Conflict Sets in Colored Petri Nets

C. Dutheillet(1)

S. Haddad(2)

(1) IBP-Laboratoire MASI
Université P. & M. Curie
75252 Paris cedex 05

(2) LAMSADE-Université Paris Dauphine
4 place Jussieu place de Lattre de Tassigny
75775 Paris cedex 16

Abstract

Generalized Stochastic Petri Nets provide the modeller with immediate transitions, but a model will be correct only if the modeller can specify how to solve the firing conflicts between these transitions. This task is usually cumbersome and may be impossible for large nets. In confusion-free nets, these conflict sets are subsets of equivalence classes of a structural conflict relation, which greatly simplifies the previous task. This paper solves the problem of detecting confusion in colored stochastic Petri nets and computing the equivalence classes of the structural conflict relation. Our approach relies on two techniques: the symbolic representation of structural relations and the definition of operations on these symbolic relations. Combined with the lumping method presented in [13], our algorithm could be the basis of an efficient tool for the analysis of colored Generalized Stochastic Petri Nets.

1 Introduction

One of the advantages of using Petri nets for the modelling of systems is that the same model can be used for the analysis of behavioral properties, such as liveness or termination, and structural properties. Structural properties enclose all the properties that do not depend on the initial state of the system, and they are closely related to the structure of the Petri net graph.

The application of structural relations to Generalized Stochastic Petri Nets (GSPNs) is twofold. On the one hand, structural relations can be used to improve the implementation of simulation tools by optimizing the management of the event list. For instance, the structural conflict relation accounts for the fact that the firing of a transition t_i may disable another transition t_j. Hence, after firing t_i, only those transitions that are in structural conflict with t_i should be tested for erasing from the event list [1].

On the other hand, confusion may be detected by means of structural relations. Actually, when several immediate transitions are enabled in some marking, two approaches can be used for the determination of the transition to be fired. The first one is to use random switches [2], i.e., define for every possible set of simultaneously enabled transitions the respective firing probabilities of each of these transitions. However, the sets of simultaneously enabled transitions completely depend on the current marking, and thus may be very difficult to enumerate.

The second approach is to associate a weight with every immediate transition [3]. In this case, the firing probability of a transition in a marking is given by the ratio of its weight to the sum of the weights of all enabled transitions. The weights correctly define the firing probabilities only if the designer knows the conflicts in the model. This problem can be overcome by computing structural extended conflict sets, which are equivalence classes of a conflict relation. However, partitioning transitions is not enough. In order to have an adequate specification, the net must be confusion-free, i.e., whenever two transitions belonging to different conflict sets are enabled, the future behavior of the net does not depend on the order of these firings. In [3], an algorithm has been proposed to compute extended conflict sets and detect confusion in GSPNs.

However, modelling complex systems with Petri nets quickly results in inextricable models. High-level net models, namely Predicate/Transition nets [4] and colored nets [5], allow a more concise representation, although the same features of the system are modelled.

A colored Petri net is a net in which tokens are identified by colors. Color domains are associated with places and transitions and determine which colors can mark the place (resp. fire the transition). When firing a transition, a number of tokens is taken from each input place, according to the incidence function labelling the arc between the place and the transition, and a number of tokens is produced in each output place, according to the function labelling the arc between the transition and the place. Hence, a structural relation in a colored net will relate a color of a node to a color of another node.

Structural relations may be detected in colored nets by studying the structure of the unfolded Petri net [6], i.e., an ordinary Petri net that has the same behavior as the colored net. However, a more efficient approach is to take advantage of the structure of the color functions to detect these relations directly at the colored net level. This can be done by means of symbolic relations, which are functions relating subsets of the color domains of two nodes [7]. The aim of this paper is thus to apply the symbolic relation approach to the definition of structural relations and the detection of confusion.

The paper is organized as follows. In Section 2, we recall the approach for structural detection of confusion in ordinary Petri nets. Section 3 contains the definition of colored nets, together with some definitions on multisets and powersets. Section 4 introduces the notion of symbolic relation and extends to colored nets the algorithm presented in Section 2. In Section 5, we discuss the interest of using symbolic relations to compute structural relations in colored nets. Finally, Section 6 presents some perspectives to this work.

2 Confusion in Ordinary Petri Nets

In this section, we recall the definition of confusion and the algorithm that makes it possible to detect confusion at the structural level in ordinary Petri nets.

The behavior of a system is made of a set of actions that are either conlicting, i.e., they may be simultaneously possible, but the execution of one may disable the other, or non-conflicting. If two actions are non-conflicting then the future behavior of the system will be independent of the order of their executions. In the Petri net modeling the system, transitions can be accordingly partitioned into equivalence classes, such that two transitions belonging to different classes represent non-conflicting actions. Such classes might be obtained by computing the transitive closure of the conflict relation linking transitions such that the firing of one transition may disable the other one. However, the resulting relation can lead to incoherent classes. In the net in Figure 1 [3], t2 and t3 are conflicting, whereas t1 is not conflicting with t2 nor with t3 and thus belongs to another class. However, depending on whether t1 fires before t2 or not, t3 will be enabled or not. This particular structure is known under the name of asymmetric confusion. This definition of confusion is slightly different from the definition in [8], [9], but is identical to the definition in [3], [10] where an algorithm is proposed to detect confusion at the structural level.

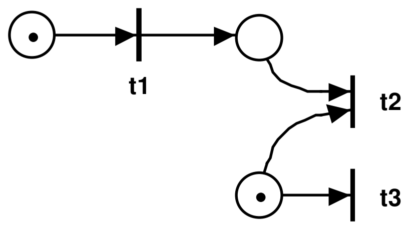


Fig. 1 : Example of asymmetric confusion

Stochastic Petri nets with immediate transitions (GSPNs) must not include asymmetric confusion structures among immediate transitions to preserve the correctness of the timing semantics of the net.

In this paper, we will consider only the case of GSPNs with two transition priority levels, one corresponding to timed transitions, and the other one to immediate transitions. For these models, the detection of confusion is useful only among immediate transitions. Thus we will consider in the rest of the paper GSPNs in which all timed transitions have been deleted.

We now recall the detection algorithm for ordinary stochastic Petri nets. Asymmetric confusion appears when the firing of a transition t_1, which has no conflict relation with another enabled transition t_2, enables a transition t_3 that instead has a conflict relation with t_2. Hence, the detection of confusion at the structural level implies that we define

1 - which transitions have a conflict relation with some transition t,

2 - which transitions may be enabled by the firing of some transition t_1, knowing that t_2 is enabled.

The first phase begins by the definition of the conflict relation. Two transitions in conflict must not be *mutually exclusive* (ME). We consider that a transition t_1 is in *structural conflict* (SC) with a transition t_2 if the firing of t_1 may disable t_2. As we want to define classes of conflicting transitions, we must make this relation symmetric. By taking the reflexive and transitive closure of this new relation, we obtain an equivalence relation, called *symmetric structural conflict* (SSC), whose classes are called *extended conflict sets* (ECS).

The second phase can be performed by defining a relation $CC(t_2)$ that allows us to find which transitions can be enabled by the firing of a transition t_1 knowing that t_2 is enabled, and are thus *causally connected* to t_1. Taking the transitive closure of this relation, we obtain the causally connected set of t_1, $CCS_{t2}(t_1)$.

Having this information, it is possible to detect confusion at the structural level. A sufficient condition for the net to be confusion-free is that given two transitions t, t' belonging to the same conflict set, no transition t" can be causally connected to t' knowing that t is enabled. The process of confusion detection is summarized in Figure 2.

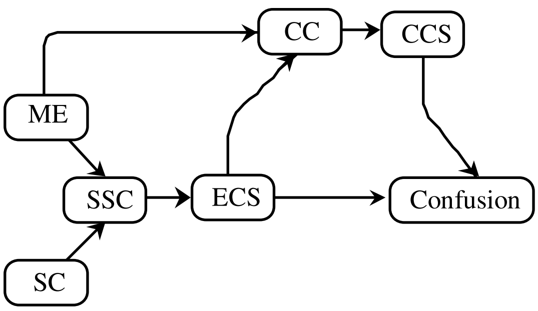


Fig. 2 : confusion detection process

Moreover if we prove that the net is confusion-free, the operations that have been performed to obtain this result can be used to improve the construction of the reachability graph. In fact, in a confusion-free net, the order for firing immediate transitions enabled in a given marking has no effect on the overall behavior of the net. Hence, the parallel firing of immediate transitions belonging to different extended conflict sets is possible and allows a reduction in the space needed for storing vanishing markings [10] [3].

The next section is devoted to the extension of this algorithm to colored nets by using symbolic relations. We will successively consider mutual exclusion, structural conflict and causal connection. From these relations, we derive a sufficient condition for a colored net to be confusion-free.

3 Colored Petri Nets

The analysis of colored Petri nets is based on the handling of multisets. We present in the next section some basic definitions on multisets.

3.1 Multisets

Definition 3.1 A multiset a over a non-empty set E is a mapping $a \in [E \rightarrow \mathbb{N}]$.

Intuitively, a multiset is a set that can contain several occurrences of the same element. It can be represented by a formal sum

$$a = \sum_{x \in E} a(x).\ x$$

in which the non-negative integer a(x) represents the number of occurrences of the element x in the multiset a. We denote Bag(E) the set of finite multisets over E. Actually, we can extend this definition to mappings $a \in [E \rightarrow \mathbb{Z}]$, and denote $Bag_Z(E)$ this new set of finite multisets over E. In this case, multisets lose their intuitive meaning.

We define a set of operations on Bag(E).

Definition 3.2 Let a, b be two elements of Bag(E). Then :

- $a + b = \sum_{x \in E} [a(x) + b(x)].\ x$
- $a - b = \sum_{x \in E} (\sup[a(x) - b(x), 0]).\ x$
- $a \cap b = \sum_{x \in E} (\inf[a(x), b(x)]).\ x$

The definition of subtraction generalizes the one usually employed, which applies only if $a \geq b$, i.e., if $\forall x \in E,\ a(x) \geq b(x)$.

The definition of functions on multisets is similar to the definition of functions on sets. In this paper, we will only consider linear functions. Linear functions on Bag(E) needs to be defined only on the items of E as, due to the linearity of f,

$$f\left(\sum_{x \in E} a(x).\ x\right) = \sum_{x \in E} a(x).\ f(x)$$

Definition 3.3 Let f be a function from Bag(E) to Bag(F). The transpose f^t of f is a function from Bag(F) to Bag(E), defined by :

$$f^t(x)(y) = f(y)(x)$$

In this notation, the first pair of brackets denotes the application of the function, while the second one refers to the multiset notation.

We will also use operations on functions that are derived from operations on multisets.

Definition 3.4 Let f be a function from Bag(E) to Bag(F). Let *op* be a binary operator on Bag(F). Then f *op* g is a function from Bag(E) to Bag(F) defined by :

$(f\ op\ g)(x) = f(x)\ op\ g(x)$

3.2 Colored Petri nets

We recall the definition of a colored net [5], the associated firing rule and the corresponding unfolded net.

Definition 3.5 A colored Petri net N is a 7-tuple $N = \langle P, T, C, W^-, W^+, H, M_0 \rangle$ where :

- P is the set of places
- T is the set of transitions with $P \cap T = \emptyset$ and $S = P \cup T \neq \emptyset$
- C is the color function from $P \cup T$ to Ω , where Ω is some finite 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.
- W^- (resp. W^+, H) is the forward (resp. backward, inhibitor) incidence matrix on P x T , where $W^-(p, t)$ (resp. $W^+(p, t)$, $H(p, t)$) is a function from Bag[C(t)] to Bag[C(p)]
- M_0 the initial marking of the net is a vector on P, where $M_0(p)$ is an item of Bag[C(p)].

The firing rule defines the dynamic behavior of the net.

Definition 3.6 Firing rule

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

$$\forall p \in P,\quad M(p) \geq W^-(p, t)(c_t)$$
$$\text{and}\quad (\forall c \in C(p),\ M(p)(c) < H(p,t)(c_t)(c)\ \text{ or }\ H(p,t)(c_t)(c) = 0\ )$$

- The firing of t for a marking M and a colour $c_t \in C(t)$ gives the marking M' defined by :

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

Several times in this paper, we will refer to the unfolded Petri net of a colored net. A colored net is in fact an abbreviation of an ordinary Petri net with the same behavior. This equivalent ordinary Petri net is obtained with the following procedure.

- One place (resp. one transition) is created for each possible couple (p, c), $c \in C(p)$ (resp. (t, c), $c \in C(t)$).
- An arc between (p, c) and (t, c') (resp. between (t, c') and (p, c)) exists and has a valuation k if the forward (resp. backward) incidence function is such that $W^-(p, t)(c')(c) = k$ (resp. $W^+(p, t)(c')(c) = k$).
- An inhibitor arc between (p, c) and (t, c') exists and has a valuation k if the inhibitor incidence function is such that $H(p, t)(c')(c) = k$.
- The initial marking of (p, c) is the number of color c tokens that p contains in the colored net.

Defining a structural relation in a colored net consists in, given a node of the net and an associated color, finding for another node the set of colors of this node that are in relation with the considered color of the initial node. As we will have to handle sets of colors, we recall some basic notions on powersets.

3.3 Powersets

The powerset of E is the set of subsets of E and is denoted by P(E). Functions can be defined on powersets, and using the union as an additive operator, we can also define linear functions. Different operations can be applied on these linear functions.

Definition 3.8 Let f and g be two functions from P(E) to P(F). Then

- $f \cap g : P(E) \rightarrow P(F)$ is defined by :
  $[f \cap g](x) = f(x) \cap g(x)$
- $f \cup g : P(E) \rightarrow P(F)$ is defined by :
  $[f \cup g](x) = f(x) \cup g(x)$

Definition 3.9 Let f be a function from P(E) to P(F). Then

- $f^t : P(F) \rightarrow P(E)$ is defined by :
  $y \in f^t(x) \Leftrightarrow x \in f(y)$
- ${}^c f : P(E) \rightarrow P(F)$ is defined by :
  ${}^c f(x) = \{\ y \mid y \notin f(x)\}$

Definition 3.10 Let f be a function from P(E) to P(F) and g be a function from P(F) to P(G). Then

- $g \circ f : P(E) \rightarrow P(G)$ is defined by :
  $[g \circ f](x) = g[f(x)]$

Definition 3.11 Let f be a function from P(E) to P(E). Then

- f^n is recursively defined by :
  $f^0 = \text{Identity}$ and $f^n = f \circ f^{n-1}$
- $f^+ = \bigcup_{n > 0} f^n$
- $f^* = \bigcup_{n \in N} f^n$

Functions on powersets can be related to functions on multisets.

Definition 3.12 Let f be a function from Bag(E) to Bag(F). Then $\overline{f}$ is a function from P(E) to P(F) defined by :

$$\overline{f}(x) = \{y \in F \mid f(x)(y) > 0\}$$

The two definitions of the transpose can also be related.

Definition 3.13 Let f be a function from Bag(E) to Bag(F). Then $\overline{f}^{\,t} = \overline{f^t}$.

4 Confusion in Colored Petri Nets

With the introduction of stochastic colored nets, the extension of the algorithm for confusion detection became necessary to ensure a correct timing semantics of the colored net. A first algorithm was proposed, which worked on the unfolded net [6]. But to avoid working on the unfolded net, we must handle color functions at a symbolic level. The next section recalls the definition of symbolic relations, together with the results that make it possible to work directly at the net level.

4.1 Symbolic Relations

A structural relation in a Petri net is a relation between two nodes of the graph of the net. Consider for instance the "*precedes*" relation. In an ordinary Petri net, a place p precedes a transition t iff there is an input arc between p and t. If we consider the unfolded Petri net of a colored net, such a relation can be easily extended :

$$(p, c)\ \textit{precedes}\ (t, c') \Leftrightarrow W^-(p, t)(c')(c) > 0$$

$W^-(p, t)$ is a function from Bag[C(t)] to Bag[C(p)], and has a corresponding function $\overline{W^-(p, t)}$ from P[C(t)] to P[C(p)], which is such that

$$\overline{W^-(p, t)}\,(c') = \{c \in C(p) \mid W^-(p, t)(c')(c) > 0\}.$$

Hence, the result of this function is the set of all the color instances of p that precede (t, c'). We can write :

$$(p, c)\ \textit{precedes}\ (t, c') \Leftrightarrow c \in \overline{W^-(p, t)}\,(c')$$

If now we want to determine which transitions precede a place (p, c), we can write :

$$\begin{aligned}(t, c)\ \textit{precedes}\ (p, c') &\Leftrightarrow W^+(p, t)(c)(c') > 0\\ &\Leftrightarrow W^+(p, t)^t(c')(c) > 0\end{aligned}$$

hence, $(t, c)\ \textit{precedes}\ (p, c') \Leftrightarrow c \in \overline{W^+(p, t)^t}\,(c')$

We can now introduce the formal definition of symbolic relations. For more details, see [7].

Definition 4.1 Let N be a colored net. Let M be a square matrix indexed by the nodes of N such that M(s', s) is a function from P[C(s)] to P[C(s')]. M is called a symbolic relation of N, and $\mathcal{R}_M$ denotes a relation between the nodes of the unfolded net, defined by :

$$(s', c')\ \mathcal{R}_M\ (s, c) \Leftrightarrow c' \in M(s', s)(c).$$

Thus, if we consider the relation $\mathcal{R}_M$ = *precedes*, the symbolic relation M is defined by :

$\forall$ p, p' $\in$ P, $\forall$ t, t' $\in$ T,

$M(p, t) = \overline{W^-(p, t)}$ $\quad$ $M(t, p) = \overline{W^+(p, t)^t}$

$M(p, p') = 0$ $\quad$ $M(t, t') = 0$

The operations, e.g. intersection or transpose, that we have defined in the previous section apply to symbolic relations. But we are also interested in computing the transitive closure of a relation by means of the associated symbolic relation. Therefore, we extend to symbolic relations some classical operators on relations.

Definition 4.2 Let N be a colored net, let M and M' be two symbolic relations of N. Then M.M' the product symbolic relation of M and M' is defined by :

$$M.M'(s', s) = \cup_{s'' \in S} M(s', s'') \circ M'(s'', s)$$

The product $\mathcal{R}_M . \mathcal{R}_{M'}$ of two relations $\mathcal{R}_M$ and $\mathcal{R}_{M'}$ is usually defined by :

$$(s', c') \in \mathcal{R}_M . \mathcal{R}_{M'}\ (s, c) \Leftrightarrow \exists (s'', c'') \mid$$
$$(s', c') \in \mathcal{R}_M\ (s'', c'') \text{ and } (s'', c'') \in \mathcal{R}_{M'}\ (s, c)$$

It is not difficult to show that the relation associated with the product of two symbolic relations is the product of the relations, i.e.,

$$\mathcal{R}_{M.M'} = \mathcal{R}_M . \mathcal{R}_{M'}$$

Definition 4.3 Let N be a colored net and M be a symbolic relation of N

M^n is recursively defined by

M^0 = Identity and $M^n = M^{n-1}$. M. We define

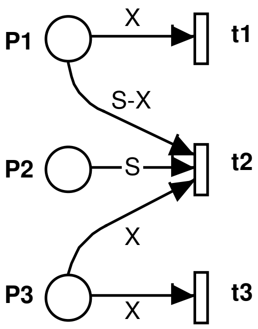
$$M^+ = \bigcup_{n > 0} M^n \qquad M^* = \bigcup_{n \in N} M^n$$

Then $\mathcal{R}_{M^+}$ is the transitive closure of $\mathcal{R}_M$, and $\mathcal{R}_{M^*}$ is the reflexive transitive closure of $\mathcal{R}_M$. We could also prove that like in the ordinary case, the union defining M^+ (resp. M^*) is stationary at some step. This would provide an algorithm to compute the transitive closure. But on ordinary graphs, Warshall's algorithm [11] is more efficient and we choose to extend this algorithm to symbolic relations. Unlike what happens in ordinary graphs, we have to compute the transitive closure inside a node of the high-level net (i.e., a set of nodes in the unfolded net). Thus, the algorithm is [7] :

For $s \in S$ do $M(s, s) := M(s, s)^+$
For $s \in S$ do
  For $s' \in S$ do
    For $s'' \in S$ do
      $M(s'', s') := M(s'',s') \cup M(s'', s) \circ M(s,s)^* \circ M(s,s')$

We illustrate on a simple example the necessity of modifying the ordinary graph algorithm by introducing the term $M(s, s)^*$.

Consider the colored net in Figure 3. We want to compute the transitive closure of the relation "shares an input token".



$\forall s \in P \cup T$

$C(s) = C = \{a, b, c\}$

$\forall x \in C$

$X(x) = x$

$S(x) = a + b + c$

Fig. 3 : example of transitive closure

By reasoning on the unfolded net, we see that (t_1, c) shares a token c in P1 with (t_2, a) and (t_2, b). (t_2, a) and (t_2, b) share the tokens a, b and c in P2 with (t_2, c), and (t_2, c) shares a token c in P3 with (t_3, c). Hence, $[(t_1, c), (t_3, c)]$ must belong to the transitive closure of the relation.

Let us now give the symbolic expression of the relation. $\overline{W^-(p, t_i)}$ (c) gives the set of colors of p that are input of t_i. By applying $\overline{W^-(p, t_j)}^{\,t}$ to this set of colors, we obtain the colors of t_j that have one of these tokens in input and thus share an input token with (t_i, c). Hence, if we call SH the symbolic relation associated with the relation "shares an input token", we have :

$$SH\,(t_i, t_j) = \bigcup_{p \in P} \left(\overline{W^-(p, t_j)}^{\,t} \circ \overline{W^-(p, t_i)} \right)$$

If we apply the transitive closure algorithm without introducing the term $M(s, s)^*$, we have :

$$SH(t_1, t_3) = SH(t_1, t_3) \cup SH(t_1, t_2) \circ SH(t_2, t_3)$$

It is easy to verify that $\forall x \in C$,

$\overline{X}$ is defined by $\overline{X}(x) = \{x\}$,

$\overline{S}$ is defined by $\overline{S}(x) = C$

$\overline{S\text{-}X}$ is defined by $\overline{S\text{-}X}(x) = C \setminus \{x\}$

and that the transpose of all these functions is equal to the function itself. Hence, $SH(t_1, t_3) = \overline{S\text{-}X}$, which means that an instance of t_3 is associated with all the instances of t_1, except itself, in the transitive closure of the relation. But we have shown that actually $[(t_3, c), (t_1, c)]$ must belong to this transitive closure. The reason for the mistake is that we have not taken into account the fact that a color instance of t_2 shares an input token with all the color instances of t_2. This is why we have to introduce the term $M(s, s)^*$ in the algorithm, and in that case, the result is correct.

The next paragraphs are devoted to the determination of basic structural relations in colored nets by means of symbolic relations. We will successively consider mutual exclusion, structural conflict and causal connection. These relations are preliminary steps for the detection of confusion. They can also be used to improve the implementation of simulation.

4.2 Mutual Exclusion

Two kinds of mutual exclusion may be checked independently of the marking of the net. Structural mutual exclusion occurs when the valuations of the arcs prevent two transitions from being simultaneously enabled. Marking mutual exclusion occurs when some place invariant ensures that no reachable marking will provide enough tokens for the transitions to be simultaneously enabled. We present here only the relation of structural mutual exclusion, but marking mutual exclusion can be found in [12].

In an ordinary net, a sufficient condition for two transitions to be in mutual exclusion is that a place is connected to one transition by an inhibitor arc and to the other one by an input arc with a greater or equal valuation. In a colored net, the valuations are no longer integers, but color functions. Hence, in order to extend the comparison of the valuations, we need to define an operator between color functions with the same codomain and different domains.

Definition 4.4 Let f be a function from Bag(E) to Bag(F) and g be a function from Bag(G) to Bag(F). Then (g \ f) is a function from $\mathcal{P}$(E) to $\mathcal{P}$(G) defined by :

$(g \setminus f)(c_1) = \{ c_2 \in G \mid \exists c \in F,\ g(c_2)(c) \geq f(c_1)(c) > 0\}$

In Figure 4, it is clear on the ordinary net that t1 and t2 are mutually exclusive. In the colored net, the application of the operator \ to color c_1 of t1 gives the set of colors of t2 such that, for at least one color c in P, the integer valuation $g(c_2)(c)$ is greater than $f(c_1)(c)$, which ensures the mutual exclusion. Note that in this case, the second inequality in the definition of (g \ f) ensures that the inhibitor arc exists.

Hence, based on Definition 4.4, we can define the symbolic relation $mutex(t_2, t_1)$ by :

$$mutex\ (t_2, t_1) = \bigcup_{p \in P} \left[W^-(p, t_2) \setminus H(p, t_1)\right]$$

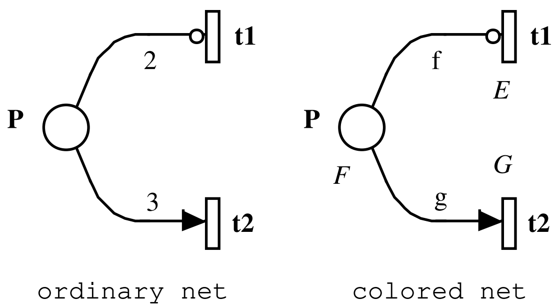


Fig. 4 : mutual exclusion in ordinary and colored nets

The set $mutex(t_2, t_1)(c_1)$ contains all the color instances of t_2 whose enabling requires more tokens in some place p than is allowed in the same place for the enabling of (t_1, c_1). This relation only accounts for the case where the inhibitor arc is between p and t1, but we are as well interested in the case where it relates p to t2. Thus, the relation ME that we want to define must be such that :

$(t_1, c_1)\ ME\ (t_2, c_2) \Leftrightarrow$

$c_2 \in mutex(t_2, t_1)(c_1)$ or $c_1 \in mutex(t_1, t_2)(c_2)$

i.e., such that if either t_1 or t_2 is related to some place by an inhibitor arc, the other transition is related to p by an input arc, and the valuations forbid the simultaneous enablings of t_1 and t_2.

An equivalent expression of this relation is :

$(t_1, c_1)\ ME\ (t_2, c_2) \Leftrightarrow$

$$c_2 \in [mutex(t_2, t_1) \cup mutex(t_1, t_2)^t](c_2)$$

This can be also written as

$$ME(t_2, t_1) = mutex(t_2, t_1) \cup mutex\ (t_1, t_2)^t$$

The following trivial property accounts for the symmetry of structural mutual exclusion.

Proposition 4.1 $ME(t_2, t_1) = ME(t_1, t_2)^t$

4.3 Structural Conflict

A structural conflict occurs when the firing of a transition (t_2, c_2) may disable the firing of a previously enabled transition (t_1, c_1). A necessary condition is that the firing of (t_2, c_2) decrements the marking of an input place or increments the marking of an inhibition place of (t_1, c_1).

The definition of structural conflict can be used for an efficient implementation of simulation. Actually, this relation makes it possible to optimize the management of the event list. After the firing of an arbitrary transition (t_2, c_2), only those transitions (t_1, c_1) that are in structural conflict with (t_2, c_2) and not in mutual exclusion with (t_2, c_2) should be tested for erasing from the event list. Such a management of the event list has proved efficient in most case studies [1].

We define the symbolic relation SC by :

$$SC\ (t_2, t_1) = \bigcup_{p \in P} \Big(\overline{W^-(p, t_2) - W^+(p, t_2)}^{\,t} \circ \overline{W^-(p, t_1)} \cup \overline{W^+(p, t_2) - W^-(p, t_2)}^{\,t} \circ \overline{H(p, t_1)} \Big)$$

The first term of the expression which defines SC corresponds to the colour instances of t_2 whose firing decrements a colour of p that is an input of t_1. The second term of this expression corresponds to the colour instances of t_2 whose firing increments a colour of p that inhibits t_1.

This relation of structural conflict is non symmetric, unreflexive and intransitive. However, it can be made symmetric rather easily. In fact, what we want is a relation SSC such that :

(t_1, c_1) SSC (t_2, c_2) / $[c_2 \in SC(t_2, t_1)(c_1)$ or $c_1 \in SC(t_1, t_2)(c_2)]$ and $c_2 \notin ME(t_2, t_1)(c_1)$

This relation can be expressed in a formal way by :

$$SSC(t_2, t_1) = [SC(t_2, t_1) \cup SC^t(t_1, t_2)] \cap {}^cME(t_2, t_1)$$

Other conflict notions can be defined. For instance, effective conflict [10] uses the reachability graph and checks if there exists some marking in which two transitions are enabled, and after the firing of one of them, the other one is no longer enabled. But such relations are very expensive to check and moreover, they are not structural because they depend on the initial marking.

4.4 Extended Conflict Sets

Relation SSC is symmetric. Thus, if we compute its reflexive and transitive closure, we obtain an equivalence relation that can be used to partition immediate transitions into classes of possibly conflicting transitions that are called extended conflict sets.

$$ECS = SSC^*$$

Extended conflict sets can be used to improve the construction of the reachability graph of a GSPN. Actually, in the case where the net is confusion free, immediate transitions belonging to different ECS can be fired in parallel, allowing to reduce the space needed for storing the vanishing markings [10].

4.5 Causal Connection

Causal connection accounts for the fact that the firing of a transition (t_1, c_1) may enable another transition (t_2, c_2). Similarly to structural conflict, causal connection can be used in the implementation of simulation. After the firing of (t_1, c_1), only those transitions (t_2, c_2) that are causally connected to (t_1, c_1) should be tested for inclusion in the event list.

4.5.1 Causal Connection Conditioned:

Confusion arises when the firing of a transition (t_1, c_1) that does not belong to the ECS of a given transition (t_k, c_k) may enable another transition (t_2, c_2) that belongs to the conflict set of (t_k, c_k), i.e., such that c_2 belongs to $ECS(t_2, t_k)(c_k)$. So we need to define a relation of causal connection between transitions (t_1, c_1) and (t_2, c_2) conditioned on the fact that a third transition (t_k, c_k) is enabled, and its enabling condition is not affected by the firing of transition (t_1, c_1).

This relation between (t_1, c_1) and (t_2, c_2) depends on the colour domain of t_k. So we extend colour and powerset functions so that we can take into account the colour domain of t_k.

Notation : we will use $\langle a, b \rangle$ to denote an element of the Cartesian product A x B.

Definition 4.5 Let $x \in Bag(E)$, $y \in Bag(F)$. Then $\langle x, y \rangle \in Bag(E \times F)$ is defined by :

$$\langle x, y \rangle = \sum_{\langle a, b \rangle \in E \times F} [x(a).y(b)] . \langle a, b \rangle$$

Example : $2.a + 3.b \in Bag(E)$, $c + 2.d \in Bag(F)$.

$$\langle 2.a + 3.b, c + 2.d \rangle = 2.\langle a, c \rangle + 4.\langle a, d \rangle + 3.\langle b, c \rangle + 6.\langle b, d \rangle$$

Definition 4.6 Let $A \in P(E)$, $B \in P(F)$. Then $\langle A, B \rangle \in P(E \times F)$ is defined by :

$$\langle A, B \rangle = \bigcup_{\langle a, b \rangle \in A \times B} \{\langle a, b \rangle\}$$

Definition 4.7 Let f be a function from Bag(E) to Bag(F), X be a function from Bag(G) to Bag(G). Then $\langle f, X \rangle$ is a function from Bag(E x G) to Bag(F x G) defined by :

$$\langle f, X \rangle(\langle a, b \rangle) = \langle f(a), b \rangle$$

A similar definition applies to powerset functions.

Definition 4.8 Let f be a function from P(E) to P(F) and X be a function from P(G) to P(G). Then $\langle f, X \rangle$ is a function from P(E x G) to P(F x G) defined by :

$$\langle f, X \rangle(\langle a, b \rangle) = \langle f(a), b \rangle$$

Definition 4.9 Let f be a function from Bag(E) to Bag(F), g be a function from Bag(G) to Bag(F) and X be a function from Bag(G) to Bag(G).
Let *op* be an operator on Bag(F). Then $\langle f \; op \; g, X \rangle$ is a function from Bag(E x G) to Bag(F x G) defined by :

$$\langle f \; op \; g, X \rangle(\langle a, b \rangle) = \langle f(a) \; op \; g(b), b \rangle$$

Now, a first condition for the firing of (t_1, c_1) to enable (t_2, c_2) is that the firing of (t_1, c_1) adds some tokens to an input place of (t_2, c_2). But knowing that (t_k, c_k) is enabled, this condition can be refined. The firing of (t_1, c_1) must add tokens to an input place p of (t_2, c_2) such that the firing of (t_2, c_2) requires more tokens in p than the firing of (t_k, c_k).

For every place p, the colors c of p, whose marking is increased when firing (t_1, c_1), verify :

$$W^+(p, t_1)(c_1)(c) > W^-(p, t_1)(c_1)(c)$$

For every place p, the colors c of p such that the firing of (t_2, c_2) requires more tokens c in p than the firing of (t_k, c_k) verify :

$$W^-(p, t_2)(c_2)(c) > W^-(p, t_k)(c_k)(c)$$

According to Definition 4.9,

$$\langle W^-(p, t_2) - W^-(p, t_k), X \rangle (\langle c_2, c_k \rangle) = \langle W^-(p, t_2)(c_2) - W^-(p, t_k)(c_k), c_k \rangle$$

$$= \sum_{c \in C(p)} [W^-(p, t_2)(c_2)(c) - W^-(p, t_k)(c_k)(c)].\langle c, c_k \rangle$$

(Definition 4.5)

Hence,

$$\overline{\langle W^-(p, t_2) - W^-(p, t_k), X \rangle} (\langle c_2, c_k \rangle)$$
$$= \{ \langle c, c_k \rangle \in C(p) \times C(t_k) \mid W^-(p, t_2)(c_2)(c) > W^-(p, t_k)(c_k)(c) \}$$

Thus, we retain only the colors c of p such that the firing of (t_2, c_2) requires more tokens c in p than the firing of (t_k, c_k).

Now consider an element c_1 of $C(t_1)$. By Definition 4.7, we have :

$$\langle W^+(p, t_1) - W^-(p, t_1), X \rangle (\langle c_1, c_k \rangle) = \langle [W^+(p, t_1) - W^-(p, t_1)](c_1), c_k \rangle$$

Hence,

$$\overline{\langle W^+(p, t_1) - W^-(p, t_1), X \rangle} (\langle c_1, c_k \rangle)$$
$$= \{ \langle c, c_k \rangle \in C(p) \times C(t_k) \mid W^+(p, t_1)(c_1)(c) > W^-(p, t_1)(c_1)(c) \}$$

and

$$\overline{\langle W^+(p, t_1) - W^-(p, t_1), X \rangle}^{\,t} (\langle c, c_k \rangle)$$
$$= \{ \langle c_1, c_k \rangle \in C(t_1) \times C(t_k) \mid W^+(p, t_1)(c_1)(c) > W^-(p, t_1)(c_1)(c) \}$$

Hence, the following symbolic relation allows us to compute a first set of couples $\langle c_1, c_k \rangle$ such that the firing of (t_1, c_1) may enable (t_2, c_2).

$$CC1(t_k)(t_1, t_2) = \overline{\langle W^+(p, t_1) - W^-(p, t_1), X \rangle}^{\,t} \circ \overline{\langle W^-(p, t_2) - W^-(p, t_k), X \rangle}$$

A second case where the firing of (t_1, c_1) may enable (t_2, c_2) is when the firing of (t_1, c_1) subtracts tokens to an inhibition place of (t_2, c_2). Knowing that (t_k, c_k) is enabled, this condition can be refined. The firing of (t_1, c_1) must subtract tokens c to an inhibition place p of (t_2, c_2) such that the maximal number of tokens c in p for which (t_2, c_2) is enabled is less than the maximal number of tokens c in p for which (t_k, c_k) is enabled.

Let x(c) be the maximum number of tokens c in p such that (t_2, c_2) is enabled. If there is an inhibitor arc between p and t_k, let y(c) be the maximum number of tokens c in p such that (t_k, c_k) is enabled. In that case, $y(c) - x(c) > 0$ accounts for the fact that the maximum number of tokens c in p for which (t_2, c_2) is enabled is less than the maximum number of tokens c in p for which (t_k, c_k) is enabled. If there is no inhibitor arc between p and t_k, this is true as soon as $x(c) > 0$.

Definition 4.10 Let x and y be two elements of Bag(E). Then $(x \lceil y)$ is an element of Bag(E) defined by : $(x \lceil y)(c) =$ if $y(c) = 0$ then $x(c)$
else if $y(c) > x(c) > 0$
then $y(c) - x(c)$ else 0.

Thus, our condition is $[H(p,t_2)(c_2) \lceil H(p,t_k)(c_k)] > 0$.

Applying the same procedure as for the first condition, $\overline{\langle H(p, t_2) \quad H(p, t_k), X \rangle} (\langle c_2, c_k \rangle)$ gives the set of tokens $\langle c, c_k \rangle$ of $C(p) \times C(t_k)$ such that the maximum number of tokens c in p for which (t_2, c_2) is enabled is less than the maximum number of tokens c in p for which (t_k, c_k) is enabled.

$\overline{\langle W^-(p, t_1) - W^+(p, t_1), X \rangle}^{\,t} (\langle c, c_k \rangle)$ gives the set of tokens $\langle c_1, c_k \rangle$ of $C(t_1) \times C(t_k)$ such that the firing of (t_1, c_1) decreases the number of tokens c in p.

As a consequence, the following symbolic relation allows us to compute a second set of couples $\langle c_1, c_k \rangle$ such that the firing of (t_1, c_1) may enable (t_2, c_2) :

$$CC2(t_k)(t_1, t_2) = < \overline{W^-(p, t_1) - W^+(p, t_1)} , X >^t \circ < \overline{H(p, t_2) \quad H(p, t_k)}, X >$$

To summarize, knowing that (t_k, c_k) is enabled, a sufficient condition for (t_2, c_2) to be enabled after the firing of (t_1, c_1) is that $<c_1, c_k> \in [CC1(t_k)(t_1, t_2) \approx CC2(t_k)(t_1, t_2)](<c_2, c_k>)$. Among these couples, we want to retain only those such that (t_1, c_1) and (t_2, c_2) are not mutually exclusive. This set can be obtained by applying the function $<^cME(t_1, t_2), X>$ to $<c_2, c_k>$. Finally, the causal connection relation $CC(t_k)(t_1, t_2)$ that allows us to compute the set of transitions (t_1, c_1) that can directly enable (t_2, c_2) knowing that (t_k, c_k) is enabled is given by :

$$CC(t_k)(t_1, t_2) = <^cME(t_1, t_2), X> \cap [CC1(t_k)(t_1, t_2) \cup CC2(t_k)(t_1, t_2)]$$

4.5.2 Causally Connected Set: The causally connected set of a transition (t_2, c_2) is the set of all transitions (t_1, c_1) whose firing can directly or *indirectly* enable (t_2, c_2), knowing that a third transition t_k is enabled. It is obtained with the transitive closure of the relation $CC(t_k)$.

$$CCS(t_k) = CC(t_k)^+$$

4.6 Confusion

A sufficient condition for the absence of confusion in a net is that a transition belonging to the conflict set of another transition (t, c) can be causally connected only with transitions that are in mutual exclusion with (t, c).

In other words, if we consider a transition (t_2, c_2) that belongs to the conflict set of another transition (t_1, c_1), any transition that may enable (t_2, c_2) is in mutual exclusion with (t_1, c_1). Due to the definition of the causal connection relation, $CCS(t_1)(t_3, t_2)(<c_2, c_1>)$ contains for every transition t_3 the set of colors of t_3 that may enable (t_2, c_2) and that are *not* in mutual exclusion with (t_1, c_1). Hence, if this set is empty, any transition (t_3, c_3) that may enable (t_2, c_2) is in mutual exclusion with (t_1, c_1) and the net is confusion-free.

This condition can be expressed formally :

$$\forall\ t_1 , t_2, t_3 , CCS(t_1)(t_3, t_2) \circ <ECS(t_2, t_1), X> = \emptyset$$

where X is defined on $P[C(t_1)]$.

NB : if the relation is applied for $t_1 = t_2$, c_1 must be withdrawn from $ECS(t_1, t_1)(c_1)$ before applying $CCS(t_1)$.

5 Efficiency of the Approach

If the color functions of the net have no structure at all, the definition of structural relations by means of symbolic relations may not offer any improvement compared with the handling of the unfolded net. For instance, without any structure, the cost of computing $\overline{f}$ might be equivalent to the cost of unfolding the net.

But if we consider a subclass of colored nets, with a restricted set of basic color functions and arc expressions being linear combinations and/or Cartesian products of these basic functions, the approach can be very efficient. Actually, once the basic powerset function $\overline{f}$ has been computed for every basic color function f, the expression of $\overline{f}$ for an arc expression will in turn be a Cartesian product of the basic powerset functions $\overline{f}$.

Besides, the detection of confusion only uses a restricted set of operations, namely intersection, union, composition and transitive closure. If the result of the application on basic powerset functions of all these operations is still a basic powerset function, then the whole algorithm handles only basic powerset functions.

As a consequence, one of the parameters that influences the efficiency of the algorithm is the size of the set of basic color functions. This approach has already been successfully applied to Unary Regular Nets [7], which are a restricted class of colored nets. In this case, the algorithm handles only four functions. We are now working on the extension of the algorithm to Well-Formed nets [13] that have the same modelling power as general colored nets. This extension mainly consists in introducing predicates and the successor function.

One of the most interesting features of these two classes of colored nets is that the definition of their color functions does not depend on the cardinality of the color domains. As a consequence, the results of the algorithm are parameterized and instead of applying the confusion detection algorithm several times for different nets with the same structure, a single application provides results that hold for a family of colored nets that differ only by the cardinality of the color domains.

6 Conclusions

We have presented in this paper a new approach for computing structural relations in colored GSPNs. This approach is based on the use of symbolic relations that directly handle the color functions labelling the arcs of the net, thus avoiding the unfolding of the net. We have applied this technique to the detection of confusion.

The use of symbolic relations results in important savings in the cost of computing structural relations. And a very interesting advantage of the technique is that the structural relations expressed with symbolic relations are parameterized, i.e., their expression does not depend on the cardinality of the color domains of the nodes. As a consequence, they hold for a family of nets that have the same structure and different sizes of color domains.

We have already shown the efficiency of the symbolic relation approach for a restricted class of colored nets, namely Unary Regular Nets. We are now working on the extension of the results to Well-Formed Nets, which have the same modelling power as general colored nets.

References

[1] G. Chiola, "A Simulation Framework for Timed and Stochastic Petri Nets", IBP-MASI Research Report n° 90.50, October 1990.

[2] M. Ajmone Marsan, G. Balbo, G. Conte, "Performance Models of Multiprocessor Systems", MIT Press, 1986.

[3] M. Ajmone Marsan, G. Balbo, G. Chiola, G. Conte, "Generalized Stochastic Petri Nets Revisited : Random Switches and Priorities", in proc. of PNPM 87, IEEE CS Press, pp 44-53, Madison USA, 1987.

[4] H.J. Genrich, "Predicate/Transition Nets", in High-level Petri Nets, K. Jensen & G. Rozenberg eds., Springer-Verlag, pp 3-43, 1991.

[5] K. Jensen, "Coloured Petri Nets and the Invariant Method", Theoretical Computer Science 14, pp 317-336, 1981.

[6] G. Chiola, G. Bruno, T. Demaria, "Introducing a Color Formalism into Generalized Stochastic Petri Nets", in proc. of 9th European Workshop on Application and Theory of Petri Nets, pp 202-215, Venice IT, 1991.

[7] C. Dutheillet, S. Haddad, “An Efficient Computation of Structural Relations in Unary Regular Nets”, Proceedings of the 7th International Symposium on Computer and Information Sciences (ISCIS VII), Antalya, Turkey, 1992, pp 73 - 79.

[8] M. Nielsen, G. Plotkin, G. Winskel, "Petri Nets, Event Structures and Domains", in proc. of Semantics of Concurrent Computation, LNCS 70, Springer-Verlag, pp 266-284, 1979.

[9] C.A. Petri, "General Net Theory. Communication Disciplines", in proc. Joint IBM University of Newcastle Seminar, B. Shaw ed., Newcastle GB, 1976.

[10] G. Balbo, G. Chiola, G. Franceschinis, G. Molinar Roet, "On the Efficient Construction of the Tangible Reachability Graph of Generalized Stochastic Petri Nets", in proc. of PNPM 87, IEEE CS Press, pp 136-145, Madison USA, 1987.

[11] A.V. Aho, J.E. Hopcroft, J.D. Ullman, "The Design and Analysis of Computer Algorithms", Addison-Wesley, 1974.

[12] C. Dutheillet, S. Haddad, "Structural Analysis of Colored Nets. Application to the Detection of Confusion", IBP-MASI Research Report 92.16, 1992.

[13] G. Chiola, C. Dutheillet, G. Franceschinis, S. Haddad, "On Well-Formed Colored Nets and Their Symbolic Reachability Graph", in High-level Petri Nets, K. Jensen & G. Rozenberg eds., Springer-Verlag, pp 373-396, 1991.