VALIDATION OF PARALLEL SYSTEMS WITH COLOURED PETRI NETS

S. HADDAD **J.M. COUVREUR**

UNIVERSITE Paris VI et C.N.R.S. MASI
4 Place Jussieu, 75252 Paris Cedex 05

phone number: 43 36 25 25 P. 3423
fax:43294196 telex: UPMCSIX 200 145 F
electronic mail: mcvax ! Inria !litp I seh

Abstract

This paper presents the coloured Petri net model and the flows computation and the reduction theory we have developped for it. We show how coloured nets can model realistic parallel systems with the help of coloured domains and functions. Since the computation flows generates invariants family. it can be used in order to verify safeness properties. The reduction rules which transform the net into a smaller one with the same behaviour as the original one is a suitable tool for verifying the liveness properties. Two significant applications and their validation are given: a database management and the synchronization of two logical clocks.

INTRODUCTION

In many models of parallelism, there are only two methods of validation: analysis of the accessibility 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 whilst the drawbacks of the second one are:

-a property may be true even if not provable 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 reduction theory [Ber83]. So, as abbreviations of Petri nets -coloured nets [Jen82] and predicate transition nets [Gen81], [Lau8S] –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 [Jen81], [Gen82] , [Vau86] , [Sil8S] and the reduction theory [Col86].

The main purpose of this paper is the presentation of the flows computation and the reduction theory we have developped for coloured nets and their efficient use in automatic prooves of a modelling [Had86] , [Had87a] , [Had87b] , [Cou88].

Generally in a parallel system the modeller searchs two kinds of properties: safeness properties and liveness properties. For instance a safeness property could state that a ressource is never shared (mutual exclusion) whilst a liveness property could state that every user who wants to own this ressource will eventually own it. Then the flows computation which generates invariants is a suitable tool for verifying safeness properties and the reduction rules which transform the net in an smaller one with the same behaviour as the original one is a suitable tool for verifying the liveness properties.

The first section presents the coloured Petri net model which is built from the ordinary Petri net model by adding colour domains to places and transitions and colour functions to the arcs. The second section presents the principle of the flows computation which is based on finding the kernel of a matrix in a polynomial ring. The third section presents the reduction rules which are defined by structural and functional conditions. In the last section two significant applications are given: a database management and the synchronization of two logical clocks.

1 COLOURED NETS [Jen82]

1.1 Definition

In a coloured net one associates to each place and transition of the net a coloured domain. Each token of a place is coloured by a colour of the place domain and it is necessary to select a colour of a transition to fire it. Then the precondition bound to a place in order to fire a transition is a coloured function which associates to a colour c1 of a transition and a colour c2 of the place, the number of tokens coloured by c2 in the place necessary to fire the transition coloured by c1. In a similar way, the postcondition is a colour function which gives the number of tokens added to the place by the firing of the transition once the tokens of the precondition are removed. The non nul functions label the arcs between places and transitions.

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 PUT to Ω , where Ω is the set of finite and not empty sets. An item of C(s) is called a colour of s and 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 $C(p) \times C(t)$ to **N** (the set of natural integers)
- M the initial marking of the net is a vector of P, where M(p) is a function from C(p) to **N**

Definition 2 The firing rule is defined by:

- A transition t is enabled for a marking M and a colour c_t E C(t) if and only if :

  $\forall p \in P, \forall c \in C(p), M(p)(c) \geq I^-(p,t)(c, c_t)$

- The firing of t for a marking M and a colour c_t E C(t) gives a new marking M' defined by :

  $\forall p \in P, \forall c \in C(p), M'(p)(c) = M(p) - I^-(p,t)(c, c_t) + I^+(p,t)(c, c_t)$

In Petri nets, a flow is a vector over the places such that the sum of the tokens in the places weighted by this vector is let unchanged by the firing of any transition. A basis of flows can be computed by Gauss elimination on the incidence matrix. In order to similarly define the flows on a coloured net, one needs also the definition of the incidence matrix. In a coloured net, a flow is now a vector over the colours of the places.

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 A flow v of a coloured net is a vector over $\cup (p,c)$ such that: $I^t.v = 0$

Notation A flow can be written $v = (v_{p,c})$ with $v_{p,c} \in$ **Q** the set of rationals or $v = \sum v_{p,c} .(p,c)$

1.2 Examples

The definition of coloured nets is quite abstract whilst in practice coloured domains and coloured functions are easily interpretable. The purpose of this section is to present the most frequently used domains and functions and to show how they can simply model a transition of a parallel system.

1.2.1 Colour domains

The objects of a parallel system can be decomposed in classes related to their nature. For instance there is two (or more) class of users of a file: the readers and the writers. In a parallel program one can distinguish the class of the processes runningconcurrently 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. For instance, let **R** be a net with three basic domains: C1 the producers class, C2 the buffers class and C3 the consumers class. A place **p** which models associations between producers and consumers will have for coloured domain $C1 \times C3$.

1.2.2 Colour functions

Let **R** be the net with the classes defined as above, let us suppose the firing of a transition **t** involves a producer **pr**, a consumer **cs** and a buffer **bf**, and that this transition needs one token **<pr ,cs>** in the place **p** defined above. This precondition is specified by a projection.

Definition 1 Let u be an injection from {1...p} to {1...n} and C_i, for $i \in \{1...n\}$, be a basic domain then $Proj_u$ related to the C_i' is defined by :

$Proj_u: (C_1 \times ... \times C_n) \times (C_{u(1)} \times ... \times C_{u(p)}) \rightarrow$ **N**

with $Proj_u(<c_1,...,c_n>, <c'_1,...,c'_p>) = \Pi\, Eq(c_{u(i)},c'_i)$

where the product is over {1...p} and $Eq(x,y) \equiv$ **If** x=y **then** 1 **else** 0

Notation The projection $Proj_u$ will be denoted $<X_{u(1)},\ldots,X_{u(P)}>$ when there is no ambiguity about $C1\times\ldots\times Cn$, We also call identity functions the projections associated to u the identity function of {1,…,n},

Now let us suppose that in **R** a transition **t'** needs for firing that all buffers are empty and that a place **p'** contains a token coloured by any empty buffer, Then this precondition is specified by a constant function sum,

Definition 2 Let C_i be a basic domain, then S_i the constant function sum related to C_i is defined by :

$S_i : C' \times C_i \rightarrow \mathbf{N}$ with $S_i(c', c_i) = 1$

At last let us suppose that the buffers are circulary ordered and that a place **p"** contains a token coloured by the next buffer to be used, If a transition t" models the access to this buffer, then once this transition is fired the token in **p"** is removed and replaced by its successor in the circular order, This postcondition is specified by a successor function,

Definition 3 Let C_i be a basic domain, then $X_i \oplus n$, the n^{th} successor function related to C_i is defined by :

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

with $X_i \oplus n (c,c')$ = **If** c' is the n^{th} successor of c **then** 1 **else** 0

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

Definition 4 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 $<f,g> (c,<c',c">) = f(c,c'),g(c,c")$

The notation is consistent with the notations of projections since for instance the projection $<X_i,X_j>$ is the product of the projections X_i and X_j.This product is associative and thus the product of $f_1,\ldots,f_n$ will be denoted < f1,…,fn >.

Notation Let f and g be two functions, we will denote the composition of f by g, g o f

1.2.3 Sample actions modelled by coloured nets

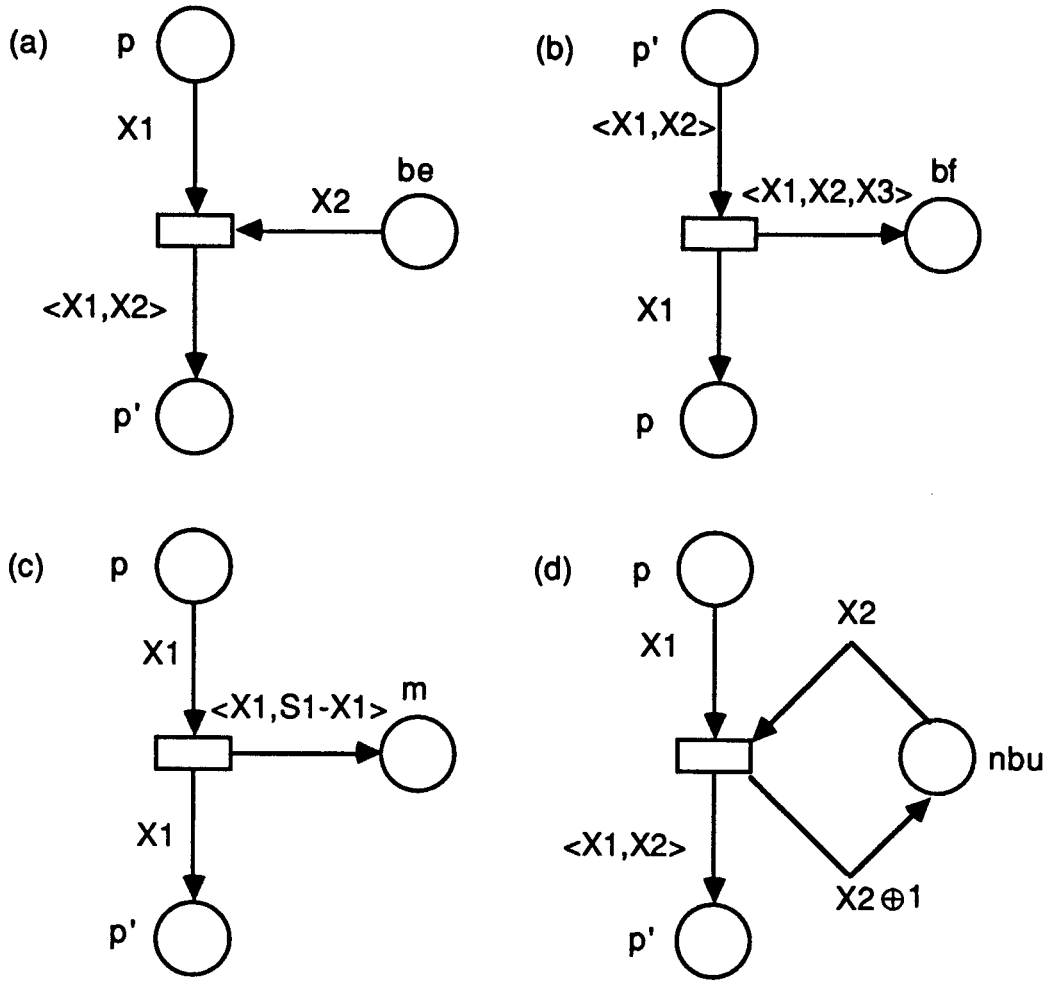
We present now some transitions in order to illustrate the coloured domains and functions and to show how they can be used in modelling:

(a) An idle producer in **p** takes an empty buffer in **be** and becomes an active producer in **p'**, The coloured domain of **p** is C_1 , the coloured domain of **be** is C_2 and the coloured domain of **p** and the transition is $C_1 \times C_2$. X_1 , X_2 are projection functions and $< X_1 , X_2 >$ is an identity function.

(b) An active producer in **p'** sends a full buffer in **bf** with the identity of the consumer and becomes idle, The coloured domain of **bf** and the transition is $C_1\times C_2\times C_3$. Let us notice that $<X_1,X_2>$ is no more an identity function since the domain of the transition is now $C_1\times C_2\times C_3$.

(c) A process in **p** begins some task in **p'** and sends messages to all others processes in **m** with its own identity, The coloured domain of **p** , **p'** and the transition is C_1 and the coloured domain of **m** is $C_1 \times C_1$, The function $< X_1, S_1- X_1>$ is obtained by the product of X_1 and $S_1- X_1$ and this latter function is obtained by substracting X1 to S_1,

(d) An idle producer in **p** takes the next empty buffer to be used in **nbu** and becomes an active producer in **p'**, The next buffer to be used is updated, The coloured domain of **p** is C_1 , the coloured domain of **nbu** is C_2 and the coloured domain of **p'** and the transition is C_1 x C_2 . $X_2\oplus 1$ is the immediate successor function of C_2.



2 COMPUTATION OF FLOWS

2.1 Methods

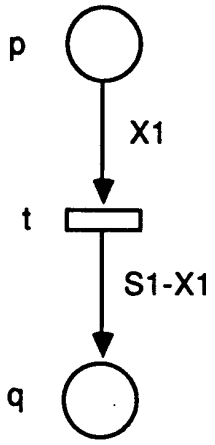
As explained above, the flows space is the kernel of the incidence matrix of the net. In Petri nets a flows basis is easily computed by the elimination of Gauss. But in the coloured nets, this problem is much harder since the any item of the matrix is itself a coloured function (and then can be wieved as a submatrix). So the whole problem is to find a symbolic computation of a basis of flows or at least a generative family of flows taking into account the structure of the coloured functions used in the net. No general algorithm exists now but there are different powerful algorithms developped by the authors depending on the coloured functions used in the net. We simply present the two algorithms that can alternatively be used for almost every realistic coloured net. The first algorithm works with the projections and the sum functions whilst the second one works with the projections and the successor functions. These algorithms are based on the following principles:

- Find one or more linear isomorphisms which transform the incidence matrix into one or more matrices the items of which are in a polynomial ring.
- Then compute a flows basis (or a generative family) of the kernel of the transformed matrix.
- At last apply the inverse isomorphisms to obtain the family of the flows.

The main difference between the two algorithms is that in the first one the isomorphisms are complex and the computation is simple and vice versa for the second one.

2.2 <u>Computation with the projectlons and the sum functions</u> [Had87b]

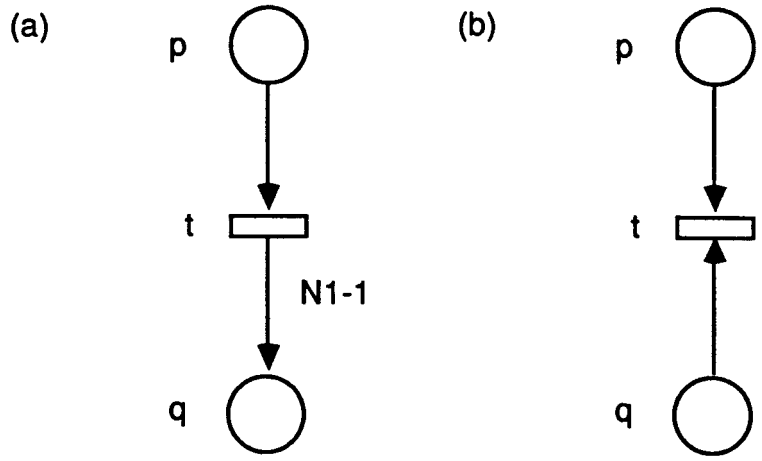
We begin with a simple example in order to show how the algorithm works.



In this net a flow v can be written: $\sum v_{p,c}.(p,c) + \sum v_{q,c}.(q,c)$. Now let us suppose that we are interested by two kind of flows:

- Homogeneous flows which count the number of tokens in a place independently from their colour. These flows can be written : $v_p. \sum (p,c) + v_q. \sum (q,c)$
- Differential flows which count the difference between the tokens of two different colours of the domain C_1. These flows can be written: $v_p .[(p,c)-(p,c')] + v_q .[(q,c)-(q,c')]$

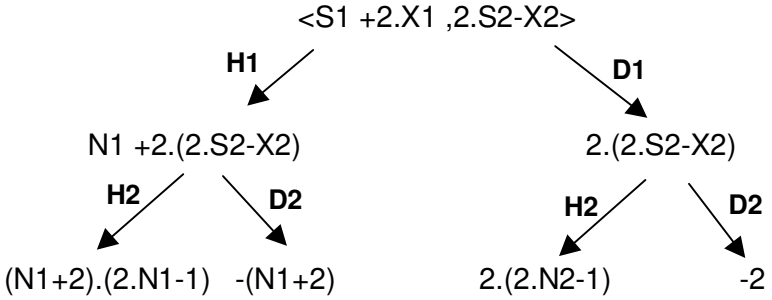
Then it can be shown that the computing of the homogeneous flows is equivalent to the computing of the flows in the Petri net (a) and that the computing of the differential flows is equivalent to the computing of the flows of the Petri net (b).



The nets (a) and (b) have be obtained by the following symbolic transformations.
<u>Homogeneisation</u>: In (a) the function S_1 becomes the scalar N_1 (the cardinality of C_1) and the functions X_1 becomes the scalar 1.
<u>Differentiation</u>: In (b) the function S vanishes and the function X_1 becomes the scalar 1.

Let us suppose now that a net has two basic domains, then once we apply these transformations on a basic domain we obtain two coloured nets with only a basic domain and we apply again these transformations on the remainder basic domain obtaining now four Petri nets. This iterative process can be wieved as the building of a tree where the root is the initial net, the nodes are the successive coloured nets and the leaves are the final Petri nets. The next figure presents the successive transformations of a coloured function.

H : Homogenisation , D : Differentiation



Now we express the theoretical results and the algorithm which is based on these results:

Theorem Let R be a coloured net, C_i a basic domain of R, $H_i(R)$ the coloured net obtained by homogeneisation of R related to C_i and $D_i(R)$ the coloured net obtained by differentiation of R related to C_i, then:
(1) The homogeneous flows space of R is isomorphic to the flows space of $H_i(R)$
(2) The differential flows space of R is isomorphic to the flows space of $D_i(R)$
(3) The flows space of R is a direct sum of the homogeneous flows space of R and the differential flows space of R

The proposition (3) is fundamental since it implies that any flow of R is an unique linear sum of homogeneous flows and differential flows. Then we can deduce:

Corollary The union of a homogeneous flows basis and a differential flows basis is a general flows basis.

Algorithm

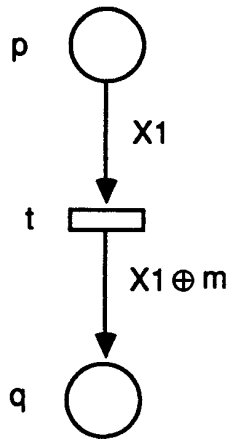
```
Calcul (R:coloured net; Var B:basis of flows);
Begin
   If R is a Petri net Then
      Extended_Gauss(R;B)
   Else
      Begin
         Calcul (Hi(R),Bh);
         Calcul (Di(R),Bd);
         B:= Ish (Bh ) ∪ Isd (Bd);
      End;
End;
```

Remarks
- Is_h and IS_d are the isomorphisms stated by the theorem.
- When one obtains a Petri net by succesive transformations, the items of the incidence matrix are not in Z but in the polynomial ring $\mathbf{Z}[N_1,...,N_n]$ where N_i are the variable cardinalities of the basic domains. Happily this ring is entire (no divisors of 0) and commutative, then it can be embedded in its fraction field and an extended Gauss elimination may be applied on it

2.3 Computation with proiections and successor functions [Cou88]

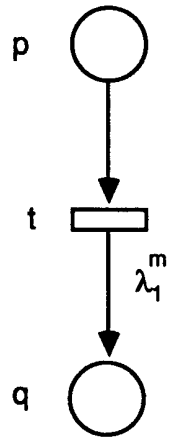
We begin with a simple example. Looking at the coloured net above, it is easy to show that: $\{(p,c') + (q,c) \ / c \in C_1$ and c' is the m^{th} predecessor of c$\}$ is a flows basis.

Let us suppose now that we choose some colour c in C1 and we denote it 1 and that any colour c' is denoted λ_1^i if c' is the i^{th} predecessor of c and (p,c') is denoted $\lambda_1^i.p$. Then the basis can be rewritten $\{ \lambda_1^{i+m} .p + \lambda_1^i .q \ I \ i=0,...,N1-1\}$ with the convention that for any i $\lambda_1^{i+N1} = \lambda_1^i$. At last the basis can be rewritten with respect to the polynomial product: $\{ \lambda_1^i . (\lambda_1^m .p + q) \ I \ i=0,...,N1-1\}$



Now looking at the following Petri net where the valuations are in the polynomial ring $Z[\lambda_1]$ quotiented by the ideal $\{P(\lambda_1).(\lambda_1^{N1} -1)\}$, it is easy to show that $\{\lambda_1^m .p + q\}$ is a flows basis.

Moreover this net have been obtained by the symbolic transformation:
Any $X_i \oplus m$ becomes λ_1^m (and thus any X_i becomes 1).



We will denote the net obtained from a coloured net R by this transformation $\lambda(R)$ and now we express the theoretical results:

Theorem Let R be a coloured net and $\lambda(R)$ the Petri net obtained by the preceeding transformation, then (with the convention on the coloured places) the three propositions are equivalent:

(1) $\sum v_{p,j1,...,jn} .(\lambda 1^{j1}... \lambda n^{jn}. p)$ is a flow of R

(2) $\forall$ k1,...,kn, $\sum v_{p,j1,...,jn} .(\lambda 1^{j1+k1}... \lambda n^{jn+kn}. p)$is a flow of R

(3) $\sum v_{p,j1,...,jn}. (\lambda 1^{j1}... \lambda n^{jn}. p)$ is a flow of $\lambda(R)$

Remark We recall that in (1) and (2), $(\lambda 1^{j1}... \lambda n^{jn}. p)$ and $(\lambda 1^{j1+k1}... \lambda n^{jn+kn}. p)$ denote coloured places whilst in (3) , $(\lambda 1^{j1}... \lambda n^{jn}. p)$ denotes the place p weighted by the polynom $\lambda 1^{j1}... \lambda n^{jn}$.

The equivalence between (2) and (3) is very important since it enables the computation of a generative family of flows:

Corollary Let F be a generative family of flows of $\lambda(R)$, then F' the family of flows of R obtained by all the transformations defined in the proposition (2) is generative.

Then one computes a generative family of the flows on the polynomial ring $\mathbf{Z}[\lambda_1,...,\lambda_n]/\ \mathbf{I}$ where **I** is the ideal generated by the polynoms $\{\lambda^{Ni} -1\}$. Let us only say that since this ring is no more entire, the computation is much harder that for the case 2.3 and for instance the Gauss elimination can not be used. Some efficient methods can be found in [Bur83] , [Laz81] , [Ful55].

3 REDUCTION OF NETS [Had87b]

For the reductions we need some properties on the coloured functions.

Definition Let f be a coloured function from C x C' to N , then
- f is unitary if and only if $\forall$ c,c' f(c,c') = 0 or f(c,c') = 1
- f is quasi injective if and only if $\forall$ c,c',c" f(c,c') $\neq$ 0 and f(c,c") $\neq$ 0 $\Rightarrow$ c' = c"

In a modelling the functions are almost ever unitary since generally the firing of a transition involves at most one token of each different colour for any place. The quasi injectivity is a significant property since it implies that a colour in a place will be used by at most one coloured firing of a transition if a quasi injective function valuates the arc between the place and the transition.

3.1 Coloured implicit place simplification

This reductions deletes a place which never disables the firing of a transition and the marking of which can be computed from the marking of the other places. In contrast to the other reductions that we will present here, the implicit place is based on a algebraic property (existence of a particular flow). Then the application of this reduction implies the existence of flows computation for coloured nets which already justifies the Interest of section 2. Since the computation of flows for coloured nets is much more complex than in ordinary Petri nets, this reduction which could be done manually in Petri nets needs now the help of a good flows computation.

Definition 1 Let (R,Mo) be a coloured net, a place p is implicit if and only if :

(1) $\forall$ c $\in$ C(p)

There is a flow f_c the support of which is $\{(p,c)\} \cup P'$ where $P' = \{(q_1,c_1),...,(q_k,c_k)\}$

$$fc = a_{pc}.(p,c) - \sum_{i=1,...,k}. ai.(q_i,c_i) \text{ with } a_{pc},a_i \in \mathbf{N} \text{ and } \forall\ c',\ (p,c') \notin P'$$

(2) $\forall$ t $\in$ T, $\forall c_t \in C(t)$

$$a_{pc}.I^-(p,t)(c,c_t) - \sum ai.I^-.(q_i,t)(c_i,c_t) \leq a_{pc}.Mo(p,c) - \sum a_i.Mo(q_i,c_i)$$

Interpretation An implicit place will never disable the firing of a transition since initially it does not disable it because of (2) and this condition is reproductible for all the reachable markings because of (1).

As in ordinary Petri nets, the transformation deletes the implicit place and its arcs.

Definition 2 The reduced net (R_r,Mo_r) obtained from the net (R,Mo) by simplification of the implicit place p is defined by :
- $P_r = P\backslash\{p\}$
- $T_r = T$
- $\forall$ t $\in T_r$ $\forall$ p' $\in P_r$, $C_r(p') = C(p')$ and $C_r(t) = C(t)$
- $\forall t \in T_r$, $\forall$ p' $\in P_r$, $I_r^-(p',t) = I^-(p',t)$ and $I_r^+(p',t) = I^+(p',t)$
- $\forall$ p' $\in P_r$ $Mo_r(p') = Mo(p')$

Theorem Let (R_r,Mor) be a reduced net obtained from the net (R,Mo) by simplification of the implicit place, then:

(R,Mo) is bounded $\Leftrightarrow$ (R_r,Mo_r) is bounded — (R,Mo) is safe $\Rightarrow$ $(R_r,Mo_r))$ is safe
(R,Mo) has an invariant $\Leftrightarrow$ (R_r,Mo_r) has an invariant — (R,Mo) is quasi-live $\Leftrightarrow$ (R_r,Mo_r) is quasi-live
(R,Mo) is live $\Leftrightarrow$ (R_r,Mo_r) is live — (R,Mo) is pseudo-live $\Leftrightarrow$ (R_r,Mo_r) is pseudo-live
(R,Mo) has a normal end $\Leftrightarrow$ (R_r,Mo_r) has a normal end
(R,Mo) has an home state $\Leftrightarrow$ (R_r,Mo_r) has an home state
(R,Mo) has an unavoidable state $\Leftrightarrow$ (R_r,Mo_r) has an unavoidable state
(R,Mo) verifies abstraction properties $\Leftrightarrow$ (R_r,Mo_r) verifies abstraction properties [And81]

3.2 Coloured Pre agglomeration

All the agglomerations are based on the following idea: given a place p, its input transition set H and its output transitions set F, one wants to disable the intermediate states with p marked. Then the agglomeration merges the two transitions set in order to simulate the firing of any transition of H followed by the firing of any transition of F.

In the pre agglomeration the set H is reduced to one transition. The principle of pre agglomeration IS the following: in every sequence of firings with an occurence of h followed later by an occurence of a transition f of F, one can postpone the firing of h and "merge" it with the firing of f.

Definition 1 Let (R,Mo) be a marked Petri net, a subset of transitions F is pre agglomerable if and only if there is a place p and a transition h E F such that the following conditions are verified:

(1) $\forall t \neq h$, $I^+(p,t) = 0$ and $\forall t \notin F$, $I^-(p,t) = 0$
$C(p) = C(h)$ and $I^+(p,h)$ is the identity function
$\forall f \in F$, $I^-(p,f) * 0$ and $I^-(p,f)$ is an unitary function
$Mo(p) = 0$
{The single input transition of p is h and the output transitions of p are F }
{P is unmarked}
{The arc from h to p is valuated by the identity function}
{Every arc coming from p is valuated by an unitary function}

(2) $\forall p' \neq p$, $I^+(p',h) = 0$ { The single output place of h is p }

(3) $\exists p' \in P$,such that $I^-(p',h) \neq 0$ {h has an input place}

(4) $\forall p' \in P \; \forall t \in T\backslash\{h\}$,
$I^-(p',h) \neq 0 \Rightarrow I^-(p',t) = 0$ and $I^-(p',h)$ is a quasi injective function
{ h does not share its input places}
{ Every arc going to h is valuated by a quasi injective function}

In the transformation, the place p and the transition h disappear. The input arcs of h are transferred on each transition f of F but the function valuating these arcs are composed by the function valuating the arc between p and f.

Definition 2 The reduced net (R_r,Mo_r) obtained from the net (R,Mo) by a coloured preagglomeration of hand F is defined by :

- $P_r = P \setminus \{p\}$,
- $T_r = T \setminus \{h\}$
- $\forall t \in T_r$, $\forall p' \in P_r$, $C_r(t) = C(t)$ and $C_r(p') = C(p')$
- $\forall t \in T_r$, $\forall p' \in P_r$, $I_r^+(p',t) = I^+(p',t)$

Let $P_h = \{ p' \in P_r \mid I^-(p',h) \neq 0 \}$

- $\forall t \in T$, $\forall p' \notin Ph$, $I_r^-(p',t) = I^-(p',t)$
- $\forall f \in F$, $\forall p' \in P_h$, $I_r^-(p',f) = I^-(p',h) \circ I^-(p,f)$
- $\forall p' \in P_r$, $Mo_r(p') = Mo(p')$

Theorem Let (R_r,Mo_r) be a reduced net obtained from the net (R,Mo) by a coloured pre agglomeration, then:

(R,Mo) is bounded $\Leftrightarrow$ (R_r,Mo_r) is bounded — (R,Mo) is safe $\Rightarrow$ (R_r,Mo_r) is safe
(R,Mo) has an invariant $\Leftrightarrow$ (R_r,Mo_r) has an invariant — (R,Mo) is quasi-live $\Leftrightarrow$ (R_r,Mo_r) is quasi-live
(R,Mo) is live $\Leftrightarrow$ (R_r,Mo_r) is live — (R,Mo) is pseudo-live $\Leftrightarrow$ (R_r,Mo_r) is pseudo-live
(R,Mo) has a normal end $\Leftrightarrow$ (R_r,Mo_r) has a normal end
(R,Mo) has an home state $\Leftrightarrow$ (R_r,Mo_r) has an home state
(R,Mo) has an unavoidable state $\Leftrightarrow$ (R_r,Mo_r) has an unavoidable state
(R,Mo) verifies abstraction properties $\Leftrightarrow$ (R_r,Mo_r) verifies abstraction properties [And81]

3.3 Coloured post agglomeration with multiple outputs

The principle of post agglomeration is the following: in every sequence of firings with an occurence of a transition h of H followed later by an occurence of a transition f of F, one can fire f immediately after the firing of h. As we enable the multiple output transitions of p, we require the functions valuating the arcs to p to be projections in order to ensure that the token game around p can be simulated by the reduced net.

Definition 1 Let (R,Mo) be a coloured Petri net, a subset of transitions F is post agglomerable if and only if there is a place p and a subset of transitions H with $H \cap F = \varnothing$ such that the following conditions are verified:

(1) $\forall t \notin H$, $I^+(p,t) = 0$ and $\forall t \in F$, $I^-(p,t) = 0$
$\forall h \in H$, $\exists C_h$ such that $C(h) = C(p) \times C_h$
and $I^+(p,h)$ is the projection of C(h) over C(p)
$\forall f \in F$, $C(f) = C(p)$ and $I^-(p,f)$ is the identity function
$Mo(p) = 0$
{All the arcs going to p are valuated by projections}
{All the arcs coming from p are valuated by identities}
{p is unmarked}

(2) $\forall c \in C(p)$, $\exists f \in F$, $\exists p' \in P$, $\exists c' \in C(p)$, such that $I^+(p',f)(c,c') \neq 0$
{ For each colour c of C(p), there is a transition of F the c-firing of which produces some tokens}

(3) $\forall f \in F$, $\forall p' \neq p$, $I^-(p',f) = 0$ {The single input place of every transition of F is p}

In the post agglomeration with multiple outputs, the place p disappears and one substitutes the "product" transitions of $H \times F$ to the transitions of H and F. The arcs of these transitions are obtained by the union of the arcs of H and F but where the output arcs of F are composed by the corresponding projection.

Definition 2 The reduced net (R_r,Mo_r) obtained from the net (R,Mo) by a coloured post agglomeration of H and F is defined by :

- $P_r = P \setminus \{p\}$
- $T = T \cup (H \times F) \setminus (H \cup F)$
- $\forall f \in F$, $\forall h \in H$, one notes hf the transition (h,f) of $H \times F$
- $\forall t \in T_r \setminus (H \times F)$, $\forall p' \in P_r$, $C_r(t) = C(t)$ et $C_r(p') = C(p')$
  $\forall f \in F$, $\forall h \in H$, $C_r(hf) = C(h)$
- $\forall t \in T_r \setminus (H \times F)$, $\forall p' \in P_r$, $I_r^-(p',t) = I^-(p',t)$ and $I_r^+(p',t) = I^+(p',t)$
- $\forall h \in H$, $\forall f \in F$,
  $\forall p' \in P_r$, $I_r^-(p',hf) = I^-(p',h)$ and $I_r^+(p',hf) = I^+(p',h) + I^+(p',f) \circ I^+(p,h)$
- $\forall p' \in P_r$, $Mo_r(p') = Mo(p')$

Theorem Let (R_r,Mo_r) be a reduced net obtained from the net (R,Mo) by a coloured post agglomeration with multiple outputs, then:

(R,Mo) is bounded $\Leftrightarrow$ (R_r,Mo_r) is bounded — (R,Mo) is safe $\Rightarrow$ (R_r,Mo_r) is safe
(R,Mo) has an invariant $\Leftrightarrow$ (R_r,Mo_r) has an invariant — (R,Mo) is quasi-live $\Leftrightarrow$ (R_r,Mo_r) is quasi-live
(R,Mo) is live $\Leftrightarrow$ (R_r,Mo_r) is live — (R,Mo) is pseudo-live $\Leftrightarrow$ (R_r,Mo_r) is pseudo-live
(R,Mo) has a normal end $\Leftrightarrow$ (R_r,Mo_r) has a normal end
(R,Mo) has an home state $\Leftrightarrow$ (R_r,Mo_r) has an home state
(R,Mo) has an unavoidable state $\Leftrightarrow$ (R_r,Mo_r) has an unavoidable state
(R,Mo) verifies abstraction properties $\Leftrightarrow$ (R_r,Mo_r) verifies abstraction properties [And81]

3.4 Coloured post agglomeration with a single output

In contrast to the post agglomeration with multiple outputs, here F is reduced to a single transition. Then the colour function which valuates an arc from a transition of H to the place p is less constrained: it must be an unitary function (a very weak condition). There is no more constraint on the colour domain of the transitions of H. The other conditions are the same as the post agglomeration with multiple outputs.

Definition 1 Let (R,Mo) be a coloured Petri net, a transitions f is post agglomerable if and only if there is a place p and a subset of transitions H with $H \cap \{f\} = \emptyset$ such that the following conditions are verified:

(1) $\forall t \notin H$, $I^+(p,t) = 0$ and $\forall t \neq f$, $I^-(p,t) = 0$
$I^+(p,h) \neq 0$ and $I^+(p,h)$ is an unitary function
$C(f) = C(p)$ and $I^-(p,f)$ is the identity function
$Mo(p) = 0$
{All the arcs going to p are valuated by unitary functions}
{All the arcs coming from p are valuated by identities}
{p is unmarked}

(2) $\forall c \in C(p)$, $\exists p' \in P$, $\exists c' \in C(p')$,such that $I^+(p',f)(c',c) \neq 0$
{ For each colour c of C(p), the the c-firing of f produces some tokens}

(3) $\forall f \in F$, $\forall p' \neq p$, $I^-(p',f) = 0$ {The single input place of every transition of F is P }

In the post agglomeration with a single output, the place p and the transition f disappear and one substitutes the "product" transitions of $H \times \{f\}$ to the transitions of H. The arcs of these transitions are obtained by the union of the arcs of H and {f) but where the output arcs of f are composed by the corresponding unitary function of the transition of H.

Definition 2 The reduced net (R_r,Mo_r) obtained from the net (R,Mo) by a coloured post agglomeration of H and f is defined by :

- $P_r = P \setminus \{p\}$
- $T = T \setminus \{f\}$
- $\forall t \in T_r \setminus H$, $\forall p' \in P_r$, $C_r(t) = C(t)$ and $C_r(p') = C(p')$
- $\forall t \in T_r \setminus H$, $\forall p' \in P_r$, $I_r^-(p',t) = I^-(p',t)$ and $I_r^+(p',t) = I^+(p',t)$
- $\forall h \in H$, $\forall p' \in P_r$, $I_r^-(p',h) = I^-(p',h)$ and $I_r^+(p',h) = I^+(p',h) + I^+(p',f) \circ I^+(p,h)$
- $\forall p' \in P_r$, $Mo_r(p') = Mo(p')$

Theorem Let (R_r, Mo_r) be a reduced net obtained from the net (R,Mo) by post agglomeration with a single ouput, then:

(R,Mo) is bounded $\Leftrightarrow (R_r,Mo_r)$ is bounded — (R,Mo) is safe $\Rightarrow (R_r,Mo_r)$ is safe
(R,Mo) has an invariant $\Leftrightarrow (R_r,Mo_r)$ has an invariant — (R,Mo) is quasi-live $\Leftrightarrow (R_r,Mo_r)$ is quasi-live
(R,Mo) is live $\Leftrightarrow (R_r,Mo_r)$ is live — (R,Mo) is pseudo-live $\Leftrightarrow (R_r,Mo_r)$ is pseudo-live
(R,Mo) has a normal end $\Leftrightarrow (R_r,Mo_r)$ has a normal end
(R,Mo) has an home state $\Leftrightarrow (R_r,Mo_r)$ has an home state
(R,Mo) has an unavoidable state $\Leftrightarrow (R_r,Mo_r)$ has an unavoidable state
(R,Mo) verifies abstraction properties $\Leftrightarrow (R_r,Mo_r)$ verifies abstraction properties [And81]

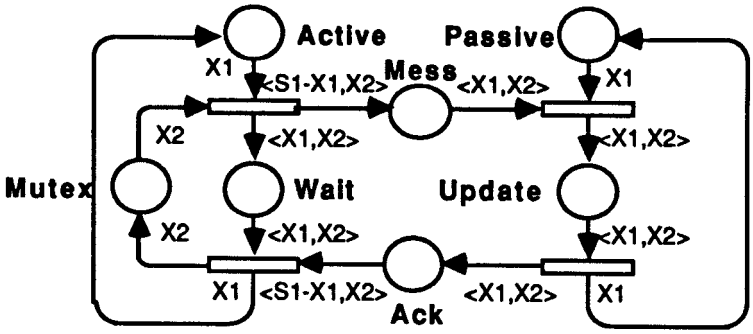
4 APPLICATION

4.1 Database management

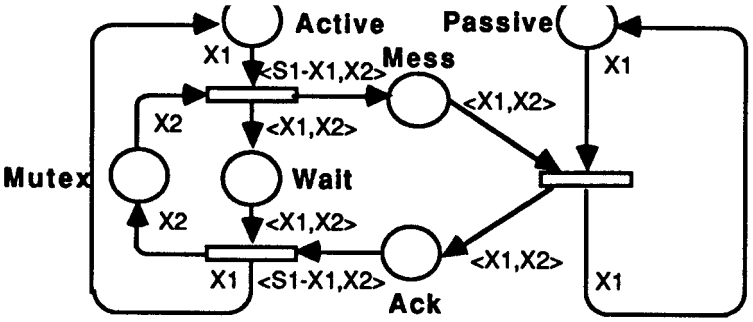
We present now the modelling of a data base management with multiple copies.This modelling is an improved version of those of [Jen81]. Each site has two processes, an active one and a passive one. The access grant of a file of the data base is centralized and submitted to the mutual exclusion. In order to modify a file the active process of a site must get its grant and once it has modified the file, it sends messages to the others sites with the updated file. Then the passive processes update their own data base and send an acknowledgment. Once the active process has received all the acknowledgments, it releases the grant. Simultaneous accesses to different files are allowed.

In the net, an active process must get in Mutex the single token coloured by the file it wants to access. The messages are composed by the name of the receiver followed by the name of the file. The acknowledgments are composed by the name of the sender followed by the name of the file. Accessing and modifying a file is modelled by a transition (atomic step) whilst the updating of the passive process is modelled by a place (divisible step). Initially there is a token per site in Active and Passive and a token per file in Mutex.

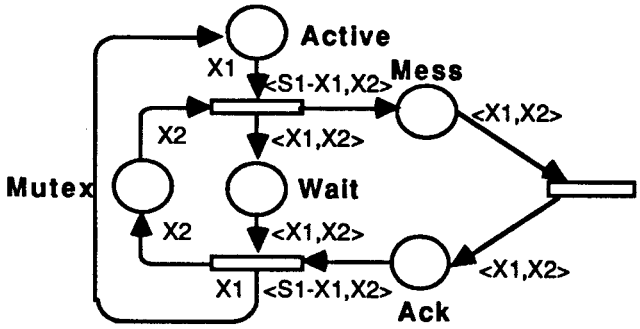
C1 = {Sites} , C2 = {Files} ,C(Active) = C(Passive) = C1 , C(Mutex) = C2
C(Wait) = C(Update) = C(Mess) = C(Ack) = $C1 \times C2$
For every transition t , $C(t) = C1 \times C2$
The colour functions are defined as in the preceeding examples.



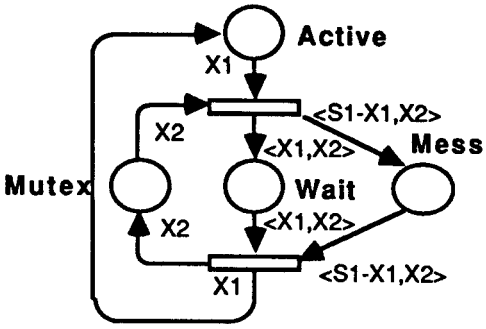
Post agglomeration with a single output around Update



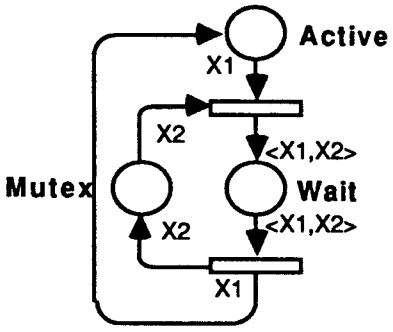
Simplification of the implicit place Passive



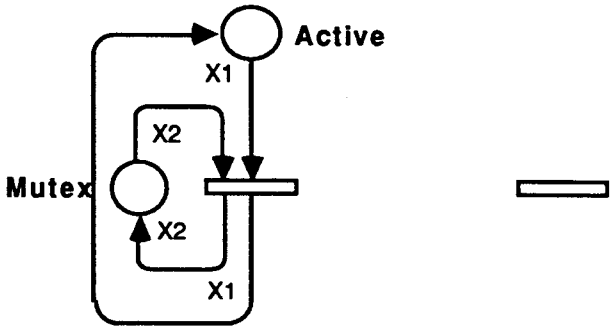
Pre agglomeration around Ack



Simplification of the implicit place Mess



Post agglomeration with a single output around Wait



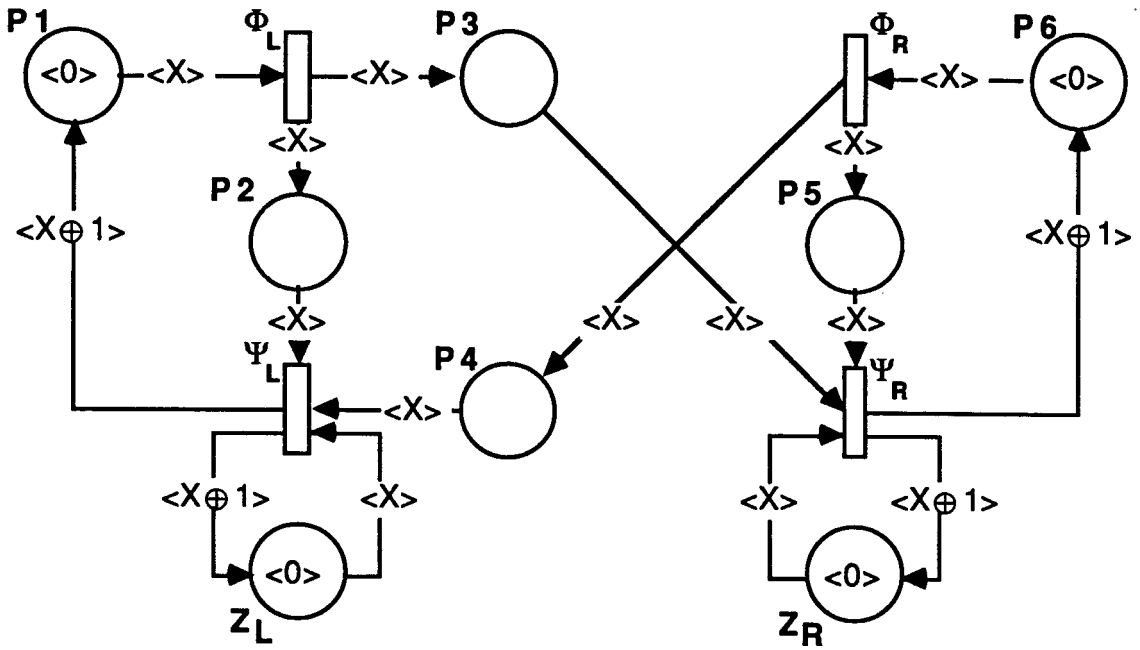
Simplification of the implicit places Mutex and Active

Comments Thus we have shown that the net has all the good behavioural properties since it is reductible to a simple transition (which is a "perfect" net). All the reductions can manually be verified except the simplification of the implicit place Mess which needs to find the flow stated in the definition 3.1.1. By application of the algorithm given in 2.2 we find the adequate flow:
$\forall x \in C1$, $Mess(x) - \sum_{x' \neq x} Wait(x') = 0$

4.2 Synchronlzed clocks

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 was done In [Gen821. 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.

The initial marking, $Mo(Z_L)= Mo(P1)= MQ(Z_R)= Mo(P6)= <0>$, $Mo(Ps)= 0$ for s=2,3,4,5, indicates that the position of Z_L and Z_R is 0 and Land R are ready to send the message <0> to each other. A partner say L, may increase its counter Z_L by firing of ψ_L if P2, P4 and Z_L are carrying the same equal token. We want to show that the absolute difference of counter positions (modulo N) is at most one. Using the method 2.3 we do not need any examination of the accessibilty set to prove it.



We give now the generative family of flows computed by the algorithm preceeded by their interpretation. On the left the flows are the ones computed on the polynomial matrix whilst on the right we present the final flows obtained by application of the isomorphisms.

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

$(1+ \lambda+...+ \lambda^{N-1})$ ZR $ZR(0) + ...+ ZR(N\Theta1)-1$

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

$P1 + P2 - Z_L$ $\forall x \in C, P1(x) + P2(x) - ZL(x)$

$P5 + P6 - Z_R$ $\forall x \in C, P5(x) + P6(x) - ZR(x)$

The next two invariants proves the synchronization of the counters:

$Z_L - \lambda Z_R + (\lambda-1)(P4+P6)$ $\forall x \in C, Z_L(x) + P4(x\Theta1) + P6(x\Theta1) - ZR(x\Theta1) + P4(x) + P6(x)$

$P1 + P3 + \lambda(P4+P6) - (\lambda+1)Z_R$ $\forall x \in C, P1(x) + P3(x) + P4(x\Theta1) + P6(x\Theta1) - ZR(x) + ZR(x\Theta1)$

If we add these last invariants for the respective colours x and $x \oplus 1$, we obtain:

$\forall x \in C, Z_L(x) + P1(x\oplus1) + P3(x\oplus1) + P4(x\Theta1) + P6(x\Theta1) = Z_R(x\Theta1) + Z_R(x) + Z_R(x\oplus1)$

From the first invariant we deduce the right term (and thus the left term) of this equality is bounded by 1. Let us now apply this equality for the current value c of Z_L then we get:

$Z_L(c) = Z_R(c\Theta1) + Z_R(c) + Z_R(c \oplus1)=1$

Thus $Z_R(c\Theta1)=1$ or $Z_R(c) = 1$ or $Z_R(c\oplus1) =1$ which is exactly the searched property

CONCLUSION

In this paper we have presented the coloured nets and shown how they can be efficiently used in the modelling of the parallel systems with the help of the coloured domains and functions. We have also caracterized the coloured domains and functions which are the most frequently used in practice.

Then we have developed our theory of flows calculus for coloured nets which is based on two principles: first we show that the flows space of a coloured net is isomorphic to the kernel of a polynomial matrix and then we compute this kernel by extended Gauss elimination in the simplest cases and by more accurate methods depending on the polynomial ring.

We have also extended the reduction rules of Petri nets [Ber83] to the coloured nets by adding to the structural conditions the adequate functionnal conditions. This extension leads to the definition of four rules the application field of which is very large.

At last by combining the tools developed above we have verified the safeness and the liveness properties of two applications: a database management and the synchronization of two logical clocks.

REFERENCES

[And81] C.ANDRE:"Systèmes a évolutions parallèles : Modélisation par réseaux de Petri a capacités et analyse par abstraction. Thèse d'état. Université de Nice. (1981)

[Ber83] G.BERTHELOT:"Transformation et analyse de réseaux de Petri, applications aux protocoles". Thèse d'état. Université P. et M. Curie. Paris (1983) [Bra83] G.W. BRAMS: "Réseauxde Petri. Théorie et pratique". Masson éditeur, 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.

[Col86] J M COLOM, J MARTINEZ, M SILVA: Packages for validating discrete production systems modeled with Petri nets. IMACS-IFAC Symposium, Lille, France (1986)

[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 ideal 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 n066, G.Rozenberg ed., Springer Verlag, 1983, pp 98-111.

[Had86] S. HADDAD, C. GIRAUL T: "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 réseaux colorés", Deuxième colloque C3 Angoulême, France,1987.

[Had87b] S. HADDAD: "Une catégorie régulière de réseau de Petri de haut niveau: définition, propriétés et réductions. Application a la validation de systèmes distribués", Thèse de l'Université Pierre et Marie Curie, Paris, juin1987.

[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, september1982, pp 261-276.

[Lau85] K. LAUTENBACH, A. PAGNONI: "Invariance and duality in predicate transition nets and in coloured nets", Arbeitspapiere der G.M.D, n0132.

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

[Mem83] G. MEMMI: "Méthodes d'analyse de réseaux de Petri, réseaux a files et applications aux systèmes temps réel", Thèse d'état,Université 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 n01, 1985, pp 113-126.

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