our first one since some principles are completely new such as the transformation of colour domains for associative nets or the computing in quotient polynomial rings for ordered nets.
- These algorithms suggest that there is **a strong link between the type of flows and the type of colour functions** used in a net.

Our two new algorithms presented here and the previous one provide **systematic methods to compute a generative family of flows** on more general models taking into account **the parametrization.** In order to finish this work and to obtain a general algorithm, we have yet to unify and to slighty increase the model (by introducing the constant functions) and to synthetize the different algorithms presented so far. At this point the modellers will have at their disposal a powerful tool to validate their applications.

Section 1 recalls classical definitions for coloured nets and the specific definitions of associative nets and ordered nets. Section 2 presents the flow calculus on associative nets and applies this calculus to well known examples of a telephone system [Jen 82] and a distributed data base [Jen 81]. Section 3 the theory of flows computation on ordered nets using polynomial matrix algebra and applies it to examples of dining philosophers and counters synchronization [Gen 82].

N.B.: For space purposes, we only give in this paper the sketches of some proofs: the **complete proofs** are given for associative nets in [Had88] and for ordered nets in [Cou88].

General notations

-N is the set of non negative integers, **Z** is the set of integers and **Q** is the set of rational numbers
-M.N, where M and N are matrices, denotes the matrices product (this notation includes the product of a vector by a matrix)
-M^t, where M is a matrix n x p, denotes the matrix p x n such that: $M^t_{i,j} = M_{j,i}$
-Let U be a finite set, then the set of functions from U to **N** is denoted Bag(U).
-An item a of Bag(U) is noted $\sum a_u.u$ where the summation is over $u \in U$.
-A partial order on Bag(U) is defined by : $a \leq b$ if and only if $\forall u \in U$, $au \leq bu$
-The sum of two items of Bag(U) is defined by $a+b = \sum (a_u + b_u).u$
-The difference between two items $a \geq b$ of Bag(U) is defined by : $a-b = \sum (a_u - b_u).u$

1COLOURED NETS

1.1 General definitions

We recall hereunder the definitions of a coloured net, the firing rule in a coloured net, the incidence matrix and the flows of a coloured net. Since the nets that we present in this paper are subclasses of coloured nets, the definitions of these nets are obtained by a restriction of the first definition and the other definitions can be applied without any transformation.

Definition 1 A coloured net $R = <P,T,C,I^+,I^-,M>$ is defined by:

- P the set of places
- T the set of transitions
- C the colour function from $P \cup T$ to Ω , where Ω is the set of finite and not empty sets.

An item of C(s) is called a colour of sand C(s) is called the colour set of s.

- I^+ (I^-) is the forward (backward) incidence matrix of $P \times T$, where $I^+(p,t)$ is a function from $C(t) \times C(p)$ to **N** (i.e. a function from C(t) to Bag(C(p)))
- M the initial marking of the net is a vector of P, where M(p) is a function from C(p) **to N** (i.e. an item of Bag(C(p)))

Notation

We note $I^+, I^-(p,t,c_t)$, where c_t belongs to C(t), the corresponding item of Bag(C(p)).

Definition 2 The firing rule is defined by :

- A transition t is enabled for a marking M and a colour $c_t \in C(t)$ if and only if :
  $\forall p \in P, M(p) \geq I^-(p,t,c_t)$
- The firing of t for a marking M and a colour $c_t \in C(t)$ gives a new marking M' defined by : $\forall p \in P, M'(p) = M(p) - I^-(p,t,c_t) + I^+(p,t,c_t)$

Definition 3 The incidence matrix I of a coloured net is defined by :

- $I = I^+ - I^-$, then I(p,t) is a function from $C(p) \times C(t)$ to **Z**
- I can be also wieved like a matrix of $\cup (p,c) \times \cup (t,c')$ of integers where the first union is over $p \in P$ and $c \in C(p)$ and the second union is over $t \in T$ and $c' \in C(t)$, by the simple transformation: $I((p,c),(t,c')) = I(p,t)(c',c)$

Definition 4 The vector space of coloured places E of a coloured net is defined as the rational vector space on $\cup (p,c)$ where the union is over $p \in P$ and $c \in C(p)$.
Then a vector of E can be written $v = (v_{p,c})$ with $v_{p,c} \in \mathbf{Q}$ or $v = \sum v_{p,c} .(p,c)$

Definition 5 The set of flows E' of a coloured net is a subset of E defined by :
$E' = \{v \in E \mid I^t.v=0\}$

In fact, because of the linearity of the product of matrices, E' is a vector subspace of E.

In practice, there are some kinds of colour domains and colour functions which are frequently used. We call these domains standard domains and these functions standard functions and we will define them in the two next paragraphs. Associative nets and ordered nets will then be defined with the help of these domains and functions.

1.2 Standard colour domains

The objects of a parallel system can be decomposed in classes related to their nature. For instance there are two (or more) classes of users of a file: the readers and the writers. In a parallel program one can distinguish the class of the processes running concurrently and the class of the ressources shared by these processes, etc... Then the classes of objects modelled by the net will be basic domains and the coloured domains of the net will be products of these basic domains.

Definition Let $C_1,...,C_n$ be basic domains of a coloured net, then a colour domain is standard if there exists p the arity of C and u an application from {1 ...p} into {1.. .n} such that:

-If $p \neq 0$ then $C = C_{u(1)} \times ... \times C_{u(p)}$

-if p = 0 then C = {e} where a token coloured by e denotes an ordinary token.

Without loss of generality we will suppose that the basic domains are circularly ordered by some enumeration of their items.

1.3 Standard colour functions

We define now the standard functions. These functions are built from the following well-known functions: the projections which select a component of an item, the constant functions which define a fixed number of ordinary tokens, the sum functions which select all the items of a basic domain and the successor functions which select some successor of an item.

Definition 1 Let $C = C_{u(1)} \times ... \times C_{u(p)}$ be a standard domain, then X_i the i^{th} projection of C is defined by:

$X_j : C \times C_{u(j)} \rightarrow \mathbf{N}$ with $X_i(<c_1,\ldots,c_p>,c) = Eq(c_j,c)$

where $Eq(x,y) =_{def}$ If x=y then 1 else 0

Definition 2 Let $C = C_{u(1)} \times ... \times C_{u(p)}$ be a standard domain, then $\boldsymbol{\lambda}$ (where λ is an integer) the constant function sum related to C is defined by:

$\boldsymbol{\lambda}: C \times \{e\} \rightarrow \mathbf{N}$ with $\lambda (<c_1,\ldots,c_p>,e) = \lambda$

Definition 3 Let $C = C_{u(1)} \times ... \times C_{u(p)}$ be a standard domain, then S_i the constant function sum related to C is defined by:

$S_i: C \times C_i \rightarrow \mathbf{N}$ with $S_j(<c_1,\ldots,c_p>,c) = 1$

Definition 4 Let $C = C_{u(1)} \times ... \times C_{u(p)}$ be a standard domain, then $X_i \oplus n$, the ith successor function with order n related to C is defined by:

$X_i \oplus n : C \times C_i \rightarrow \mathbf{N}$

with $X_i \oplus n (<c_1,\ldots,c_p>,c) =$ **If** c is the nth successor of c_i **then** 1 **else** 0

Remark

A consequence of these definitions is that the function symbol X_i and S_i have no absolute meanings in a net, but depend on the context (i.e. the colour domain of the adjacent transition).

All these functions can be rewritten in the usual bag notation:

$X_i(<c_1,\ldots,c_p>) = c_i$, $\lambda(<c_1,\ldots,c_p>) = e$, $S_i(<c_1,\ldots,c_p>) = \sum c$ where c ranges over C_i and

$X_i \oplus n (<c_1,\ldots,c_p>) = c_i \oplus n$ where $c_i \oplus n$ denotes the n^{th} successor of c_i in $C_{u(i)}$

As a colour function can be wieved as a matrix, new colour functions can be built by the standart operations on matrices such as the addition, the multiplication by a scalar, the composition and a non standard "product":

Definition 5 Let f be a coloured function defined on $C \times C'$ and g be a coloured function defined on $C \times C''$, then f+g the sum of f and g is defined by :

$f+g : C \times C' \rightarrow \mathbf{N}$

with $f+g(c,c') = f(c,c')+g(c,c')$

Definition 6 Let f be a coloured function defined on $C \times C'$ and μ be an integer, then $\mu.$ f the multiplication of f by μ is defined by :

$\mu. f : C \times C' \rightarrow \mathbf{N}$

with $(\mu.f)(c,c') = \mu.f(c,c')$

Definition 7 Let f be a coloured function defined on $C \times C'$ and g be a coloured function defined on $C' \times C''$, then fog the composition of f and g is defined by :

$fog: C \times C'' \rightarrow \mathbf{N}$

with $fog(c,c'') = \sum f(c,c').g(c',c'')$ where c' ranges over C'

Definition 8 Let f be a coloured function defined on $C \times C'$ and g be a coloured function defined on $C \times C''$, then <f,g> the product of f and g is defined by :

$<f,g> : C \times (C' \times C'') \rightarrow \mathbf{N}$

with $<f,g>(c,<c',c">) = f(c,c').g(c,c'')$

Remark This product is associative and thus the product of $f_1,...,f_n$ will be denoted $< f_1,...,f_n >$, it also verifies: $< f_1,...,\mu.f_i,...,f_n > = \mu. < f_1,...,f_i,...,f_n >$ and $< f_1,\ldots,f_i+g_i,...,f_n > = < f_1,...,f_i,...,f_n > + < f_1,...,g_i,...,f_n >$

1.4 Associative nets

1.4.1 Presentation and definition

Definition 1 An associative coloured net R is a coloured net where the colour domains and the colour functions fulfill the following conditions:

- There is a single basic domain D
- The colour domain of any transition is $D \times D$
- The colour domain of a place is either {e} either D or $D \times D$

A colour function of R is built by scalar multiplications, additions and products of projections, constant functions and sum functions

Notations In the following,

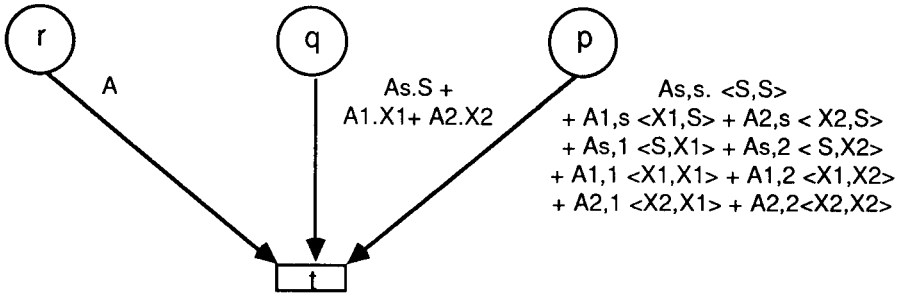
- The cardinality of the class D will be denoted by n
- The sum function of D will simply be denoted by S
- P will denote a place with type(p) = 2, i.e. with $C(p) = D \times D$
- q will denote a place with type(q) = 1, i.e. with C(q) = D
- r will denote a place with type(r) = 0, i.e. with C(r) = {e}

We need another restrictive definition which will be useful in the computation of flows.

Definition 2 A normalized associative net is an associative net with no occurence of X1 ,X2 and S appearing simultenaously around a transition. If X1 and X2 appear around a transition t then the type of t is 2 else the type of t is 1 .

Remark In a normalized net, if X2 appears around a transition t with type(t) = 1 we can rename it by X1 (since in this case X1 does not appear). We then assume without loss of generality that in a transition t with type(t) = 1, X2 never appears. Consequently, the domain of a transition of type 1 may be reduced to D.

The following figure summarizes all the eventualities of an associative net.



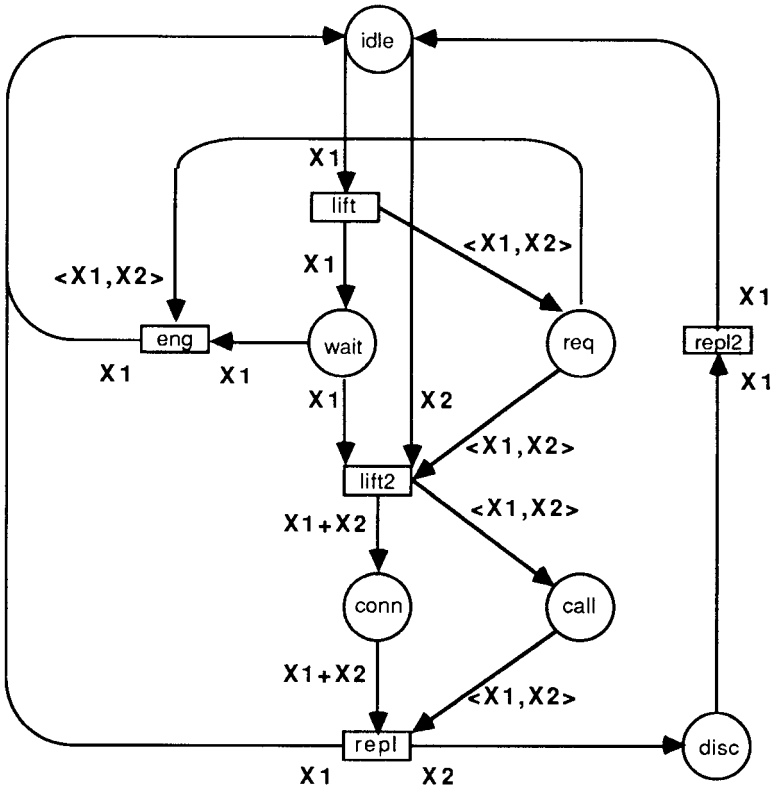
1.4.2 Two examples of associative nets

We present two examples of associative nets which have already been developped by other autors. We have made this choice in order to show that associative nets are not an arbitrary model. (Notice that the two nets are normalized associative nets)

A simplified version of a telephone system

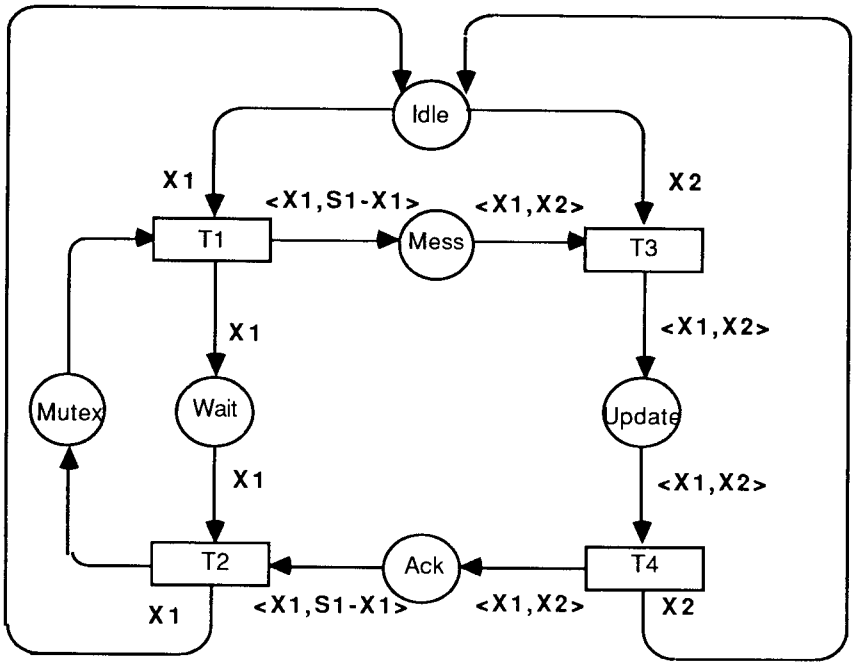
In this model the user (phone number) who has taken the initiative of the call has also the initiative of the disconnection. From the original net we have suppressed intermediate states (like the time between lifting and dialing) and some transitions like replacing the receivers. This transformation does not essentially change the protocol and the simplified net is enough complex to enable the computing of interesting flows. Notice that the original model was already an associative net and then we could also apply the algorithm presented below.

In the net, D the objects class is the phone numbers set. A user lifts and dials a number to make a call. If the number is engaged, then the user replaces. In the other case, the second user may then lift the receiver (2) turning the request into a call. The communication is ended when the first user replaces the receiver and then the second user has no other alternative but to also replace(2) his receiver. Initially there is a token of each number phone in Idle.



A database management with multiple copies [Jen81] , [Vau84] , [Sil85]

In order to modify the database, a site must get a grant modelled by a unique token in Mutex. Then the site sends messages with the updates to all the other sites, and it releases the token after having received all the acknowledgments. The owner's identity of the grant is kept in all the messages and their acknowledgments. D is the class of sites. Initially there is a unique token in Mutex and a token of each site in Idle.



1.5.1 Presentation and definition

Definition 1 An ordered coloured net R is a coloured net where the colour domains and the colour functions fulfill the following conditions:

- The set of basic domains is $\{C_1,...,C_q\}$
- The colour domain of any transition and any place is $\Omega=C_1\times...\times C_q$
- A colour function of R is built by scalar multiplications, additions and products of successor functions

Notations In the following,

- N_i will denote the cardinality of the class C_i
- g_i will denote $<X_1,...,X_i\oplus 1,...,X_q>$

Remark The general form of a colour function of an ordered net is :

$$\sum_{k1,...,kq \in \mathbf{N}} a(k_1,...,k_q) <X_1 \oplus k_1,...,X_q \oplus k_q >$$

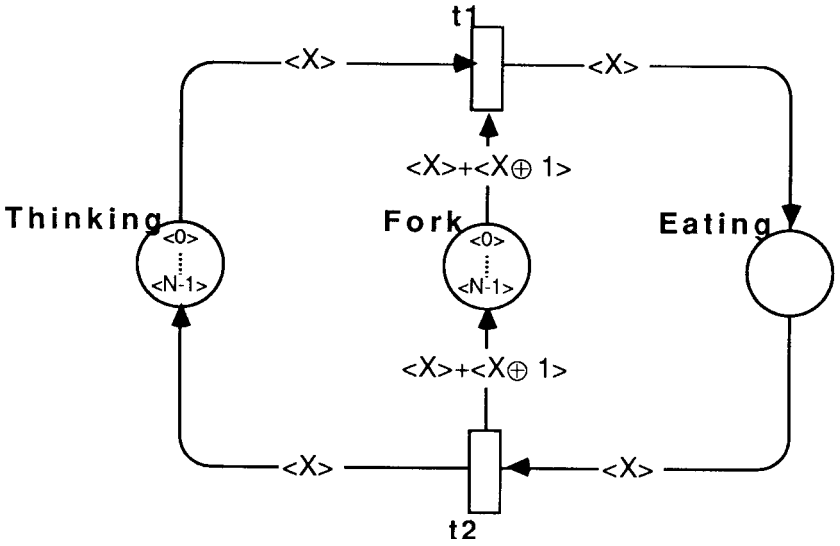
where $a(k_1,...,k_q)$ are elements of **N**, almost all of which are equal to 0.

1.5.2 Two examples of ordered nets

As for associative nets, we present two examples of ordered nets which have already been developped by other autors.

The philosophers problem

In the classical N philosophers problem, we have N philosophers and N forks, one fork between each philosopher. In order to eat, a philosopher must take the two closest forks. The single class of this model is the numbering of forks and philosophers {0,...,N-1}. The two places *Thinking* and *Eating* define the states of the philosophers and the place *Fork* contains the tokens corresponding to the free forks. For the philosopher <x> to be able to eat, both his fork <x> and the fork of his successor <x⊕1> must be free.

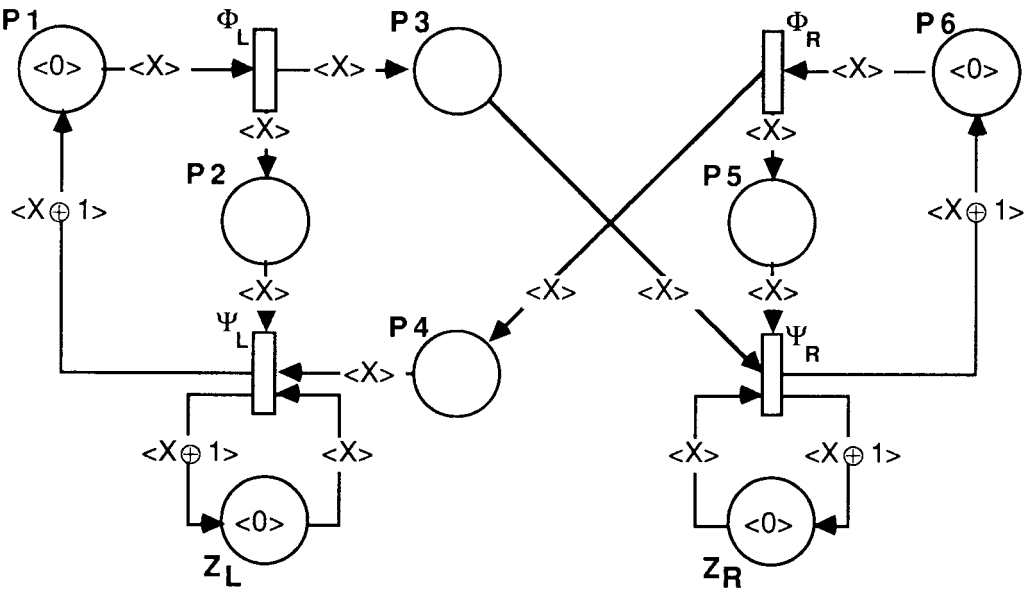


Two synchronized counters

This example is due to H.Eckert and R. Prinoch who used it to verify certain facts in the field of communication protocols. A study of this model has been improved in [Gen82]. Two partners, L and R, are sending each other the positions of their local counters Z_L and Z_R, resp., by messages. The initial position of both counters is 0, their capacity is assumed to be N. The only class C is the position of counters {0,...,N-1}. The messages (on the "channels" P3 and P4), the position of counters Z_L and Z_R and the situations of both partners are modeled by places of colored domain C.

The initial marking, $M_0(Z_L)= M_0(P1)= M_0(Z_R)= M_0(P6)= <0>$, $M_0(Ps)= 0$ for s=2,3,4,5, indicates that the position of Z_L and Z_R is 0 and L and R are ready to send the message <0> to each other.

A partner say L, is permitted to increase its counter Ψ_L by firing of ZL if P2, P4 and ZL are carring equal token.
We want to show that the absolute difference of counter positions is one at the most.



2 COMPUTATION OF FLOWS OF AN ASSOCIATIVE NET

Our computation of a generative family of flows may be decomposed in five steps. For each step we consider a family of partial flows (related to a subset of transitions) each flow having "a colour domain". Initially this family is the family of places (related to the empty set) with their colour domains. Then for each step one can imagine a coloured net the transitions of which are obtained from the transitions of the initial net not yet eliminated and the places of which are the partial flows. The aim of step 1,2 and 4 is to simplify the structure of colour functions in such a way that one can apply a symbolic elimination (step 3 and 5) which builds at each time a generative family (related to the subset of the transitions eliminated).
We now summarize the five steps:

First step Transforming an associative net into a normalized associative net the flows of which are isomorphic to the flows of the original net.
Second step Transforming the net in another net by decomposition of the colour domain D^2 in $D_+=\{(x,y) / x < y\}$ and D
Third step Computing partial flows by elimination of the transitions of type 2. (This step is an iterative process)
Fourth step Transforming the partial flows computed in the preceeding step so that the domain of partial (but not global) flows is D. (This step only may possibly generate linear combinations between the flows and if not, we obtain a basis of flows)
Fifth step Computing definitive flows by elimination of the transitions of type 1. (This step is an iterative process)

2.1 Normalization of an associative net

In this step we are going to split each transition in four transitions, each place p with type(p) = 2 in four places and each place q with type(q) = 1 in two places so that the new net is a normalized net and such that the flows of the new net are isomorphic to the flows of the initial net.
Let xo be a colour of D and D* be D\{xo}. D* will be the class of the new net. Now we define the transformation:

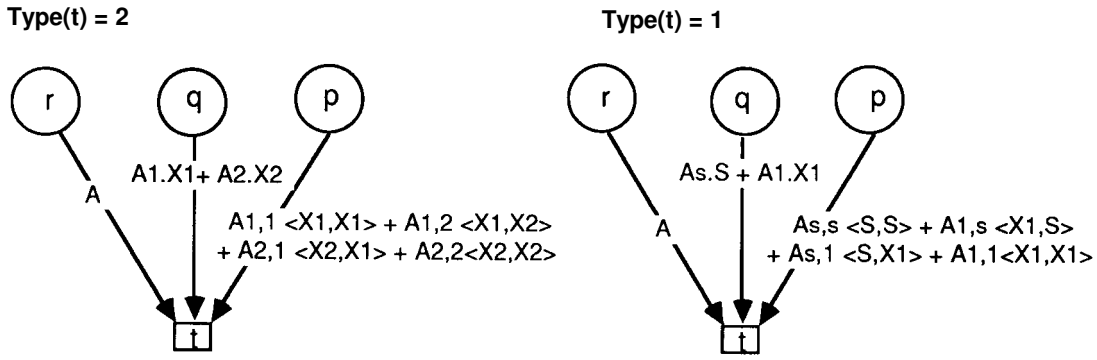
- Each transition t gives in the transformed net t', t_1 , t_2 and t_e. We have type(t') = 2 and type(t_1) = type(t_2) = type(t_e) = 1. The underlying isomorphism between the old transition and the new ones is the following:
  $\forall$ x1,x2 $\in$ D* ,
  $t'(x1,x2) \equiv t(x1,x2)$ $\quad$ $t_1(x1,x2) \equiv t(x1,xo)$
  $t_2(x1,x2) \equiv t(xo,x2)$ $\quad$ $t_e(x1,x2) \equiv t(xo,xo)$
- Each place p gives in the transformed net p', p_1, p_2 and p_e. We have type(p') = 2 and type(p_1) = type(p_2) = type(p_e) = 1. The underlying isomorphism between the old place and the new ones is the following:
  $\forall$ x1,x2 $\in$ D* , (all the summations are over D)
  $p'(x1,x2) \equiv p(x1,x2)-p(x1,xo)-p(xo,x2) + p(xo,xo)$
  $p_1(x1) \equiv \sum_y [p(x1,y) -p(xo,y)]$ $\quad$ $p_2(x1) \equiv \sum_x [p(x,x1) -p(x,xo)]$
  $p_e(e) \equiv \sum_x \sum_y p(x,y)$
- Each place q gives in the transformed net q' and q_e. We have type(q') = type(q_e) = 1. The underlying isomorphism between the old place and the new ones is the following:
  $\forall$ x1,x2 $\in$ D* , (all the summations are over D)
  $q'(x1) \equiv q(x1)-q(xo)$ $\quad$ $q_e(e) \equiv \sum_x q(x)$

Now we only have to give the transformation of colour functions. The tables of transformations are given in [Had88]. These transformations can easily be deduced as shown by the following example: Let I(p,t) = <X1,S>, then if we want to find in the new net I'(p_1,t_2) , we apply:
$<X1,S> ((xo,x2) , \sum_y [(x1 ,y) -(xo,y)]) =$
$\sum_y <X1,S> ((xo,x2) , (x1,y)) - \sum_y <X1,S> ((xo,x2) , (xo,y)) =$
$\sum_y 0 - \sum_y 1 = -N$ (0 since $xo \neq x1$)
Then I'(p_1, t_2) (x2,x1) = -n that is to say I'(p_1, t_2) = -n.S
Then once the flows family is computed on the transformed net, one applies the isomorphism on places to find the flows family of the original net. Notice that these transformations are parametrized by n, the cardinality of D.
Justification of this step The flows problem can be reformulated like that:

- Let E be the rational vector space $\mathbf{Q}^P$ where P is the set of coloured places
- Let E* be the dual of E (i.e. the vector space of linear forms on E)
- Then a transition (t,c) can be identified to an item of E* by: <<p,c'),(t,c)>> = I(p,t)(c,c')
- Let G be the subspace of E* generated by the transitions
- Then F the flows space of E is defined as the orthogonal of G :
  $f \in F \Leftrightarrow \forall g \in G , <<f ,g>> = 0$

In this step, we have only substituted a new base for E and a new generative family of G (i.e the places and the transitions of the normalized net) and have deduced from the isomorphisms the new incidence matrice. The interest of this step is that the valuations of the arcs are simplified while the inherent drawback is a limited unfolding which is independant from n. ◊◊◊

We present now the colour functions of a normalized associative net:



2.2 Decomposition of the colour domain D^2

Intuitively, on can see that flows on items <x ,x> and flows on items <x,y> ($x \neq y$) of the domain D^2 have not the same interpretation. More formally, this step decomposes the colour domain D^2 in such a way that the two kinds of items do not appear in the same colour domain. One notes $D^+ = \{<x,y> \mid x>y\}$ (This notation supposes an arbitrary order on D, see 1.2)

Each transition t with type(t) = 2, gives in the transformed net a transition $t_=$ with $C(t_=) = D$ and two transitions t_+ and t_- with $C(t_+) = C(t_-) = D^+$. The underlying isomorphism between the old transition and the new ones is the following:

$\forall x \in D, \forall <x,y> \in D^+$,

$t_=(x) \equiv t(x,x)$ $\qquad t_+(x,y) \equiv t(x,y) + t(y,x)$ $\qquad t_-(x,y) \equiv t(x,y) - t(y,x)$

Each place p with type(p) = 2, gives in the transformed net a place $p_=$ with $C(p_=) = D$ and two places p_+ and p_- with $C(p_+) = C(p_-) = D^+$. The underlying isomorphism between the old place and the new ones is the following:

$\forall x \in D, \forall <x,y> \in D^+$,

$p_=(x) \equiv p(x,x)$ $\qquad p_+(x,y) \equiv p(x,y) + p(y,x)$ $\qquad p_-(x,y) \equiv p(x,y) - p(y,x)$

Now we only have to give the transformation of colour functions. The tables of transformations are given in [Had88]. These transformations can easily be deduced as shown by the following example:

Let $I(p,t) = <X1,X1>$, then if we want to find in the new net $I'(p_=,t_=)$, we apply:

$<X1,X1> ((x1,x2) - (x2,x1) , (x,x)) =$

$<X1 ,X1 > ((x1 ,x2) ,(x ,x)) - <X1 ,X1 > ((x2,x1) , (x ,x)) =$

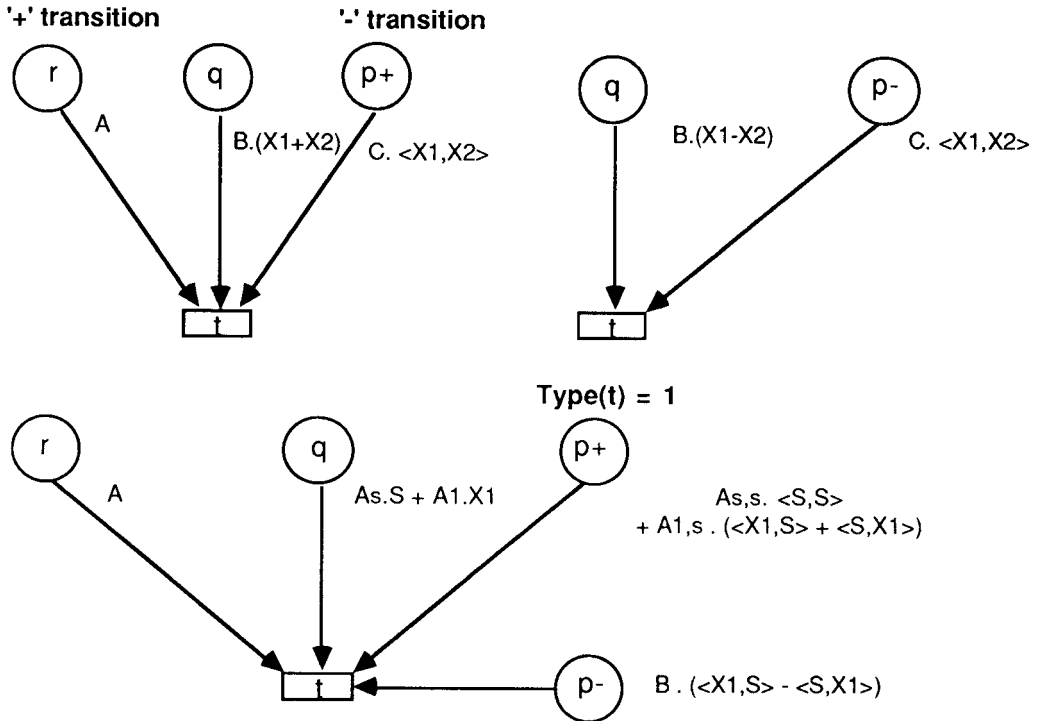
$Eq(x1 ,x).Eq(x1 ,x) - Eq(x2,x).Eq(x2,x)$

Then $I'(p_=,t_=) (<x1,x2>,x) = Eq(x1,x) - Eq(x2,x)$ that is to say $I'(p_=,t_=) = X1 - X2$

Then once the flows family is computed on the transformed net, one applies the isomorphism on places to find the flows family of the original net. Notice that these transformations are independant of n, the cardinality of D. We have now three kinds of transitions the '+' transitions, the '-' transitions and the transitions of type 1 .

Justification of this step As in the preceeding step, we have only substituted new bases for E and G (i.e the places and the transitions of the normalized net) and have deduced from the isomorphisms the new incidence matrice. The interest of this step is that the valuations of the arcs are simplified while the inherent drawback is a limited unfolding which is independant from n. ◊◊◊

We present now the colour functions of the transformed net:



2.3 Elimination of the transitions with domain D^+

We are now going to eliminate the '+' transitions and the '-' transitions. Let p^+ be a partial flow with type(p^+)='+' , let p^- be a partial flow with type(p^-)='-', let q be a partial flow with type(q)=1 and let r be a partial flow with type(r)=0. Finally let F = $\{p^+ ,p'^+ ,...\} \cup \{p^-,p'^- ,...\} \cup \{q,q',... \} \cup \{r,r',...\}$ be the generative family computed at each time. First, let t be a '+' transition and let the incidence matrix column of t be : $I(p^+,t) = C_p$.<X1 ,X2> and $I(q,t) = B_q$.(X1 + X2) and $I(r,t) = A_r$ The new generative family is built as follows:

a) All partial flows p^- are kept

b) **If** there exists a partial flow p^+ so that $C_p \neq 0$ **then**
for each partial flow p'^+ we build a new partial flow: $(Cp . p'^+) - (Cp' . p^+)$
else all partial flows p+ are kept in the new family **end if**

c) **If** there exists a partial flow q such that $B_q \neq 0$ **then**
for each partial flow q' we build a new partial flow: $(B_q .q') - (B_{q'} .q)$
else all partial flows q are kept in the new family **end if**

d) **If** there exists a partial flow r such that $Ar \neq 0$ **then**
for each partial flow r' we built a new partial flow: $(A_r .r') - (A_r, .r)$
else all partial flows r are kept in the new family **end if**

e) **If** there exists a partial flow p^+ such that $C_p \neq 0$ **then**
If there exists a partial flow q such that $B_q \neq 0$ **then**
a new flow the type of which is 1, is built: $(C_p .q) - (B_q . \mathbb{p}^+)$
where $\mathbb{p}^+ (x) \equiv \sum_y p^+(x,y) + \sum_y p^+(y,x)$ (where in the first sum the summation is over $\{y \mid y < x\}$
and the second one is over $\{y \mid y > x\}$) **endif**
If there exists a partial flow r such that $A_r \neq 0$ **then**
a new flow the type of which is 0, is built: $(C_p .r) - (A_r .\mathbf{p}^+)$
where $\mathbf{p}^+(e) \equiv \sum_{<x,y>} p(x,y)$ (where the summation is over D^+) **end if**
else
If there exists a partial flow q such that $B_q \neq 0$ and there exists a flow r such that $A_r \neq 0$ **then**
a new flow the type of which is 0, is built: $(2 .B_q .r) - (A_r .\mathbf{q})$
where $\mathbf{q}(e) \equiv \sum_x q(x)$ (where the summation is over D)
endif
endif

Now, let t be a '-' transition and let the incidence matrix column of t be :

$I(p^-,t) = C_p$.<X1 ,X2> and $I(q,t) = B_q$.(X1 -X2)

The new generative family is built as follows:

a) All partial flows p^+ and r are kept

b) **If** there exists a partial flow p^- such that $C_p \neq 0$ **then**
for each partial flow p'^- we build a new partial flow: $(C_p .p'^-) - (C_{p'} .p^-)$
else all partial flows p^- are kept in the new family **endif**

c) **If** there exists a partial flow q such that $B_q \neq 0$ **then**
for each partial flow q' we build a new partial flow: $(B_q .q') - (B_{q'} .q)$
else all partial flows q are kept in the new family **endif**

d) **If** there exists a partial flow p^- such that $C_p \neq 0$ **and** there exists a partial flow q such that $B_q \neq 0$ **then**
a new flow the type of which is 1, is built: $(C_p .q) - (B_q . \mathbb{p}^-)$
where $\mathbb{p}^-(x) == \sum_y p^-(x,y) - \sum_y p^-(y,x)$
(where in the first sum the summation is over $\{y \mid y < x\}$
and the second one is over $\{y \mid y > x\}$)
endif

In order to enable the iteration of this step, we will only have now to give the transformation of colour functions for the partial flows **q** , $\mathbf{p^+}$, $\mathbb{p}^+$ and $\mathbb{p}^-$. The tables of transformations are given in [Had88].
These transformations can easily be deduced as shown by the following example:

Let $I(p^+,t^+)$ = <X1,X2> , then if we want to find in the new net $I'(\mathbb{p}^+, t^+)$, we apply:

<X1 ,X2> ((x1,x2) , $\sum_y$ (x,y) + $\sum_y$ (y,x)) =

$\sum_y$ <X1 ,X2> ((x1,x2) ,(x,y)) + $\sum_y$ <X1 ,X2> ((x1 ,x2) , (y,x)) =

Eq(x1,x) + Eq(x2,x)

Then $I'(\mathbb{p}^+, t^+)$ (<x1 ,x2>,x) = Eq(x1,x) + Eq(x2,x) that is to say $I'(\mathbb{p}^+, t^+)$ = X1 + X2

Justification of this step

As said above, this step is an elimination, then the underlying idea is the following: in an elimination rather than choosing a pivot belonging to the partial flows, one can divide the family of partial flows in subsets, then one can do the elimination for any subset with a local pivot and at last one can eliminate the local pivots with one of them. The elimination of a subset of partial flows gives the same kind of flows while the elimination between two local partial flows gives a new flow similar to one of the two old flows. ◊◊◊

2.4 Reduction of the Partial flows

Once we have eliminated the '+' transitions and the '-' transitions, let F be the generative family computed before this step.

$F = \{p^+ ,p'^+ ,...\} \cup \{p^-,p'^- ,...\} \cup \{q,q',... \} \cup \{r,r',...\}$

The aim of this step is to transform each partial flow p^+ and p^- into definitive flows before the elimination of transitions of type 1.

Let p^+ be a partial (but not global) flow. We associate to p^+ two new flows p^{++} and $\mathbb{p}^+$ the type of the first one is 2 and the type of the second one is 1. Moreover p^{++} is a global flow. The underlying isomorphisms are defined by:

- $\mathbb{p}^+(x) \equiv \sum_y p^+(x,y) + \sum_y p^+(y,x)$ as above
(where in the first sum the summation is over $\{y \mid y < x\}$
and the second one is over $\{y \mid y > x\}$)
- $p^{++}(x,y) \equiv p^+(x,y) - [1/(n-1)] .[\mathbb{p}^+(x) + \mathbb{p}^+(y)] + [2/(n-1).(n-2)] . \mathbb{p}^+(e)$

Let p^- be a partial (but not global) flow. We associate to p^- two new flows p^{--} and $\mathbb{p}^-$ the type of the first one is 2 and the type of the second one is 1. Moreover p^{--} is a global flow. The underlying isomorphisms are defined by :

- $\mathbb{p}^-(x) \equiv \sum_y p^-(x,y) - \sum_y p^-(y,x)$ as above
(where in the first sum the summation is over $\{y \mid y < x\}$
and the second one is over $\{y \mid y > x\}$)
- $p^{--}(x,y) \equiv p^-(x,y) - [1/n] .[\mathbb{p}^-(x) - \mathbb{p}^-(y)]$

Once this tranformation done, the only partial but not global flows are the partial flows of type 1 and type 0.

Justification of this step

It is easy to show that p^+ is generated by the partial flows p^{++} and $\mathbb{p}^+$ and that p^- are generated by the partial flows p^{--} and $\mathbb{p}^-$ .Thus the only difficulty of this step is to show that p^{++} and p^{--} are global flows and this can be found in [Had88]. ◊◊◊

2.5 Elimination of the transitions with domain D

Let F be the generative family computed before this step:

$F = \{p^+, p'^+, ...\} \cup \{p^-p'^-, ...\} \cup \{p^{++}, p'^{++}, ...\} \cup \{p^{--}, p'^{--}, ...\} \cup \{q,q',... \} \cup \{r,r',...\}$

The only partial flows of F are the flows of type 1 and the flows of type 0. Moreover we have only to eliminate the transitions of type 1. Thus this elimination is a very simple case of the general method given in [Had87a].

2.6 Application to the examples

The algorithm given above may be applied with simplification to special cases of associative nets but we will not examin it here. For more explanations, the reader may refer to [Had88]. For instance a simple simplification is to skip the first step when the net is already normalized and a more subtle simplification may also be done when no valuations <X2,X1 > appears in the net, etc...

We strongly use these simplifications on the two examples given above. In this paper we only give the flows computed by the algorithm and their interpretations.

Example 1 The database with multiple copies

(1) $\sum_x$ Wait(x) + Mutex(e) = 1 ,where the summation is over D
There at most one site waiting for acknowledgments.

(2) $\forall x \in D$, Idle(x) + Wait(x) + $\sum_y$ Update(y,x) = 1, where the summation is over D
Each site is idle, waits for acknowledgments or updates its database.

(3) $\forall x \in D$, (N -1) .Wait(x) - $\sum_y$ [Mess(x,y) + Update(x,y) + Ack(x,y)] = 0 , where the summation is over {y I y≠x}
If one site is waiting then n-1 messages, changes and acknowledgments are in the net for the other sites.

(4) $\forall x \in D$, Mess(x,x) + Update(x,x) + Ack(x,x) = 0
There is no message,change or acknowledgment done by a site on its own request.

(5) $\forall x,y \in D$, (N-1).[Mess(x,y) + Update(x,y) + Ack(x,y)] - $\sum_y$ [Mess(x,y) + Update(x,y) + Ack(x,y)] = 0
where the summation is over {y I y≠x}

Two sites different from a third one x have the same number of messages,changes or acknowledgments on the request of x. Then from (3) we can deduce that this number is 1 if and only if x is waiting and 0 in the other case.

Example 2 The telephone system

(1) $\forall x \in D$, Idle(x) + Wait(x) + Conn(x) + Disc(x) = 1
A phone number is idle or waiting or connected or disconnected.

(2) $\forall x \in D$, Wait(x) -$\sum_y$ Req(x,y) = 0 , where the summation is over D
With the help of (1), a phone number is waiting if and only if there is one request done by this number.

(3) $\forall x \in D$, Conn(x) -$\sum_y$ [Call(x,y) + Call(y,x)] = 0 , where the summation is over D
With the help of (1), a phone number is connected if and only if there is a call the calling or the called number of which it is. Then no number may call itself.

3 COMPUTATION OF FLOWS OF AN ORDERED NET

The study developped hereunder requires progressing from integer matrix algebra to polynomial matrix algebra. The complexity of polynomial algebra doesn't allow both the computing of a generative family of flows and parameterization in the general case of a complete structure. We have produced three different solutions using the following restrictions:

1- no parametrization, complete structure, generative family,
2- parametrization, complete structure, no generative family,
3- parametrization, restriction of structure, generative family.

The first case is simply an application of invariant theory and we use the formal Macaulay [Mac86] calculus system to solve the polynomial linear equations. The second, developed in [Cou88], finds invariants generated both by diffusion and by successor functions. The last one is the case developped in this paper. The colour domain is restricted to one class to obtain polynomial algebra with one variable. This simpler algebra permits diagonalization of the matrix and solution of the complete equations together with parametrization.

Notations

a- $[\Omega]$ is the space of rational combinations of Ω elements. Hence a coloured place vector $V= (v_p)_{p\in P}$ with $v_p \in [\Omega]$ is an element in $E = [\Omega]^P$.

b- φ_x is the characteristic function of the element x in $[\Omega]$:

for $X=\sum_{k1,...,kq\in \mathbf{N}} a(k_1,...,k_q)<k_1,...,k_q>$

$\varphi_X = \sum_{k1,...,kq\in \mathbf{N}} a(k_1,...,k_q)<X_1\oplus k_1,..., X_q\oplus k_q>$

Noticeably φ_X (<0>) = x where <0> is <0, ...,0>.

c- <<x, y>> is the scalar product of two elements of $[\Omega]$.

d- Let **Q**[$g_1,...,g_q$] be the polynomial function ring generated by $g_1,...,g_q$

e- Each polynomial function f defines a linear endomorphism in $[\Omega]$. We denote f^t its transposed function:

$\forall x,y \in [\Omega]$ << f(x), y >> = << x , $^tf(y)$ >>

This function is determined by

for $f = \sum_{k1,...,kq\in \mathbf{N}} a(k_1,...,k_q)<X_1\oplus k_1,..., X_q\oplus k_q>$

$f^t = \sum_{k1,...,kq\in \mathbf{N}} a(k_1,...,k_q)<X_1\Theta k_1,..., X_q\Theta k_q>$

f- Let $Q[\lambda_1,...,\lambda_q]$ be the polynomial ring over **Q**.

Let $[\lambda_1^{N1}-1, ..., \lambda_q^{Nq}-1]$ be the ideal generated by $\lambda_1^{N1}-1, ..., \lambda_q^{Nq}-1$

We denote **A*** the quotient ring $Q[\lambda_1,...,\lambda_q]/ [\lambda_1^{N1}-1, ..., \lambda_q^{Nq}-1]$.

g- Let **U*** the place module over the ring **A***.

3.1 Flows of an ordered net

The flows of coloured net are the solutions of the equation:

(1) **V.I=0**

where $I = (I_{p,t})$ is the incidence matrix, $V = (V_p)$ a vector in **E**. This equation gives the equation system:

(2) $\forall\ t\in T, \forall y\in[\Omega]\ \sum_{p\in P} <<V_p, I_{p,t}(y)>>=0$.

Let a flow $V = (V_p)$, the P-tuple (φ_{vp}) of functions generates a set of flows $V(x)=(\varphi_{vp}(x))$ for every $X \in [\Omega]$. Hence, using the characteristic functions, the flows in an ordered net have the symbolic expression $(\varphi_{vp}(x))$.

Theorem 1 If the vector V = (vp) is a flow then $V(x) = (\varphi_{vp}(x))$ is a flow for all element x in [Q].

Proof:

Consider the flows $V= (v_p)$, and the P-tuple of functions $V(x) = (\varphi_{vp}(x))$ such that the flows V are equal to (V<0>).

The flows definition gives the identity:

$\forall\ t\in T, \forall x,y\in[\Omega]\ \sum_{p\in P} <<\varphi_{vp}(<0>), I_{p,t}(y)>>=0$.

Using the commutativity of functions and property of the transposed function, we have:

$\forall\ t\in T, \forall x,y\in[\Omega]$

$\sum_{p\in P} <<\varphi_{vp}(x), I_{p,t}(y)>>= \sum_{p\in P} <<\varphi_{vp}o\varphi_x\ (<0>), I_{p,t}(y)>>= \sum_{p\in P} <<\varphi_x\ o\varphi_{vp}\ (<0>), I_{p,t}(y)>>=$

$\sum_{p\in P} << \varphi_{vp}\ (<0>), {}^t\varphi_x\ (I_{p,t}(y))>>= \sum_{p\in P} << \varphi_{vp}\ (<0>), I_{p,t}\ ({}^t\varphi_x\ (y))>> =0$

Thus, for x in $[\Omega]$, $V(x) = (\varphi_{vp}(x))$ is a flow.

◊◊◊

The flows are simply linked to linear equation in coloured function ring.

Theorem 2 $V(x) = (\varphi_{vp}(x))$ is a flow if and only if the vector function (φ_{vp}^t) is the solution of the linear equation $(\varphi_{vp}^t).I = 0$ in the coloured function ring.

Proof:
Consider the functions $V(x) = (\varphi_{vp}(x))$.
The identity

$\sum_{p\in P} <<\varphi_{vp}(x), I_{p,t}(y)>>= \sum_{p\in P} << x, \varphi_{vp}^{t} . I_{p,t}(y)>>= <<x, \sum_{p\in P} \varphi_{vp}^{t} . I_{p,t}(y)>>$

proves the theorem.

Indeed, if $V(x) = (\varphi_{vp}(x))$ is a flow, then $\sum_{p\in P} <<\varphi_{vp}(x), I_{p,t}(y)>> = 0$.

According to the previous identity $<<x, \sum_{p\in P} \varphi_{vp}^{t} . I_{p,t}(y)>>=0$, $\forall x,y\in[\Omega]$.

This property proves that the functions $\sum_{p\in P} \varphi_{vp}^{t} . I_{p,t} = 0$ for every transition t.

These equalities represent exactly the linear equation $({}^{t}\varphi_{vp}).I = 0$.

In the same way, if the vector function (φ_{vp}^{t}) is a solution of the equation $(\varphi_{vp}^{t}).I = 0$ then
$V(x) = (\varphi_{vp}(x))$ is a flow.
◊◊◊

The previous theorem shows that the flow calculation can be reduced to the resolution of a linear equation in the colour function ring. This ring contains all polynomial functions generated by $g_1,...,g_q$.
The surjective canonical homomorphism Ψ

$\Psi: Q[\lambda_1,..,\lambda_q]\rightarrow Q[g_1,...,g_q]$

$\alpha[\lambda_1,..,\lambda_q]\rightarrow \Psi(\alpha(\lambda_1,..,\lambda_q))=\alpha[g_1,...,g_q]$

gives an ring isomorphism between $Q[g_1,...,g_q]$ and $Q[\lambda_1,..,\lambda_q] / Ker(\Psi)$.

Theorem 3 The polynomial function ring $Q[g_1,...,g_q]$ is isomorphic to the ring **A*** = $Q[\lambda_1,..,\lambda_q] / [\lambda_1^{N1}, ..., \lambda_q^{Nq}-1]$

Proof:

We want to prove that $Ker(\Psi) = [\lambda_1^{N1}-1, ..., \lambda_q^{Nq}-1]$
Consider the polynomial $P(\lambda_1,..,\lambda_q)$ in $Ker(\Psi)$.

Using the simple equality $\lambda_i^{Ni} = (\lambda_i^{Ni}-1)+1$, the polynomial P gives

$P(\lambda_1,..,\lambda_q) =\alpha(\lambda_1,..,\lambda_q)+\sum_{i=1,...,q} \beta(\lambda_1,..,\lambda_q). (\lambda_i^{Ni}-1)$

where the order of $\alpha(\lambda_1,..,\lambda_q)$ over λ_i is smaller than N_i for all i = 1,...,q.

$\alpha(\lambda_1,..,\lambda_q)= \sum_{k1<N1,...,kq<Nq} a(k_1,...,k_q) \lambda_1^{k1}\ldots\lambda_q^{kq}$

The principal property of successor function g_i is the equality $g_i^{Ni} = Id$. So we have

$\psi(\lambda_i^{Ni}-1) = (g_i^{Ni} -Id) =0$.

If we transform now the polynomial P by functionψ, we obtain

$\psi(P(\lambda_1,..,\lambda_q)) =\psi(\alpha(\lambda_1,..,\lambda_q)) =0$.

and

$\alpha(g_1,...,g_q)(<0>) = \sum_{k1<N1,...,kq<Nq} a(k_1,...,k_q) g_1^{N1}\ldots g_q^{kq} (<0>)$

$= \sum_{k1<N1,...,kq<Nq} a(k_1,...,k_q)<k_1,...,k_q>$

This equality gives the unique representation of the element $\alpha(g_1,...,g_q)(<0>)$ in $[\Omega]$.

Such a combination is zero if and only if the coefficients are all zero. So we obtain

$\alpha(\lambda_1,..,\lambda_q)=0$

$P(\lambda_1,..,\lambda_q) =\sum_{i=1,...,q} \beta(\lambda_1,..,\lambda_q). (\lambda_i^{Ni}-1)$

which proves the theorem.
◊◊◊

The isomorphism allows us to identify the elements of $Q[g_1,...,g_q]$ with the elements of **A***. Thus, every property in $Q[g_1,...,g_q]$ can be translated into a property in **A***.

The first definition translates the incidence matrix into its equivalent polynomial matrix.

Definition 4 Let I be an incidence matrix of an ordered regular net. We call polynomial incidence matrix of I, the PxT matrix I* over the ring A* defined by:

if $I(p,t,X_1,...,X_q)= \sum_{k1,...,kq \in \mathbf{N}} I(p,t,k_1,...,k_q)<X_1 \oplus k_1,..., X_q \oplus k_q>$

then $I^*(p,t)= \sum_{k1,...,kq \in \mathbf{N}} I(p,t,k_1,...,k_q)\ \lambda_1^{k1}...\lambda_q^{kq}$

In the philosophers example, the polynomial matrix

	t1	t2
Talking	-<X>	<X>
Eating	<X>	-<X>
Fork	[-<X+ 1>-<X>	<X+1>+<X>]

$I^* = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ -\lambda-1 & \lambda+1 \end{bmatrix}$ (columns: t1, t2)

The following theorem translates the theorem 2 in the ring **A***.

Theorem 5 These two following propositions are equivalent:

1- The vector V* of the module **U*** defined by

$V^*(p)= \sum_{k1,...,kq \in \mathbf{N}} v_p(b_1,...,b_q)\ \lambda_1^{b1}...\lambda_q^{bq}$

is a solution of the linear equation ${}^tV^*.I^*=0$ in **A***.

2- For every (x1,...,xq) in $C_1 \times ... \times C_q$. the vector V of the vector space **E** defined by

$V(p)= \sum_{k1,...,kq \in \mathbf{N}} v_p(b_1,...,b_q)<X_1 \Theta b_1,..., X_q \Theta b_q>$

is a flow.

Thus, the problem comes down to a resolution in the ring **A*** which allows as to use the classical results on the subject [Bur83], [Bus85], [Ful83], [Laz77], [Laz81], [Mac81] [Tur83].

3.2 Polvnomial eguations

The calculus of invariants in an ordered net is the resolution of linear equations in **A***

$V^*.I^*=0$ (1)

The classical method uses elementary matrix transformations. These transformations simplify the matrix and in most cases a diagonal matrix can be obtained. Using the resulting diagonal matrix, a solution can be obtained.

For our purposes, this method may be used by restricting the coloured domain to a single class. In this way, the coefficients of the matrix are single-variable polynomials. This kind of matrix in $\mathbf{Q}[\lambda]$ is called a λ-matrix in mathematical literature [Tur61]. Smith's canonical form for the A-matrix states: Every λ-matrix can be diagonalized by elementary transformations. These elementary transformations are always valid in **A*** so the Smith method allows us to diagonalize the matrix in A*. Using this key result, the linear equations obtained are equivalent to the following equations in the polynomial ring $\mathbf{Q}[\lambda]$:

$\alpha(\lambda).P(\lambda) = \beta(\lambda).(\lambda^{N-1})$

Such equations can be solved with N let variable.

3.2.1 Interchange of rows and columns in matrix

Consideration of the identity of

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & \\ 1 & -1 & \\ -\lambda & -1 & \lambda+1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & \\ -1 & 1 & \\ -\lambda-1 & \lambda+1 & \end{bmatrix}$$

shows that the effect of premultiplying by a unit matrix in which the first two rows have been interchanged is equivalent to interchange those rows of I*. In general, let $P_{(i,j)}$ denote a unit matrix in which the i th and j th rows have been interchanged without disturbing the remaining rows. Evidently the effect of multiplying 1* by such a matrix will be:

$P_{(i,j)}.I^*$: interchange row i and row j.

$I^*.P_{(i,j)}$:interchange column i and column j.

It must be observed that $(P_{(i,j)})^2 = Id$, since a second interchange restores the unit matrix.

3.2.2 **Linear combination of Rows or Columns in matrix**

Consider again the identity:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \lambda+1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ -\lambda-1 & \lambda+1 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 0 & 0 \end{bmatrix}$$

Premultiplication of I* by a unit matrix supplemented by the element A+ 1 in the indicated position (3,2) has performed the operation $row_3 + (+1)\, row_2$. More generally let:

$$H = Id + (h)_{(i,j)} \qquad (i \neq j)$$

denote the unit matrix with an element h inserted in the (i,j)th position; then we have readily

$H.I^*$: $row_i + h.row_j$

$I^*.H$: $col_i + h.col_j$

To these facts, may be added the following useful compositions involving H^{-1}, which are readily seen to satisfy the relation $H^{-1} = Id - (h)_{(i,j)}$.

3.2.3 **Linear transformations of equations**

Let P be an inversible linear transformation of **U*** and suppose we look at the solutions of V*.I* = 0, but with the solutions related to the image of the canonical basis by P , then if we denote W" the new variables, we obtain: V* = W* .P*, that is to say W* must verify:

W* .(P .I*) =0

In other words, the effect of transforming the variables V* into W* is equivalent to premultiplying of I* by P, while the effect of rearranging or combining the equations in the elementary familiar way, is equivalent to postmultiplication by P. In canonical reduction we use both methods of transformation, sometimes operating on the matrix itself, sometimes setting up new systems of variables. Since P is inversible the new equation is equivalent to the old one.

In the philosophers example:

a) One substitutes t1+ t2 for t1

I*

	t1	t2
Thinking	[-1	1]
Eating	[1	-1]
Fork	[-λ-1	λ+1]

$I^*.H$ where $H = Id + (1)_{(2,1)}$

	t1	t1+12
Thinking	[-1	0]
Eating	[1	0]
Fork	[-λ-1	0]

b) Now one changes the basis *(Thinking, Eating,Fork)* to the new basis *(Thinking +Eating ,Eating ,Fork)* which is equivalent to premultiply by $H = Id + (1)_{(2,1)}$.

	t1	t1+t2
Thinking +Eating	[0	0]
Eating	[1	0]
Fork	[-λ-1	0]

c) If we continue such transformations we obtain a much simpler matrix

	t1	t1+t2
Eating	[1	0]
Thinking +Eating	[0	0]
Fork +(λ+1) *Eating*	[0	0]

This simple example, which shows how a matrix transformation may be carried out through the introduction of suitable new variables and the combination of equations, and incidentally how the kind of transformation $H_1.I^*.H_2$ may arise, introduces the work of the following section.

3.2.4 Smith's canonical form

A class of matrix of historical interest is the one in which each element belongs to the ring of integers. It was shown by H.J.S. Smith that such a matrix A could be reduced by matrices P and Q which are products of elementary matrices (Cf 3.2.1 & 3.2.2) to the diagonal matrix:

$$P.A.Q = \begin{bmatrix} \alpha_1 & \dots & \dots & \dots & \dots \\ \dots & \alpha_2 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \alpha_r & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

The ring property which allows such reductions is the existence of the euclidian division. In the general case of the ordered net we manipulate multi-variable polynomials. In this ring there is no euclidian division. So we restrict the coloured domain to a single class to get polynomials with a single variable. The euclidian division exists in **Q**[λ]. Then we can transform the matrix in **Q**[λ] by transformations which are always valid in **A***. We denote the single variable by λ, and all polynomial matrices are called λ-matrices. The Smith theorem is given by:

Theorem Every λ-matrix of rank r can be reduced by linear transformations to the equivalent diagonal form containing exactly r isolated elements,

$$D^* = H_1.I^*.H_2 = \begin{bmatrix} E_1(\lambda) & \dots & \dots & \dots & \dots \\ \dots & E_2(\lambda) & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & E_r(\lambda) & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

where $E_1(\lambda) = \delta_1$, $E_2(\lambda) = \delta_1\delta_2$,.., $E_r(\lambda) = \delta_1\delta_2\dots\delta r$ each δ_i being a non-zero polynomial in λ. (The proof of this theorem is in [Tur61])

This theorem shows that elementary transformations convert the linear equations in **A*** into a system of the following type:

for i=1, ...,r $\quad \alpha_i(\lambda) .E_i(\lambda) = 0$

In the philosophers example, the diagonal matrix

	t1	t1+t2
Eating	[1	0]
Thinking +Eating	[0	0]
Fork +(λ+1) *Eating*	[0	0]

We find at once, two solutions

(1) *Thinking +Eating*

(2) *Fork* + (λ+ 1) *Eating*

Hence, using the ring isomorphism, this solution gives the invariants for all x =1,...,N-1:

(1) *Thinking (x)+Eating* (x) = 1

Each philosopher is thinking or eating.

(2) *Fork* (x) + *Eating* (x-1) + *Eating* (x) = 1

Each fork is free or used by one of the closest philosophers.

3.2.5 Resolution of the equation α(λ) .E(λ) = 0

In the case of polynomials with one variable A, the set **A*** is the ring of polynomials **Q**[λ] quotiented by the ideal $[\lambda^{N-1}]$. In such a ring, a polynomial P(λ) is equal to zero if it is a multiple of λ^{N-1}. So the equation α(λ) .E(λ) = 0 in **A*** is equivalent to the equation with two unknowns α(λ) .E(λ) = β(λ) . (λ^{N-1}) in **Q**[λ]. We now consider the last equation and polynomials in **Q**[λ].

If we denote $HCF(E(\lambda),\lambda^{N-1})$ as the highest common factor of E(λ)) and λ^{N-1}, by using divisibility rules, the solutions of the second equation are

$\alpha(\lambda) = K(\lambda) .[(\lambda^{N-1}) / HCF(E(\lambda),\lambda^{N-1})]$

$\beta(\lambda) = K(\lambda) .[E(\lambda) / HCF(E(\lambda),\lambda^{N-1})]$

where K(λ) is a polynomial in **Q**[λ]. Hence, the polynomial $G_N(\lambda) = [(\lambda^{N-1}) / HCF(E(\lambda),\lambda^{N-1})]$ generates all the solutions of the equation in **A***.

Theorem 1 The polynomial $G_N(\lambda) = [(\lambda^{N-1}) / HCF(E(\lambda),\lambda^{N-1})]$ generates all solutions to the equation α(λ).E(λ)= 0 in **A***.

One remarks that if the polynomials $E(\lambda)$ and λ^{N-1} are relatively prime, $G_N(\lambda)$ represents zero in **A*** and so the equation has only the solution zero.

The solution to the equation in **A*** is based on the calculation of $HCF(E(\lambda),\lambda^{N-1})$. Because of the finite number of divisors of $E(\lambda)$, the set of HCF for every integer N is finite. We represent this set by $\{ HCF(E(\lambda),\lambda^{N1-1}),..., HCF(E(\lambda),\lambda^{Nd-1})\}$ where the integer $N_1,...,N_d$ are the smallest ones associated with each of these HCFs. Therefore, for an integer N, there exists an integer N_i such that HCF is equal for both N and N_i and moreover we can prove that N is divisible by N_i [Cou88].

Theorem 2 Let N be an a integer and N_i an integer such that

$$HCF(E(\lambda),\lambda^{N-1}) = HCF(E(\lambda),\lambda^{Ni-1})$$

Hence N_i is a factor of N. If we denote $N=k.N_i$, the generator $G_N(A)$ is as a function of the $G_{Ni}(A)$

$$G_N(\lambda) = (1 + \lambda^{Ni} +...+ \lambda^{(k-1)Ni})\ G_{Ni}(\lambda)$$

The preceeding theorem induces an algorithm to find the generator of the solutions for N given, but this algorithm implies the computation of the corresponding HCF which needs polynomial operations and thus can be expansive. The following theorem shows that if we keep the $G_{Ni}(\lambda)$, we do not need this computation but a computation in the integer ring which is simpler.

Theorem 3 Let N an integer. A generative family of solutions of the equation $\alpha(\lambda).E(\lambda)= 0$ over **A*** is defined by:
$\{ (1 + \lambda^{Ni} +...+ \lambda^{(k-1)Ni})\ G_{Ni}(\lambda) \mid \exists\ k , N = k.\ N_i \}$

Example:

The philosophers problem gives a good example of such eqation. We have better solve entirely the invariant equation but we can continue if we want to find an invariants on the free forks. We come down to solve the equation:

$$\alpha(\lambda).(\lambda+1)= 0$$

The calculation of $HCF(\lambda+1), \lambda^N-1)$ gives two characteristic numbers:

- For N=1, $HCF((\lambda+1), \lambda-1) =1$. In this case, the two polynomials are relatively prime, we obtain only the solution zero.
- For N=2, $HCF((\lambda+ 1), \lambda^2-1) = \lambda+1$ and $G_2(\lambda)= \lambda-1$.

Using the previous theorem, the generative family of solution for an integer N is given by

If $N = 2.k$ then $(1+ \lambda^2 + ...+ \lambda^{2(k-1)})(\lambda-1)$ is a solution.

By application of the ring isomorphism, this solution gives an invariant of the philosophers model in the case of N=2.k

Fork(<0>)+ *Fork*(<2>)+ ...+ Fork(<2.k-2>)
= *Fork*(<1>)+ *Fork*(<3>)+ ...+ Fork(<2.k-1>)

Hence, when the number of philosophers is even, there is the same number of even and odd free forks.

3.3 Application to the synchronized clocks model

Incidence matrix I

	ϕ_L	ψ_L	ϕ_R	ψ_R	
ZL [	0	<X+1>-<X>	0	0	]
P1 [	-<X>	<X+1>	0	0	]
P2 [	<X>	-<X>	0	0	]
P3 [	<X>	0	0	-<X>	]
P4 [	0	-<X>	<X>	0	]
P5 [	0	0	<X>	-<X>	]
P6 [	0	0	-<X>	<X+1>	]
ZR [	0	0	0	<X+1>-<X>	]

Polynomial matrix I*

	ϕ_L	ψ_L	ϕ_R	ψ_R	
ZL [	0	λ-1	0	0	]
P1 [	-1	λ	0	0	]
P2 [	1	-1	0	0	]
P3 [	1	0	0	-1	]
P4 [	0	-1	1	0	]
P5 [	0	0	1	-1	]
P6 [	0	0	-1	λ	]
ZR [	0	0	0	λ-1	]

Diagonalization of the matrix I*

With elementary operations on rows and columns, we obtain the diagonal matrix:

	ϕ_L	$\lambda\phi_L+\psi_L$	ϕ_R	$\psi_R+\lambda\phi_R+\psi_L+\lambda\phi_L$
P1	[-1	0	0	0]
P4+P6- Z_R	[0	-1	0	0]
P6	[0	0	-1	0]
Z_R	[0	0	0	λ-1]
P1+P2- Z_L	[0	0	0	0]
PS+P6- Z_R	[0	0	0	0]
Z_L -λZ_R +(λ-1)(P4+P6)	[0	0	0	0]
P1+P3+λ(P4+P6)- (λ+ 1)Z_R	[0	0	0	0]

Solutions of the linear equations

(a) $(1+\lambda+\ldots+\lambda^{N-1})Z_R$
(b) P1 + P2 -Z_L
(c) P5 + P6 -Z_R
(d) ZL -λR +(λ-1)(P4+P6)
(e) P1 +P3+λ (P4+P6)- (λ+1)Z_R

Generative family of flows

(a) $Z_R(O) + \ldots + Z_R(N-1)=1$

There is at most one token in the place Z_R.It represents the position of the counter Z_R.

(b) $\forall x \in C$, $P1(x) + P2(x) = Z_L(x)$
(c) $\forall x \in C$, $P5(x) + P6(x) = Z_R(x)$

There is at most one token in the place P1 or P2 (resp. P5, P6).This token is also the position of the corresponding counter.

(d) $\forall x \in C$, $Z_L(x) +P4[x-1] +P6[x-1] = Z_R[x-1] + P4(x) +P6(x)$
(e) $\forall x \in C$, $P1(x) +P3(x) + P4[x-1] +P6[x-1] = Z_R(x) + Z_R[x-1]$

These two invariants prove the synchronization of the counters. If we add the both invariants (d) and (e) for the respective colours x and x+1, we obtain:

$Z_L(x) + P1(x+1) +P3(X+1) + P4[x-1] +P6[x-1] = ZR[x-1] + ZR(x) + ZR[x+1]$

Thus if ZL(x)=1 then ZR[x-1] =1 or ZR(x) =1 or ZR[x+1] =1.

CONCLUSION

In this paper we have extended the flows computation for the regular coloured nets to new categories of coloured nets. These new models are not arbitrary models since many of the examples which have been presented here were introduced by the other authors and describe real situations. The first extension, called associative nets, is obtained by allowing associations between objects of a same class. For instance, the associative nets are suitable for communication protocols (telephone system). The second extension, called ordered nets, is defined by ordering the colour domains and introducing the successor functions. For instance, the ordered nets are suitable for files, rings or counters gestion. As for regular coloured nets, the parametrization is integrated in the definition in order to have a general specification and possibly a complete validation.

The algorithms, we have developped here, verify - as the previous one - the fundamental criteria for a good flows computation: the intensive representation of flows, the independance from the cardinality of colour domains and the computation of a generative family of flows.

Following the same principles these algorithms have been found after a theoretical study of the algebraic structure of the flows space which leads to the decomposition of this space into subspaces of particular flows and the isomorphism theorems between flows space of different nets. On the one hand, the algorithms compute also in the polynomial ring in order to handle the parametrization. On the other hand completely new ideas have been found such as in associative nets the transformation of colour domains or in ordered nets the computing in quotient polynomial rings. Moreover these algorithms suggest that there is a strong link between the type of flows and the type of colour functions used in a net.

We would like to emphasize that we have presented systematic methods to compute a generative family of flows on general models taking into account the parametrization of the net. In order to finish this work and to obtain a general algorithm, it remains to unify the different algorithms, to slightly increase the model (by introducing the constant functions) and to synthetize the different algorithms presented till now. At this point the modellers will have at their disposal a powerful tool to validate their applications.

REFERENCES

[All84] H. ALLA, P. LADET, J. MARTINEZ, M. SILVA: "Modelling and validation of complex systems by coloured Petri nets", Fifth european workshop on applications and theory of Petri nets, Aarhus, Denmark june1984, in "Advances in Petri nets 84", L.N.C.S. n° 188, G.Rozenberg ed., Springer Verlag, 1985 pp15-31.

[Bra83] G.W. BRAMS: "Reseaux de Petri. Theorie et pratique". Masson editeur, Paris, 1983.

[Bur83] B. BUCHBERGER: "An algorithmic method in polynomial ideal theory", in Recent tools in multidimentional systems theory, N.K.Bose ed., D.Reidel Publishing comp., 1983.

[Bus85] A.T. BUSTON, B.M. STEWART: "Systems of linear congruences", Canadian Journal of Mathematics vol 7, n° 3,1955, pp 358-368.

[Cou88] J.M. COUVREUR "Un calcul d'une base de flots pour les réseaux ordonnés" Rapport de recherche Masi to appear

[Ful55] L.E. FULLER "A canonical set for matrices over a principal ring modulo m" Canadian Journal of Mathematics, vol 7, n° 1,1955, pp 54-59.

[Gen81] H.J. GENRICH, K. LAUTENBACH: "System modelling with high-level Petri nets", Theoretical computer science 13,1981, pp 103-136.

[Gen82] H.J. GENRICH, K. LAUTENBACH: "S-invariance in predicate transition nets", third european workshop on applications and theory of Petri nets, Varenna Italy, september1982, in "Applications and Theory of Petri nets", Informatik Fachberitche no66, G.Rozenberg ed., Springer Verlag, 1983, pp 98-111.

[Had86] S. HADDAD, C. GIRAULT: "Algebraic structure of flows of a regular net", Seventh european workshop on applications and theory of Petri nets, Oxford England, june 1986, in "Advances in Petri nets 87", L.N.C.S. n° 266, G. Rozenberg ed., Springer-Verlag, 1987 pp 73-88.

[Had87a] S. HADDAD: "Un calcul d'une base de flots pour les reseaux colores", Deuxieme colloque C3 Angouleme, France, 1987.

[Had87b] S. HADDAD: "Une categorie reguliere de reseau de Petri de haut niveau: definition, proprietes et reductions. Application a la validation de systemes distribues", These de l'Universite Pierre et Marie Curie, Paris, juin1987.

[Had88] S. HADDAD: "Un calcul d'une base de flots pour les reseaux associatifs", Rapport de recherche MAS!, Universite Paris 6, to appear.

[Jen81] K. JENSEN: " How to find invariants for coloured Petri nets" 10th symposium on Mathematical foundations of computer science 1981, L.N.C.S. vol 118, Springer-Verlag 1981, pp 327-338

[Jen82] K. JENSEN: "High-Level Petri nets" Third European workshop on applications and theory of Petri nets, Varenna Italy,September 1982 pp 261-276.

[Lau85] K. LAUTENBACH, A. PAGNONI: "Invariance and duality in predicate transition nets and in coloured nets", Arbeitspapiere der G.M.D, no132.

[Laz77] D. LAZARD: "Algebre lineaire sur K[X1, ...,Xn] et elimination", Theorical Computer Science, 15, 1981, pp 77-110.

[Laz81] D. LAZARD: "Résolution des systèmes d'équations algébriques", Theorical Computer Science, 15, 1981, pp 77-110.

[Mac86] MACAULAY: "A computer algebra system for algebraic geometry", version 1.219, Columbia University, april 1986.

[Mem83] G. MEMMI: "Methodes d'analyse de reseaux de Petri, reseaux a files et applications aux systemes temps reel", Thèse d'etat,Universite Pierre et Marie Curie, Paris, juin1983.

[Sil85] M. SILVA, J. MARTINEZ, P. LADET, H. ALLA: "Generalized inverses and the calculation of symbolic invariants for coloured Petri nets", Technique et science informatique, Vol.4 no1, 1985, pp 113-126.

[Tur61] H. W. TURNBULT, A.C.AITKEN: "An introduction to the theory of canonical matrices", Dover publications, New York 1961

[Vau84] J. VAUTHERIN, G. MEMMI: "Computation of flows for unary predicate transitions nets", Advances in Petri nets, Lecture Notes in Computer Science n° 188, Springer-Verlag 1984, pp. 455-467.