Research on the Particular Subclass of a Class Coloured Petri Nets

Mihaita Dragan

Abstract—This article presents a particular subclass of class coloured Petri nets, that’s abbreviated as HLP (high-level particular Petri net). In this subclass it’s possible to resolve the system of equations that’s providing linear invariants of the subclass. The paperwork presents an efficient method for establishing such invariants.

Keywords—Multi-set, ω-Multi-set, multiextension, coloured Petri net, action, linear invariant.

I. INTRODUCTION

Because even for simple systems, modeling by place transition nets (PT-nets) is sometimes very complex, it occurs the necessity of transferring the information from location to the location’s markings and to the network’s transitions.

We’ll simplify the net’s structure if we associate color sets to their markings and transitions, and if we define functions between them, which we associate to the net’s arcs.

It defines a new class of Petri nets named colored Petri nets. For example, in order to modelate a FIFO queue with n elements and one that modelates types of messages through a PTWnet, are necessary n(p+1) places: one that modelates the queue with the n-elements and p types of messages.

II. THEORETICAL CONSIDERATIONS

To be able to introduce the class of colored Petri nets, we need the following notions:

Definition 1.1 [2, 5] Be A a finite set, A={a1, a2, ..., an}.

a) It’s called multiset over A any application m: A → Z and unnegative multiset of A, any application m+: A → N.

b) It’s called a characteristic polynomial of set A noted Π(A), the multiset m, with the property m+(x)=1, ∀x ∈ A.

We’ll note the multisets as formal sums m=∑ni=1 m(a_i) a_i, or, wider, m=∑a∈A m(a) a, with a ∈A.

By convention, we’ll skip the coefficients m(a_i) = 1 and multiset’s terms for which m(a_i) = 0.

Also, we’ll note with 0 the multiset for which 0(a_i) =0, ∀i=1,..,n.

With these notations, the characteristic polynomial of set A is Π(A)=∑x∈A x.

The multisets can be seen as sets where are allowed multiple occurrences of the elements.

We’ll note with MS(A) the set of all multisets over A and with MS+(A) the set of all unnegative multisets over A.

Example 1.3

Be A={a, b, c, d}. Unnegative multiset m+: A→N given by:

1) m+(a)=2, m+(b)=1, m+(c)=1, m+(d)=0 will be in the form m+=2a+b+c.

It can be seen as the object {a, b, a, c} where the a element appers twice.

Note that | m+ | =4 ⋅ d ∈ m+.

Definition 1.4 [2, 5]

Be u, v ∈ MS(A)+ vectors, having elements multisets over A, and C a matrix by dimension (n x p) with elements from MS(A).

Generalize the vector’s scalar product and the product of a matrix and a vector, when they have a meaning, by:

u′ ⊗ v=∑ni=1 u_i ⊗ v_i iar (u′ ⊗ C_j) =∑ni=1 u_i ⊗ C_j_i, j=1...p ,

where ( ⊗ ) is the function composition.

If C is a matrix with elements from MS(A) by order (n x p) we’ll note | C | the matrix resulted by replacing any multiset c_j_i with its length | c_j_i |.

Definition 1.5 [2] Be A a nonempty set. It’s called ω -multiset over A any application m: A→Nω
An $\omega$-multiset will be represented such as a multiset by a formal sum $m = \sum_{a \in A} m(a)a$ where $m(a) \in N$, is the number of occurrences of element $a \in A$.

If $m(a) = \omega$, the number of element’s occurrences can be any big and is not exactly known.

We’ll note $MS_{\omega}(A)$ the set of all finite $\omega$-multisets over $A$. If the $m \omega$-multiset has the support $\{a \in A | m(a) \neq 0\}$ finite, than we see that $m$ is finite. Operating with $\omega$-multisets is defined as for the multisets:

$$m_1 + m_2 = \sum_{a \in A} (m_1(a) + m_2(a))a, \quad m_1, m_2 \in MS_\omega(A)$$

$$(k \times m) = \sum_{a \in A} (km(a))a, \quad k \in N, \quad m \in MS_\omega(A)$$

$$m_1 \geq m_2 \iff \forall a \in A, m_1(a) \geq m_2(a) \text{ and } m_1 > m_2 \iff \exists a \in A \text{ so that } m_1(a) > m_2(a)$$

$$m_1 - m_2 = \sum_{a \in A} (m_1(a) - m_2(a))a \text{ defined for } m_1, m_2 \in MS_\omega, \quad \mu_1 \geq \mu_2.$$  

Definition 1.6 [2]

Be $A, B$ nonempty sets and $[A \to B]_L$ the set of linear applications from $A$ to $B$.

Be $F: A \to MS(B)$:

a) We call multiset-extension of $F$ the only linear application $F^* \in [MS(A) \to MS(B)]_L$ defined by:

$$\forall m \in MS(A), \quad F^*(m) = \sum_{a \in A} (m(a) \bullet F(a)) \text{ with } a \in A.$$  

b) We call $\omega$-multiset-extension of a function $F: A \to MS(B)$ the only linear application $F^*: MS_\omega(A) \to MS_\omega(B)$, defined by:

$$\forall m \in MS_\omega(A), \quad F^*(m) = \sum_{a \in A} (m(a) \bullet F(a)) \text{ with } a \in A.$$  

Definition 1.7[3, 7, 8]

It’s called colored Petri net or high-level Petri net (HL-net or CPN) the sixtuplu $\Sigma = (S, T, A, \text{ Pre, Post, c, } \mu_0)$, where:

a) $S$ and $T$ meet the requirements from PTW-nets.

b) $A$ is a finite family of nonempty sets of network’s colors.

c) $C$ is the coloring function of net $C: S \cup T \to A$.

d) Pre and Post are functions defined on $(S \times T)$ so that for $(s, t) \in (S \times T)$ Pre(s, t), Post(s, t), Pre(t, s) linear functions from:

$$[MS_\omega(C(t)) \to MS_\omega(C(s))]_{\Sigma}, \text{ and can be represented (for finite nets) by matrix of dimension } |S| \times |T|.$$  

For any two colors, $c' \in C(t)$ and $c' \in C(s)$, the elements $\text{Pre}(s, t)(c')(c''), \text{Post}(s, t)(c')(c'') \in N$, and represent the arc’s weight from color $c'$ of $s$ to color $c'$ of $t$, and the arc’s weight from color $c'$ of $t$ to color $c'$ of $s$.

e) $\mu_0$ called initial marking of colored Petri net is a function $\mu_0: S \to MS_\omega(C(S))$ so that for $\forall s \in S$, $\mu_0(s) \in MS_\omega(C(s))$ and by which $\mu_0(s)$ gives the number of tokens in every color for $s$ place.

A marking in any HL net $\Sigma$, is any application $\mu: S \to MS_\omega(C(S))$ so that for $\forall s \in S$, $\mu(s) \in MS_\omega(C(s))$ and which gives, for any color of $s$ place, the number of tokens in that color assigned to $s$ place.

$A \omega$-marcare of an HL net $\Sigma$ means a function $\mu_0: S \to MS_\omega(C(S))$ so that for $\forall s \in S$ we’ll have $\mu(s) \in MS_\omega(C(s))$.

Definition 1.8 [3, 8]

a) We call action in a HL net or the positive weight a function $x: T \to MS_\omega(C(T))$ so that $x(t) \in MS_\omega(C(t))$, $\forall t \in T$, which gives for any $t$ transition the weight of each color associated to that transition.

b) Action $x$ is called simple $\iff$ it applies a single $t$ transition into a single color $c \in C(t)$, connecting to the rest of transitions of the empty multiset. $x(t) = c$ is noted $x = (t, c)$.

A Petri HL-net where all it’s actions are simple is called simple colored Petri net.

An $x$ action is allowed on marking $\mu \iff \forall s \in S, \mu(s) \geq \sum_{(s, t) \in (S \times T)} \text{Pre}(s, t)(x(t))$, which is the same with $\mu \geq \text{Pre} \otimes x$.

c) By producing an $x$ action that has concession (is allowed) on a $\mu$ marking, is made a new distribution of each color’s tokens in each place, meaning a new marking $\mu^*$ pointed directly accessible from $\mu$ and given by:

$$\mu^*(s) = \mu(s) - \sum_{(s, t) \in (S \times T)} \text{Pre}(s, t)(x(t)) + \sum_{(t, s) \in (S \times T)} \text{Post}(s, t)(x(t))$$  

by introducing the $\Delta$ incidence matrix of colored Petri net:

$$\Delta(s, t) = \text{Post}(s, t) - \text{Pre}(s, t), \quad \forall (s, t) \in (S \times T),$$  

we can write $\forall s \in S, \mu^*(s) = \sum_{t \in T} \Delta(s, t)(x(t)) + \mu(s)$.

This relationship is the same with $\mu^* = \mu + \Delta \otimes x$.

As for PT-nets, we’ll note the relationship of directly accessibility with [ ] and the crowd of accesible markings with $\Delta \Sigma, \mu_0)$ or $[\mu_0]$.

Concepts of action,concession and accessibility are naturally generalised on $\omega$-markings, using $\omega$-multisets instead of multisets and replacing the word „marking“ with „$\omega$-marking“.

Any colored Petri net $\Sigma$ finite can be represented as a biparted digraph, as for the PT-nets, with the difference that the arc’s wights are functions:

$$\text{Pre}(s, t) \neq 0 \text{ for the arc that connects place } s \text{ with transition } t.$$  

$$\text{Post}(s, t) \neq 0 \text{ for the arc that connects transition } t \text{ with place } s.$$  

The null values of $\text{Pre}(s, t)$ or of $\text{Post}(s, t)$ signals the absence of the arc in the associated digraph. We’ll note with HL or CPN the class of colored Petri nets.

Lemma 1.9

Be $\Sigma$ a HL-net and $\Delta$ it’s incidence matrix. Then for $\forall \mu \in [\mu_0]$ the accessible marking of net and for any $u$ solution of equation $x^* \otimes \Delta = 0$ we have the following linear invariant of markings: $u^* \otimes \mu = u^* \otimes \mu_0$

Proof:

Because $u$ is the solution of equation $x^* \otimes \Delta = 0$, results that multiplying on the left side with $u^*$ fundamental equation $\bar{\mu} = \mu + \Delta \otimes x$ producers we obtain:

$$u^* \otimes \bar{\mu} = u^* \otimes \mu + u^* \otimes \Delta \otimes x = u^* \otimes \mu (1).$$  

As $\mu \in [\mu_0]$ results that there exist the actions:

$$x_1, x_2, ..., x_p$$  

and the markings $\mu_0, \mu_2, ..., \mu_{p-1}$ so that
\[ \mu_0[x_1 > \mu_1[x_2 > \cdots > \mu_{p-1}[x_p > \mu]. \]

Using relation (1) we get
\[ u' \otimes \mu_0 = u' \otimes \mu = \cdots = u' \otimes \mu, \]
which proof the Lemma.

The linear invariant obtained, \[ u' \otimes \mu = u' \otimes \mu_0 \], is very important for verifying certain properties of the net, such as limits, the lock of blocking, etc. There are not known till now methods for solving generally the equation system \( x' \otimes \Delta = 0 \).

Alla, Ladet, Martinez and Silva-Suarez introduced a subclass of CPN class noted HLP that can solve the upper system.

III. HLP SUBCLASS OF CLASS CPN

Definition 2.1[1, 8]

It’s called patricular colored Petri net (HLP-net) the structure \( \Sigma = (S, T, Pre, Post, C, \mu_0) \) in which
\[ Pre(s,t) = a'(s,t) \circ g_r, \quad Post(s,t) = a'(s,t) \circ g_r, \]
where \( a'(s,t) \in N \) for \( \forall (s,t) \in S \times T, \) and
\[ f_s : \bigcup_{t/\alpha(s,t)} C(t) \to MS(C(s)), a(s,t) = a''(s,t) - a'(s,t) \]
\[ g_r : C(t) \to MS(C(t)) \]
(usually, \( g_r \) function is the identity function id).

If \( |S| = n, |P| = m \) and we note \( F = \begin{bmatrix} f_1 & 0 \\ f_2 & \vdots \\ 0 & f_m \end{bmatrix}, \)
\[ G = \begin{bmatrix} g_1 & 0 \\ g_2 & \vdots \\ 0 & g_m \end{bmatrix}, \]
\[ A = (a_{ij} \cdot id)_{i,j \in l,m} \], than we get
\[ \Delta = F \otimes A \otimes G. \]

Example 2.2[4, 6]

Be \( \Sigma = (S, T, Pre, Post, C, \mu_0) \) the particular colored Petri net in figure 1 that models a FIFO queue of capacity \( n \) messages , that can transmize at once \( p \) types of messages \( m_0, \)
\[ l \in [1, p]. \]

\[ S = \{s_1, s_2\}, s_1 \text{ simbolising the empty elements of queue, and } s_2 \text{ the full elements} \]
\[ T = \{t_0, t_1, t_2\}, C = \{m_1\}_{i=l,p} \cup \{k\}_{i=k,m} \cup \{<m_i, e_k>\}_{i=l,p} \]
\[ Pre = s_1 \begin{bmatrix} 0 & f \circ g & f \circ u \\ v & id & 0 \end{bmatrix}, \]
\[ Post = s_1 \begin{bmatrix} f \circ v & f & 0 \\ 0 & g & u \end{bmatrix} \]
\[ \mu_0 = \left( \begin{array}{c} \sum_{k=1}^n e_k \\ 0 \end{array} \right). \]

The functions \( u, v, f, g \) are defined as following:
\[ u : [m_i] \to ([m_i, e_k]) \]
\[ v : [m_i, e_k] \to ([m_i, e_k]) \]
\[ f : [m_i, e_k] \to [e_k] \]
\[ g : [m_i, e_k] \to [m_i, e_k] \]
\[ \forall l \in [1, p], \forall k \in [1, n], \forall l \in [1, p], \forall k \in [1, n-1]. \]

A mark of color \( e_k \) in place \( s_1 \) says that the \( k \) element from the queue is empty, and a mark of color \( <m_k, e_k> \) in place \( s_2 \) says that there is an \( l \) message in the \( k \) element of the queue. The \( \mu_0 \) marking says that at the begining, all the queue’s elements are empty \( (\mu_0(s_1) = \sum_{k=1}^n e_k) \).

![Fig 1. Colored net that models a FIFO queue](image-url)

The net will function as: transition \( t_2 \) is allowed at \( \mu_0 \) in color \( m_1 \) if there is in \( s_1 \) a token of color \( f \circ u(m_1) = e_1 \)
(meaning that the queue’s element 1 is empty). The \( t_2 \) occurrence is transferring a token of color \( u(m_0) = <m_k, e_k> \) in \( s_2 \), which says that the element 1 of the queue contains a type 1 message. By producing \( t_1 \), this message will be transfered to the next element of the queue. If now, \( s_2 \) contains, generally,a token of color \( <m_k, e_k> \) then \( t_1 \) will be allowed on the correspondent marking, respecting the color \( <m_k, e_k> \) if location \( s_1 \) contains a token of color \( f \circ g((m_i, e_k)) = e_{k+1} \), which means that the element \( k + 1 \) of the queue must be empty.

The occurrence of \( t_1 \) brings in \( s_1 \) a token of color \( f((m_i, e_k)) = e_k \) (saying that the element \( k \) of the queue becomes empty) and a token in \( s_2 \) of color \( g((m_i, e_k)) = (m_i, e_{k+1}) \) saying that element \( k+1 \) of the queue has been occupied. This transfer functions as long as in the queue are empty elements, the message getting to the \( n \) element...
of the queue. By occuring the $t_0$ transition, the message will leave the queue, getting from the producer to the consumer.

IV. ALGORITHM FOR DETERMINING INVARIANTS OF SUBCLASS HLP

To solve the equation $\Delta' \odot \Delta = 0$ in this class of colored Petri nets, where $x$ is a vector with $n$ components-linear functions, we'll note $\text{Ker} \Delta = \{x \mid x' \odot \Delta = 0\}$ where $\Delta = F \odot A \odot G$ and we'll calculate $\text{Ker} \Delta$ in two stages.

First we take $A' = (a_{ij})_{j=1,n}$ built in $A$, where $a_{ij} \in Z$.

We can consider $A'$ a linear application from $Z^n \rightarrow Z^m$. So for $\forall V \in Z^n, \exists U \in Z^n$ so that $U' = V' \ast A'$.

As $Z^n$ is module over $Z$ it has a base and because $Z$ is main ring, its kernel $A'$ adica ($\text{Ker} (A')$) is a submodule in $Z^n$ and accepts a base of dimension $q$ so $q = n - \text{rg} A'$. This base can be determined with the classic known methods.

Now be $\{V_{r}\}_{r=1,q}$ a base of $\text{Ker} A', V = (v_{ir})_{j=1,n}$. We have $V_{r} \in \text{Ker} A' \Rightarrow V_{r}' \ast A' = 0 \Rightarrow \sum_{i=1}^{n} v_{ir} a_{ij} = 0$ for $\forall j \in 1, m$ and $\forall r \in 1, q$ (1).

We note for every $r \in 1, q$ with:

$$FS_{r} = \{k \in 1, n \mid v_{kr} \neq 0 \text{ and the functions } f_{k} \text{ are different two by two}\}.$$ 

Then we expand $\forall f_{k} \ast k \in FS_{r}$ to $f_{r}'$ so $f_{r}'|_{dom f_{k}} = f_{k}$ and so that the composition of functions $f_{k}$ with $k \in FS_{r}$ make sense.

We note this composition with $\pi_{r} = \prod_{k \in FS_{r}} f_{k}$. This way, tot every $V_{r}$ from the base, we associate the composition $\pi_{r}$ (if more functions are equal, we take the one with the minimum index).

We assume that for $\forall i \in FS_{r}$ we have the commutativity property:

$$\prod_{k \in FS_{r}} f'_{k} = \left(\prod_{k \in FS_{r}} f_{k}\right) \circ f_{i}.$$ 

Applying $\pi_{r}$ to relation (1) we get:

$$\pi_{r}\left(\sum_{i=1}^{n} v_{ir} a_{ij}\right) = \prod_{k \in FS_{r}} f'_{k} \sum_{i=1}^{n} v_{ir} a_{ij} = \sum_{i=1}^{n} \prod_{k \in FS_{r}} f'_{k} v_{ir} a_{ij} =$$

$$=\sum_{i=1}^{n} \left(\prod_{k \in FS_{r}} f'_{k}\right) \circ f_{i} v_{ir} a_{ij} = \sum_{i=1}^{n} \prod_{k \in FS_{r}} f'_{k} v_{ir} a_{ij} = 0, \forall j \in 1, m$$

Define a vector $U_{r} = (u_{ir})_{j=1,n}$, $u_{ir} = \left(\prod_{k \in FS_{r}} f_{k}\right) v_{ir}$ and so we have $\sum_{i=1}^{n} u_{ir} f_{ir} a_{ij} = 0$ which is equivalent with:

$$U_{r}' \odot F \odot A = 0$$

so $U_{r} \in \text{Ker} (F \odot A)$. It follows that $U_{r}$ can be built if you can get the extensions of the functions $f_{k}'$ so that it meets the commutativity condition.

It can take as an example $f_{1}'(x) = x$ for any $x \notin \text{dom} f_{k}$. Now be $\Delta' \odot F \odot A \odot G$ and $\forall x \in \text{Ker} (F \odot A)$. Obviously $x' \odot F \odot A = 0$ and so:

$$x' \odot F \odot A \odot G = 0 \Rightarrow x' \odot F \odot A = 0 \Rightarrow x \in \text{Ker} (F \odot A)$$

saying that in this case:

$$\text{Ker} \Delta = \text{Ker} (F \odot A).$$

It follows the next algorithm to solve the equation:

$$x' \odot \Delta = 0.$$

Algorithm 3.1

1) Starting from $\Delta = (a_{ij} f_{i} \circ g_{j})_{j=1,n}$ and determining $A = (a_{ij} id)_{j=1,n}$.

2) Determin $A' = (a_{ij})_{j=1,n}$ starting from $A$, where $a_{ij} \in Z$.

3) Pick a base $\{V_{r}\}_{r=1,q}$ of $Ker A'$ so that for $\forall r \in 1, q$ the non-zero components $(v_{ir})_{j=1,n}$ to meet a minimum number of different functions $f_{k}$ so they can be extended to $f_{r}'$ functions for which the commutativity property to occur.

4) Build for every $r \in 1, q, FS_{r} = \{k \in 1, n \mid v_{kr} \neq 0 \text{ and the proper functions } f_{k} \text{ are different two by two}\}$

5) Expand for every $r \in 1, q$ the functions $f_{k}$ to $f_{r}'$ so that they check the commutativity property:

$$\prod_{k \in FS_{r}} f'_{k} = \prod_{k \in FS_{r}} f_{k} \circ f'_{i}, \forall i \in FS_{r}$$

6) Build the vector $U_{r} = (u_{ir})_{j=1,n}$ by $u_{ir} = \left(\prod_{k \in FS_{r}} f_{k}\right) v_{ir}$ for every $r \in 1, q$ and so obtain $q$ solutions of equation :

$$x' \odot \Delta = 0.$$
Advances in Communications, Computers, Systems, Circuits and Devices

shown in Fig. 1. We saw in example 2.2 how \( k \bowtie \Delta = 0 \) in several stages, by the decomposition of the \( \Delta \) matrix, following the next algorithm:

**ALGORITHM 3.4**

1. Pick in \( \Delta \) a sub-matrix \( \Delta_1 \) of minimum rank so that \( \Delta_1 \) be the incidence matrix of type required in the class HLP.

2. Determine with algorithm 3.1. crowd \( S_i \) of solutions of the equation \( x' \bowtie \Delta_1 = 0 \). So \( S_i \bowtie \Delta_1 = 0 \).

3. Pick in \( S_i \bowtie \Delta_1 \) a sub-matrix \( \Delta_2 \) with the same conditions as for \( \Delta_1 \), getting a second crowd of solutions \( S_2 \) for the equation \( x' \bowtie \Delta_2 = 0 \). So having \( S_i \bowtie \Delta_2 = 0 \).

Further, use the matrix, \( S_2 \bowtie S_i \bowtie \Delta = 0 \).

Iterate this process until obtaining on the \( i \) iteration:

\[ S_i \bowtie S_{i-1} \bowtie \ldots \bowtie S_1 \bowtie \Delta = 0 \]

In this case, the final solution is \( S_1 \bowtie S_2 \bowtie \ldots \bowtie S_i \bowtie \Delta = 0 \).

**V. THE STUDY OF THE PROPERTIES QUEUE OF THE FIFO BY INVARIANTS OF THE SUBCLASS HLP**

Be \( \Sigma \in HLP \) shown in Fig. 1. We saw in example 2.2 how this net works. Note that:

\[ \Delta = \text{Post} \text{ - Pre} e \begin{bmatrix} f \circ v & f - f \circ v & - f \circ v \\ v & g - id & u \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & id & -id \\ id & -id & id \end{bmatrix} \begin{bmatrix} v & 0 & 0 \\ 0 & g - id & 0 \\ 0 & 0 & u \end{bmatrix} \]

Going through algorithm 3.1. we obtain:

\[ A = \begin{bmatrix} id & -id & -id \\ -id & id & id \end{bmatrix}, \text{ so } A' = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{bmatrix}. \]

Have \( A' : Z^2 \rightarrow Z^3 \) and \( \text{rg} A' = 1 \) and \( \text{Ker} A' \) is a submodule of \( Z^2 \).

So the base dimension of module \( \text{Ker} A' \) is \( q = n - \text{rg} A' = 1 \).

So there is a single vector in the base; be this \( V_i = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \).

So \( FS_i = \{ 1, 2 \} \).

Because: \( f: \{ m_i, e_j \} \rightarrow \{ e_k \} \) we’ll expand:

\[ \text{id}: \{ m_i, e_j \} \rightarrow \{ m_i, e_j \} \text{ to } \text{id}: \{ m_i, e_j \} \cup \{ e_k \} \rightarrow \{ m_i, e_j \} \cup \{ e_k \} \]

\[ f': \{ m_i, e_j \} \cup \{ e_k \} \rightarrow \{ m_i, e_k \} \cup \{ e_k \} \]

Analogous expand \( f \) to:

\[ f': \{ m_i, e_j \} \cup \{ e_k \} \rightarrow \{ m_i, e_k \} \cup \{ e_k \} \text{ by } f'. \]

\[ f': \{ m_i, e_j \} = f \text{ and } f'(e_k) = e_k. \]

Having:

\[ f' \circ \text{id}: \{ m_i, e_j \} = f'(m_i, e) = e_k \]

\[ f' \circ \text{id}: \{ m_i, e_j \} = \text{id}' \circ f'(m_i, e) = \text{id}'(e_k) = e_k \]

\[ f' \circ \text{id}: \{ m_i, e_j \} = \text{id}'(e_k) = e_k \]

\[ f' \circ \text{id}: \{ m_i, e_j \} = \text{id}'(f'(e_k)) = \text{id}'(e_k) = e_k \]

So \( f' = \text{id} \circ f'. \)

So the vector: \( U_1 = (u_{11})_{k=1,2} \in \text{Ker} \Delta \) where:

\[ u_{11} = \frac{1}{k_{1,1}} v_{1,1} = \text{id}' \text{ and: } u_{11} = \frac{1}{k_{1,2}} v_{1,2} = \text{id}' \text{ or } \forall \mu \in \{ \mu_0 \}. \]

This relationship is equivalent to:

\[ (\text{id}', f') \circ \begin{bmatrix} \mu(s_1) \\ \mu(s_2) \end{bmatrix} = (\text{id}', f') \circ \begin{bmatrix} \sum_{k=1}^n e_k \\ 0 \end{bmatrix} \]

\[ \Rightarrow \mu(s_1) + f'(s_2) = \sum_{k=1}^n e_k. \]

This invariant can be interpreted as follows:

If \( s_1 \) contains a token of color \( e_j \), this color \( f'(s_2) = 0 \) means \( \mu(s_2) = 0 \) which is saying that, \( \forall l \in \{ 1, p \}, s_2 \) doesn’t contain any token of color \( \{ m_i, e_j \} \) equivalent with the fact that the element \( j \) of the queue doesn’t contain any message \( m_i, l \in \{ 1, p \} \).

Also, by this invariant we notice that element \( j \) of the queue contains only one message \( m_i \) because, if we assume the absurd that it contains the messages \( m_i, m_i \) then:

\[ \mu(s_2) = \{ m_i, e_j \} + \{ m_i, e_j \} \text{ and so: } f'(\mu(s_2)) = f'(\{ m_i, e_j \} + \{ m_i, e_j \}) = 2e_j \]

which is noticed, through the invariant, that is impossible, because for any color \( e_j \), in the right member \( e_j \) appears a single time.

Also through the invariant we obtain \( \mu(s_1) \leq \sum_{k=1}^n e_k \) so:

\[ f'(\mu(s_2)) \leq \sum_{k=1}^n e_k \text{ equivalent with } \mu(s_2) \leq \sum_{k=1}^n m_i, e_j \text{ for any } l \in \{ 1, p \}. \]

But these two relations tell us that our net is limited.

Using the invariant above we’ll show that the net that
modulates a FIFO queue can not every block.
Indeed, be \( \forall \mu \in [\mu_0] \). Notice of invariant that there are 3 possibilities:

1. \( \mu(s_1) \leq \sum_{k=1}^n e_k \); in this case, all queue’s elements are empty and transition \( t_2 \) is allowed at \( \mu \).

2. \( \mu(s_1) = 0 \Rightarrow f'(\mu(s_2)) = \sum_{k=1}^n e_k = \mu(s_2) = \sum_{k=1}^n (m_i, e_k) \) i.e place \( s_2 \) is marked and so is allowed \( t_0 \) at \( \mu \).

3. \( \mu(s_1) = e_{k_1} + e_{k_2} + \ldots + e_{k_j}, j \in 1, n - 1 \)
   a) If for \( \forall r \in \overline{1, j}, k_r \neq n \) then by invariant follows that \( f'(\mu(s_2)) = e_n \) i.e \( \mu(s_2) = \langle m_i, e_n \rangle \) and so the message \( m_i \) reached the last element of the queue, equivalent with the fact that \( t_0 \) will be allowed at \( \mu_1 \) and by its producing, \( m_i \) will leave the queue, getting to the consumer.
   b) If \( \exists r \in \overline{1, j} \) so \( k_r = n \) so then \( \exists k \in \overline{1, n - 1} \) so \( f'(\mu(s_2)) = e_k \) i.e \( \mu(s_2) = \langle m_i, e_k \rangle \). This says that there is a type \( l \) message in the \( k \) element of the queue.

Because we have seen that \( s_2 \) can’t contain a token of color \( \langle m_i, e_{k+1} \rangle \), means that \( s_1 \) will contain a token of color \( e_{k+1} \) (according to the invariant above \( \mu(s_1) = e_{k+1} \)) and so the element \( k+1 \) of the queue is empty, meaning that transition \( t_1 \) can occur.

In this case \( t_1 \) is allowed at \( \mu \).

Results of this analysis that any marking that’s accesible from the initial marking is not the last one that’s possible, so the colored net which has shaped the FIFO queue can not block.

VI. CONCLUSION

It is known the fact that the invariants of a class of Petri nets allow the study of system’s properties that can be shaped with nets of that class, properties such as: boundedness, liveness, lack of blocking, etc. This is also available for the systems shaped through Colored Petri nets, which is known that drastically simplifies the complexity of the study.

This paper presents an efficient method to determine the invariants of a class of colored Petri nets by combination of algorithm 3.1 and 3.4.

Once the invariants are found, the properties of the shaped systems in the class of colored nets are studied as it was shown in application of paragraph 4.

REFERENCES