ISSN 2047-3338

Maximality-based Step Graph for Safety-Petri Nets

Adel Benamira and Djamel-Eddine Saidouni

***Abstract*—This paper proposes for safety-petri nets an algorithm for reducing on the fly a Maximality-based Labeled Transition Systems via partial order technique, in which made possible the consideration of the branches, therefore the reduction is important. The reduction graph (Maximality-based Step Graph) is a complete graph preserving the general properties (deadlock states and liveness).**

***Index Terms*—Formal Method, Petri Nets, Partial Order Semantics, Maximality Semantics and Maximality-Based Labeled Transitions Systems**

I. INTRODUCTION

THE state space generation is the first step of verification methods for concurrent systems. This paper contributes to the resolution of the state space combinatorial explosion problem. More particularly, our interest concerns the state space explosion due to the representation of parallelism by the interleaving execution of concurrent actions, which generates several execution sequences starting from the same state and finishing in another one, where the order of execution is arbitrary. Partial order techniques seek to eliminate superfluous interleaving while being based on the independency relations directly calculated from the formal specification of the system to analyze, e.g. in Petri net of Fig. 1 (a), we have the independency relations as $\iota=\{(a,b)\}$.

In general two strategies may be distinguished: the first one is based on the elimination of interleaving and the second one is based on the covering steps. The various techniques of the first strategy try to obtain a sub-graph of the state space, containing less possible equivalent sequences [1], [2], [3]. This approach was generalized in [4]-[7], which revealed the concepts of persistent sets and sleep sets. Their principal weakness is the indeterminism of the obtained result, where several sub-graphs may be generated for the same state space [8] (Fig. 1 (c)). The second approach was proposed in [9], [10], [11], in which, we regroup independent events in only one step (Fig. 1 (d)). The built graph is referred as Covering Step Graph (CSG) [10], which is a complete graph. Deadlock and liveness properties are preserved; however, several versions were proposed to preserve observational equivalence [10], and failure semantics [12].

In the both strategies, the calculated independency relation is structural. Consequently, we can build and on the fly states graph without superfluous interleaving, thus, this superfluous was detected previously.

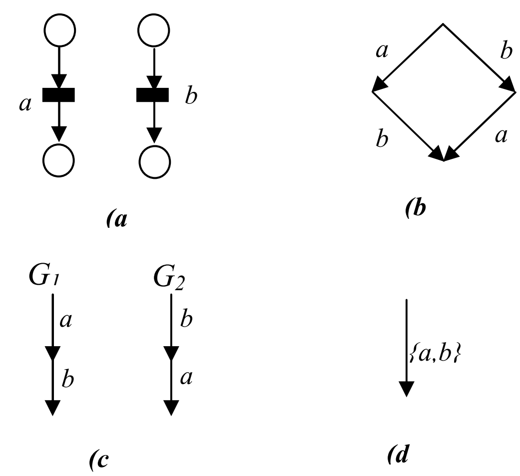


Figure 1. Transition Systems of the behavior expression $a|||b$

Unfortunately, the partial order approaches cannot exploit all the independency relations. Therefore, there are cases where it is impossible to take independent transitions in the same step (or to eliminate some equivalent sequences) to the risk to lose deadlock preservation. Among these cases, one can quote differed conflict (see Fig. 2) where its strong presence decreases the reduction ratio. Indeed, branches are not considered any more in the possible reductions.

In this paper, we propose for safety-Petri nets a reduction method modulo partial order technique which use the Maximality-based Labeled Transitions Systems model (MLTS) [13], [14] as states graph model. We prove that it possible to answer the limit quoted above. Note that the MLTS model has been used in work relating to the specification and the verification of concurrent systems [15]-[25].

A. Benamira is with Computer Science Dept., University of 08 Mai 45, 24000 Guelma and MISC Laboratory; (e-mail: benamirar@misc-umc.org).

D. E. Saidouni is with MISC Laboratory, Computer Science Dept., University of Mentouri, 25000 Constantine, Algeria; (e-mail: saidouni@misc-umc.org).

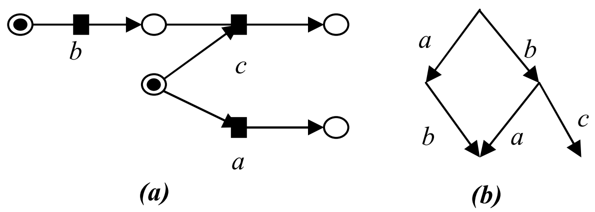


Figure 2. Differed conflict

The MLTS model can be used as a semantic representation of systems behaviors. Hence, various specification models may be used (RdP [26], CCS [27], LOTOS[28],. . .); for that, it is enough to define semantics in MLTS term for each one.

Let us take for example the MLTS of Fig. 3 (a) representing the behavior of Fig.2.(a). In the initial state, no action is in execution. Transition t_1 (resp t_2) represents the beginning of execution (identified by event *x* (resp *y*)) of the action *a* (resp *b*). In state *1* (resp *2*), action *a* (resp *b*) is potentially in execution, this is represented respectively by the events *x* and *y* known as maximal in this state. In state *2*, the occurrence of *c* is conditioned by the termination of *b*, which is translated by the presence of the event *y* on the level of the transition t_5; therefore *z* is the only maximal event in state *4*. In state *3*, events *x* and *y* are maximal, i.e. in this state the corresponding actions (*a* and *b*) can be in execution. For more detail of the MLTS model [14].

In this paper, we prove that reducing state graph within differed conflict is possible through maximal events concept. e.g., Fig. 3(a) may be reduced, as a result, we obtain the MLTS represented by Fig. 3(b).

The paper is organized as follows. Section 2 presents preliminaries definitions. In Section 3, we present reduction of MLTS modulo partial order technique. On the fly maximal step graph generation algorithm is presented in Section 4. In section 5, we present a brief descript of the implementation of our technique and we discuss the obtained results. The paper is enclosed by conclusion.

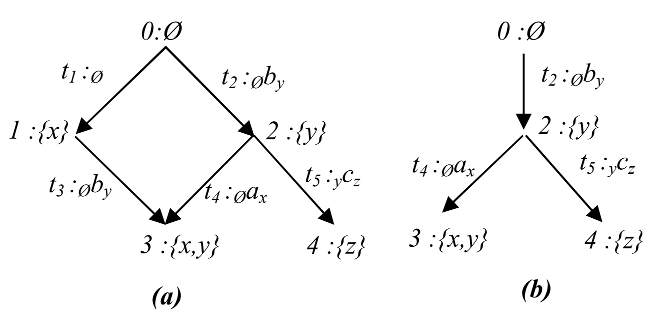


Figure 3. MLTSs

II. PRELIMINARIES

We briefly recall the definitions of some basic concepts necessary in the following sections.

A. Petri Nets Related Definitions

- A Petri net is a tuple (S, T, W) where S is the set of places, T is the set of transitions such that $S \cap T = \emptyset$, and $W: ((S\times T) \cup (T\times S)) \to \mathbf{N} = \{0,1,2,...\}$ is the weight function. Graphically, transitions of T are represented by rectangles, places of S by circles and weight function by arrows associated with their weights. We suppose that all nets are finite, i.e. $|S \cup T| \in \mathbf{N}$.
- For $x \in S \cup T$, the pre-set ${}^\bullet x$ is defined by ${}^\bullet x = \{y \in S \cup T | W(y,x) \neq 0\}$ and the post-set $x^\bullet$ is defined by $x^\bullet = \{y \in S \cup T | W(x,y) \neq 0\}$.
- The *marking* of a Petri net (S,T,W) is defined as a function $M: S \to \mathbf{N}$. A marking is generally represented graphically by putting tokens in places.
- Safety-Petri net is a Petri net (S,T,W) such that for any *s of S*: $M(s) \leq 1$.
- The transition rule stipulates that a transition t is enabled by M iff $M(s) \geq W(s,t)$ for all $s \in S$. The firing of a transition t will produce a new marking M' defined by $M'(s)=M'(s)-W(s,t)+W(t,s)$ for all $s \in S$. The occurrence of t is denoted by $M[t>M'$.
- Two transitions t_1 and t_2 (not necessarily distinct) are concurrently enabled by a marking M iff $M(s) \geq W(s,t_1)+W(s,t_2)$ for all $s \in S$.
- A marked Petri net (S,T,W, M_0) is a Petri net (S,T,W) with an initial marking M_0.
- An alphabet A is a finite set; we suppose that $\tau \notin \mathrm{A}$ (τ will indicate invisible action, or *silent action*).
- The labeling of a Petri net $N=(S,T,W)$ is a function $\lambda : T \to \mathrm{A} \cup \{\tau\}$. If $\lambda(t) \in \mathrm{A}$ then t is said to be *observable* or *external*; at the opposite, t is *silent* or *internal*.
- $\Sigma=(S,T,W, M_0,\lambda)$ is a labeled system iff $(S,T,W, M_0)$is a marked Petri net and λ is a labeling function of (S,T,W).
- An action $a \in \mathrm{A}$ of a system $\Sigma=(S,T,W, M_0,\lambda)$ is *auto-concurrent* in a marking M iff M concurrently enables two observable transitions t_1 and t_2 (not necessarily distinct) such that $\lambda(t_1)= \lambda(t_2)=a$.
- A sequence $\sigma=M_0t_1M_1t_2...$ is an occurrence sequence iff $M_{i-1}[t_i>M_i$ for $1\leq i$. A sequence $t_1t_2...$ is a transition sequence starting with M iff there is an occurrence sequence $M_0t_1M_1t_2....$ If a finite sequence $t_1t_2...t_n$ leads from M to M', we write $M[t_1t_2...t_n>M'$. The set of reachable markings of a marked Petri net (S,T,W,M_0) is defined as $[M_0>=\{M \mid \exists\, t_1t_2...t_n : M_0[t_1t_2...t_n>M\}$.

B. Maximality-based labeled transition systems [13], [14]

1) Definition of MLTS

Let **M** be a countable set of event names, a maximality-based labeled transition system of support **M** is a tuple $(\Omega,\lambda,\mu,\xi,\psi)$ with:

1. $\Omega = <S,T,\alpha,\beta,s_0>$ is a transition system such that:
   - S is the set of states in which the system can be found, this set can be finite or infinite.

- T is the set of transitions indicating state switch that the system can achieve, this set can be finite or infinite.
- α and β are two applications of T in S such that for all transition t we have: α (t) is the origin of the transition and β (t) its goal.
- S_0 is the initial state of the transition system Ω.

2. (Ω,λ) is a transition system labeled by the function λ on an alphabet *Act* called support of (Ω, λ). In the other word $\lambda : T \rightarrow Act$.
3. $\psi : S \rightarrow 2^{\mathbf{M}}$ is a function which associates to each state the finite set of maximal event names present in this state.
4. $\mu : T \rightarrow$ is a function which associates to each transition the finite set of event names corresponding to actions that have already begun their execution and the end of their executions enables this transition.
5. $\xi : T \rightarrow \mathbf{M}$ is a function which associates to each transition the event name identifying its occurrence.

Such that $\psi(s_0)=\emptyset$ and for all transition t, $\mu(t) \subseteq \psi(\alpha(t))$, $\xi(t) \notin \psi(\alpha(t)) - \mu(t)$ and $\psi(\beta(t)) = (\psi(\alpha(t)) - \mu(t)) \cup \{\xi(t)\}$.

2) α−equivalent relation

The purpose of this relation, it's to put in correspondence MLTSs describing the same behavior of which the only difference resides in the choice of event names. For example, both MLTSs of Fig. 4 describes the same behavior (the parallel execution of actions *a* and *b*), we can obtain the MLTS of Fig. 4(a) from that of Fig. 4(b) by substituting event names *e* by *x* and event name *z* by *y*.

Définition 2.1 "α-equivalent": Let $=_\alpha$ be the smallest relation over MLTSs such as $mlts_1 =_\alpha mlts_2$ if and only if :

- $mlts_1 \cong mlts_2$ *(Isomorphism)*. or
- $mlts_1 \cong \sum_{i\in I} {}_{M_i} a_{i\,x_i} T_i$, $mlts_2 \cong \sum_{j\in J} {}_{M_j} a_{j\,x_j} T_j$, and
- $\psi(S) = \psi(T)$, and there is a bijection $f : I \rightarrow J$ such as, for any $i \in I$, $\mathbf{M} = \mathbf{M}_{f(i)}$, $a_i = a_{f(i)}$, and
  - ✓ $x_i = x_{f(i)}$ *and* $T_i =_\alpha T_{f(i)}$,
  - ✓ $x_{f(i)} \notin \psi(T_i)$ and $T_i[x_{f(i)}/x_i] =_\alpha T_{f(i)}$.

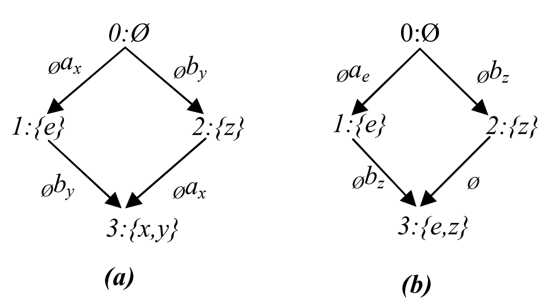


Figure 4. Two MLTSs α-equivalents

A reduction consists to eliminate the redundant via certain relations by preserving properties to be checked. In this section, we will use the α-relation as a criterion of redundant behaviors. As illustration, the MLTS of Fig.5.(a) represents the behavior in which we have two sub-MLTSs S_1 and S_2 of Fig.5.(a) are α−equivalent. Indeed, it exists two functions of substitution $\sigma_1=\{x/x,y/y,z/z\}$ and $\sigma_2=\{x/v,y/u,z/e\}$ such as $S_1\sigma_1 \cong S_2\sigma_2$. To remove such a redundancy, we must, initially, apply the substitution function $\sigma_1 \cup \sigma_2$ to the MLTS of Fig. 5(a), group the start stats of $S_1$and S_2, and then, we remove S1σ1 or S2σ2. As a result we obtain the MLTS of Fig. 5(b).

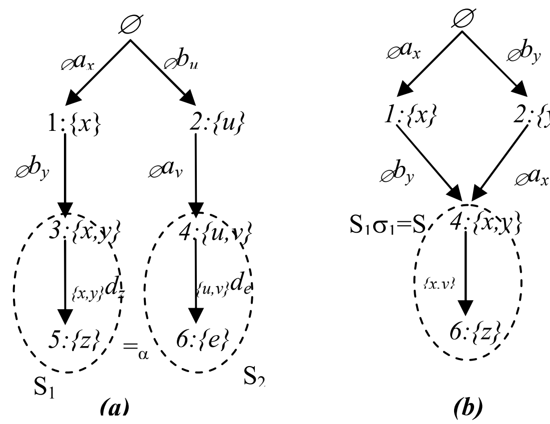


Figure 5. Reduction modulo α-equivalent

C. Safety-Petri Nets and Maximality Semantics

In [21], we have maximality semantics for Petri nets, the α-equivalent over Petri nets is not solved (for safety-Petri nets is trivial) until now, unfortunately this equivalent is important to build a MSG. Therefore, the building of MSG is restricted to Safty-Petri nets. In the following, we propose a restriction of [21] for safety-Petri nets.

Let (S,T,W) be a safety-Petri net with a marking *M*:

1. The set of maximal event names in $\mathbf{M}$ is the set of all event names identifying bound tokens in the marking *M*. Formally, the function ψ will be used to calculate this set, it can be defined as $\psi(M) = \cup_{s\in S} \cup_{i=1,\dots,ms} xs_i$ such that $M(s)=(FT,BT)$ with $BT=\{(t,x)\}$.
2. Let $N \subset M$ be a non-empty finite set of event names, *makefree*(N,*M*) is defined recursively by:
   - *makefree*($\{x_1,x_2,\dots,x_n\}$,*M*)=*makefree*($\{x_2,\dots,x_n\}$, *makefree*($\{x_1\}$,*M*)).
   - *makefree*($\{x\}$,*M*)=*M'* such that for all $s \in S$, if $M(s)=(FT,BT)$ then:
     - If there is $(t,x)=BT$ then $M'(s) = (FT+1, \emptyset)$ (Conversion of *BT* bound tokens identified by the event name *x* to free tokens).
     - Otherwise, $M'(s)=M(s)$.
3. Let *t* be a transition of *T*; *t* is said to be *enabled* by the marking *M* iff $\|M(s)\| \geq W(s,t)$ for all $s \in S$. The set of all transitions enabled by the marking *M* will be noted *enabled*(*M*).

4. The marking M is said to be minimal for the firing of the transition t iff $|M(s)|=W(s,t)$ for all $s\in S$.
5. Let M_1 and M_2 be two markings of the Petri net (S,T,W). $M_1€M_2$ iff $\forall\ s\in S$, if $M_1(s) = (FT_1,BT_1)$ and $M_2(s)=(FT_2,BT_2)$ then $FT_1 \leq FT_2$ and $BT_1\leq BT_2$.
6. Let M_1 and M_2 be two markings of the Petri net (S,T,W) such that $M_1€M_2$. The difference M_2-M_1 is a marking M_3 (M_2-M_1=M_3) such that for all $s\in S$, if $M_1(s)=(FT_1,BT_1)$ and $M_2(s)=(FT_2,BT_2)$ then $M_3(s)= (FT_3,BT_3)$ with $FT_3=FT_2-FT_1$ and if $(t,x) \in BT_1$ and $(t,x) \in BT_2$ then $(t,x) \notin BT_3$.
7. $Min(M,t)=\{M'|M'\leq M\}$ and $M'$is minimal for the firing of t.
8. Let **M** be a set. The function get:$2^{\mathbf{M}}$ -{Ø}→M is a function which satisfies $get(E) \in E$ for any $E\in 2^{\mathbf{M}}$ -{Ø}.
9. Given a marking M, a transition t and an event name $x\notin \psi(M)$, $occur(t,x,M)=M'$ such that for all $s\in S$, if $M(s)=(FT,BT)$ then $M'(s)=(FT,BT')$ with $BT'=BT\cup \{W(t,s),t,x)\}$ if $W(t,s)\neq 0$ and $BT'=BT$ otherwise. Hence, M' is the resultant marking from the addition of tokens bound to t to the marking M.

Let $\Sigma=(S,T,W, M_0,\lambda)$ be a labeled system. The marking graph Mg labeled by λ associated to Σ is a graph in which the states are defined by all reachable markings from the initial marking $M0$ and the transitions between states are labeled according to the derivation rule of Definition2.2.

Definition 2.2 Let M be a reachable marking of the marked Petri net (S,T,W,M_0,λ), $t\in enabled(M)$ then for all $M''\in Min(M,t)$, $E= \psi(M'')$ and $M'''=makefree(E,M-M'')$; the following derivation is possible: $M \xrightarrow{{}_E t_x} M'$ (also denoted by $(M,{}_Et_x,M')$) such that

1. E is the set of maximal event names associated with actions in which the end is required for the launch of the action related to the firing of t.
2. $x = get(M- \psi(M'''))$ and
3. $M' = occur(t,x,M''')$.

1. **Proposition 2.1** Let $\Sigma=(S,T,W, M_0,\lambda)$ be a labeled system and Mg its marking graph built according to Definition2.2, then the structure $\Sigma_{mlts}=(Mg,\lambda,\mu,\xi,\psi)$ is a maximality-based labeled transition system with:
4. $Mg=<Sg,Tg,\alpha,\beta,M_0>$ is the marking graph associated to Σ such that
   - Sg is the set of states defined by the set of reachable markings from the initial marking M_0.
   - $Tg=\{(M,{}_Et_x,M')\}$ such that $M,M'\in Sg$ and $(M,{}_Et_x,M')$ is a valid derivation.
   - For $(M,{}_Et_x,M')\in Tg$ we have $\alpha((M,{}_Et_x,M'))=M$ and $\beta((M,{}_Et_x,M'))=M'$.
5. $\psi : Sg\rightarrow 2^{\mathbf{M}}$ is defined as of MLTS.
6. For $d=(M, {}_Et_x,M') \in Tg$ we put $\lambda(d)= \lambda(t)$, $\mu(d)=E$ and $\xi(d)=x$.

III. Reduction of MLTS modulo order partial

In [23], we propose reduction technique of MLTS modulo order partial semantics, it is a generic solution (independent to any specification model), in which 1) we build, under certain conditions, a step allowing directly reaching the final state which would have been reached by each interlaced sequence 2) we eliminate the superfluous interleaving, in the other word, we use together the two strategies. The Fig.6 shows the obtained benefit in the case of the derivation of three parallel actions a, b and c in the presence of differed conflict. The graph of Fig.6.(b) is the step graph of the MLTS of Fig.6.(a) in which all interleaving runs were converted into two steps (p_1 and p_2); the first step expresses the beginning of execution of c and the other expresses the parallel execution of a and b. The built step graph covers the initial MLTS via the Mazurckiewicz's traces equivalence [29]. It will prove that our approach preserves deadlock states and liveness property. On the fly generation of MSG is possible.

The following definitions introduce the step concept (known as maximal step):

2. Events sequence :< _ > is a function inductively defined by:
   - $<\varepsilon> =_{def} \varepsilon$
   - $<_M a_x.p> =_{def} x.<p>$
3. Support of a transitions sequence : $\|\ \|$ is a function is defined as follows:
   - $\|\varepsilon\| =_{def} \varepsilon$
   - $\|u.w\| =_{def} \{u\} \cup \|w\|$
4. Extension of Mazurckiewicz's trace to MLTS : Let $mlts =< S, s0, T, \psi, \mu, \xi >$ be a MLTS. $U._Ma_x._Nb_y.V$ and $U._Nb_y._Ma_x.V$ are two paths of $mlts$. Let $\approx$ be the relation defined on $T^*\times T^*$ by $< U._Ma_x._Nb_y.V > \approx < U._Nb_y._Ma_x.V >$ if $x\notin N$ and $y\notin M$, by construction, $\approx$ is reflexive and symmetric. The trace equivalence $\equiv$ can be defined by the transitive closing of the relation $\approx$. Equivalence classes of $\equiv$ are called traces. $[< w >]$ the trace generated by w.
5. Maximal path : Let $mlts=< S, s_0, T, \psi, \mu, \xi >$ be a MLTS and $w \in T*$, w is a maximal path $\exists s,s'\in S, s \overset{w}{\Rightarrow} s' : \|<w>\| \subseteq \psi(s')$ and $(s' \not\rightarrow)$ $\vee (\exists t \in T : wt$ is not a maximal path)
6. Minimal path : Let C_s be a maximal paths set associated to the state s. $Min(C_s)=$ $\{c \backslash \nexists c'\in C_s : \|<c'>\| \subset \|<c>\|\}$.
7. Maximal paths equivalence: Two maximal paths w and w' are equivalent, noted $w \approx_c w_0$, if and only if: $s \overset{w}{\Rightarrow} s'$ implies that $s \overset{w'}{\Rightarrow} s'$.
   It is particular case of the relation of Mazurckiewicz's trace equivalence in which all events are independent.

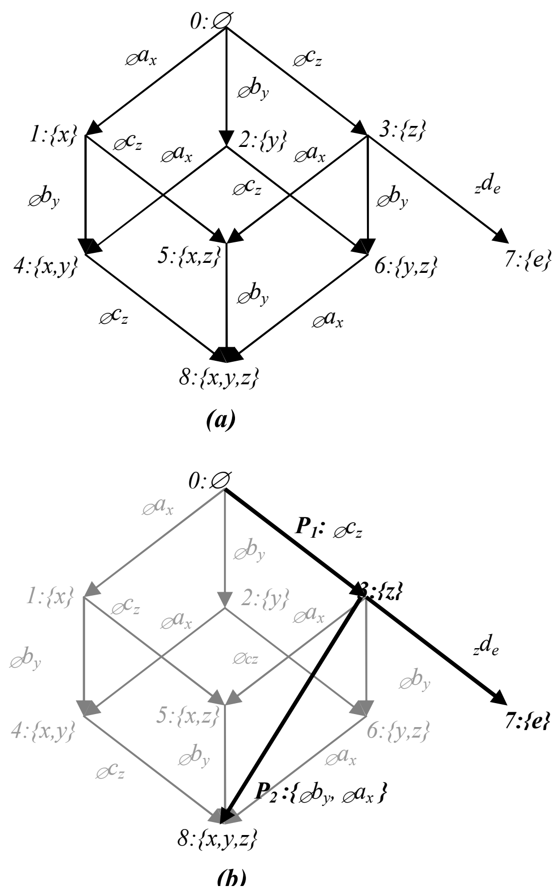


Figure 6. A MLTS and their MSG

8. Maximal step: Let *mlts*=< $S, s_0, T, \psi, \mu, \xi$ >, and $w \in T^*$, $\|w\|$ defines a step if and only if
$$: \exists s, s' \in S, w \in T^*, s \overset{w}{\Rightarrow} s' : \forall e \in \| < w > \|, \quad e \in \psi(s').$$

9. Extension of the accessibility relation to the maximal transitions steps : Let $\rightarrow_p$ be an extension of $\rightarrow$ to the maximal steps, and w be a maximal path $s \overset{w}{\Rightarrow} s'$. The associated step is $\xrightarrow{\|w\|}_p$.

10. Maximality-based Step Graph: Let *mlts* =($\Omega,\lambda,\mu,\xi,\psi$) such that Ω=<S,T,α,β,s_0> be MLTS, *msg*=($\Omega',\lambda',\mu',\xi',\psi$) such that Ω'=<$S', \Xi, \alpha,\beta,s_0$> is a MSG of *mlts* if and only if:
    1. $\forall s' \in S' : s' \in S$,
    2. $\forall t' \in \Xi$: t' is a step, where $\|t'\|$ constitute a maximal path in *mlts*.
    3. $\forall s \in S', s \xrightarrow{_M a_x} s' \in T$,
$$\forall s'' \in S', \forall w \in T^*, s' \overset{w}{\Rightarrow} s''$$
$$\Rightarrow \left\{ \exists w' \in \Xi^*, s \overset{w'}{\Rightarrow}_p s'' : [<_M a_x . w >] = [< w' >] \right\}$$

Such that :

- $\zeta' : 2^T \rightarrow 2^{\mathsf{M}}$:
    - $\zeta'(\varepsilon) =_{\text{def}} \varepsilon$,
    - $\zeta'(\{t\} \cup E) =_{\text{def}} \zeta(t) \cup \zeta'(E)$.
- $\mu' : 2^T \rightarrow 2^{\mathsf{M}}$:
    - $\mu'(\varepsilon) =_{\text{def}} \varepsilon$,
    - $\mu'(\{t\} \cup E) =_{\text{def}} \mu(t) \cup \mu'(E)$

Where for any step $s \xrightarrow{E}_p s'$, the following conditions are satisfied: $\psi(s') = (\psi(s) \setminus \mu'(E)) \cup \zeta'(E)$ and $\zeta'(E) \not\subseteq \psi(s) - \mu'(E)$ et $\mu'(E) \subseteq \psi(s')$

Proposition 3.1: Let $s \overset{w}{\Rightarrow} s_1$ and $s \overset{w'}{\Rightarrow} s_2$, if S_1 and S_2 are α-equivalents then w ≈w'.

Proposition 3.2: The maximal steps graph preserves deadlock states and liveness property.

Proposition 3.3: Let $s \overset{w}{\Rightarrow} s_1$ and $s \overset{w'}{\Rightarrow} s_2$ such that S_1 and S_2 are α-equivalents, if $u \in \mathrm{Min}(C_s)$ such that *w*=*u.v*, the branch *w* preserves deadlock states and liveness property.

IV. On the fly maximal step graph generation for Safety-Petri nets

The Algorithm 4.1 is a basic on the fly maximal step graph generation for safety-Petri nets which is similar to standard algorithm for computing a reachable marking graph.

The reduction resides in:

- We build a step by the Proposition 4.1 in which we check for each developed transition, if it can form part of a maximal step or it is itself a step.
- And Elimination of the superfluous interleaving, which released when we detect two states α-equivalent (see Definition 4.1), so we have diamond in which all branches are Mazurckiewicz's trace equivalent (see Proposition3.1). By Proposition3.3 we can eliminate all the superfluous interleaving and take only the branch $w=u.v$ such that $u \in \mathrm{Min}(C_s)$ and s is the head of diamond.

In the algorithm, we represent the state s_i by the marking M_i.

Proposition 4.1: Let *msg* be a MSG in generation in state *s'*, and let $s \xrightarrow{p}_p s'$ step of *msg*: for any transition generated from this state $s' \overset{t}{\rightarrow} s''$, we have:

- Either pt is step, we replace $s \xrightarrow{p}_p s'$ by $s \xrightarrow{pt}_p s''$. if $pt \in \mathrm{Min}(C_s)$
- $s' \xrightarrow{t}_p s''$ is a step of *msg*.

Definition 4.1: Let $\Sigma=(S,T,W,M_0,\lambda)$ be a labeled system. The α-equivalence relation is recursively defined over configurations as follows:

- $M(s) =_\alpha M'(s)$ iff
  - $-FT(s)=FT'(s)$, and
  - $-(t,x)= (t,x')$ such that $(t,x) \in BT(s)$ and $(t,x') \in BT'(s)$.
- $M =_\alpha M'$ iff $\forall s \in S$, $M(s) =_\alpha M'(s)$.

Algorithm4.1 ″ basic on the fly maximal step graph generation ″
Require: *R be a safety-Petri net ;*
Ensure: *msg*=$(\Omega,\lambda,\mu,\xi,\psi)$ such that Ω=<S, Ξ ,α,β,s_0> ;
Variables :
S' *: list of no treated states initialized by* s_0;
S *: list of treated states;*
X*:list of states;*
T *: list of transitions ;*
Début
1 ***While*** *S' no empty* ***Do***
2 *Select and remove an element s of S' ;*
3 *Insert s in S ;*
4 *T'←enabled (s)=* $\cup\left\{ s \xrightarrow{t_j} s_i \right\}$; $X \leftarrow \cup\{s_i\}$
5 ***For each*** $s'' \xrightarrow{p}_p s$ ***Do***
6 ***For each*** t_j *de T* ***Do*** *Build step w.r.t Proposition 4.1;*
7 ***For each*** s_i *state of X α-equivalent with s''of S* ***Do*** *implement the Proposition.3.3*
8 *Insert all new states* s_i *of X modulo α-equivalent in S' ;*
9 ***Endwhile***
FinAlgo.

As example, given the safety-Petri net of Fig. 7, we have nine iterations:

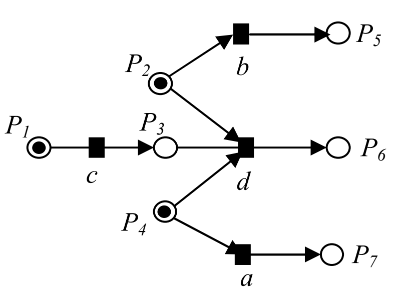


Figure 7. Differed conflict.

1. By initialization S'={M_0}, we take from line 3 S={M_0}. By Definition2.2, we have Fig. 8, with S'=X={M_1,M_2,M_3} and any states α-equivalent. With
   - M_0=[(1,Ø),(1, Ø),(0, Ø),(1, Ø),(0, Ø),(0, Ø),(0, Ø)].
   - M_1=[(1,Ø),(1, Ø),(0, Ø),(0, Ø),(0, Ø),(0, Ø),(1, (a,x_1))].
   - M_2=[(1,Ø),(0, Ø),(0, Ø),(1, Ø),(0, (b,x_2)),(0, Ø),(0, Ø)].
   - M_3=[(0,Ø),(1, Ø),(1, (c,x_3)),(1, Ø),(0, Ø),(0, Ø),(0, Ø)].

   We have any modification by the application of Proposition4.1.

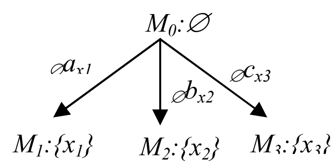
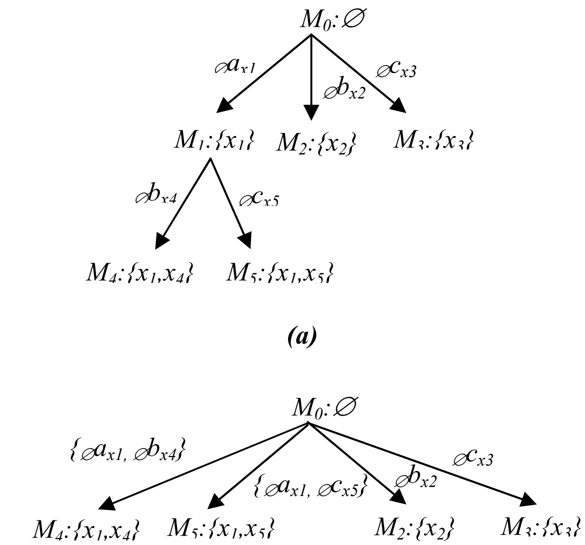


Figure 8. msg after first iteration

2. In the second iteration, we select and remove M_1 from S' to S, so S={M_0,M_1}. By Definition2.2, we have fig.9.(a), with X={M_4,M_5} and S'={M_2,M_3,M_4,M_5}. Any states α-equivalent. But, we build two steps by the use of Proposition4.1, so we remove definitively M_1 from the graph. We obtain as result the Fig.9.(b). With
   - M_4=[(1,Ø),(0, Ø),(0, Ø),(0, Ø),(0, (b,x_4)),(0, Ø),(1, (a,x_1))].
   - M_5=[(0,Ø),(1, Ø),(0, (c,x_5)),(0, Ø),(0, Ø),(0, Ø),(1, (a,x_1))].

3. In the third iteration, we select and remove M_2 from S' since S is as {M_0,M_2}. By Definition 2.2, we have X={M_6,M_7} and from Proposition4.1 we remove definitively M_2 and we build two steps. But in this iteration, we have $M_6 =_\alpha M_4$, so, we remove definitively M_6 (or M_4) and we substitute the graph by σ={x_7/x_8,x_1/x_8,x_2/x_9,x_4/x_9}. We obtain as result the Fig.10 with S={M_0} and S'={M_3,M_4,M_5 ,M_7}. With
   - M_6 =[(1,Ø),(0, Ø),(0, Ø),(0, Ø),(0, (b,x_4)),(0, Ø),(0, (a,x_7)].
   - M_7 =[(0,Ø),(0, Ø),(0, (c,x_6)),(1, Ø),(0, (b,x_9)),(0, Ø),(0, Ø)].



(a)

(b)

Figure 9. msg after second iteration.

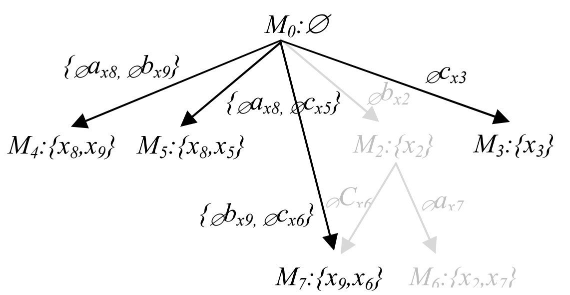


Figure 10. msg after third iteration

4. We select and remove M_3 from S'. S={M_0, M_3}. By Definition2.2, we have X={M_8,M_9}. We remark that $\emptyset c_z$ is a minimal path so by Proposition4.1 we have $\emptyset c_z$ as step. In the other hand, we have $M_7 =_\alpha M_9$ and we substitute the graph by σ={x_3/x_{10},x_9/x_{11}}. So, we obtain as result the Fig.11 with S={M_0, M_3}and S'={M_4,M_5 ,M_7 ,M_8}. With
   - M_8=[(0,Ø),(0, Ø),(0, (c,z)),(0, Ø),(0, Ø),(0, (d,e)),(0, Ø)].
   - M_3=[(0,Ø),(0, Ø),(1, (c,z)),(1, Ø),(0, (b,x)),(0, Ø),(0, Ø)].
5. We select and remove M_4 from S'. By Definition2.2, we have X={M_{10}} and any states α-equivalent. By the application of Proposition4.1 we remove definitively M_4 with building a step as Fig.12 with S={M_0, M_3}and S'={M_5,M_7 ,M_8,M_{10}}.

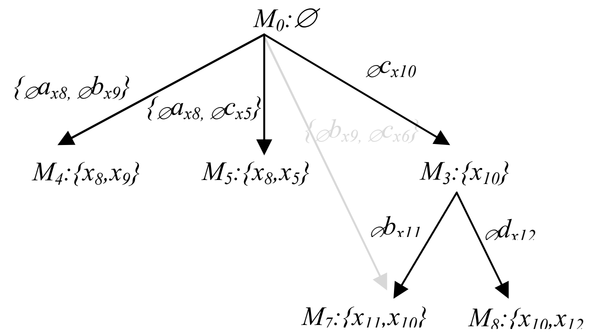


Figure 11. msg after fourth iteration

6. We select and remove M_7 from S'. By Definition2.2, we have X={M_{11}} and from Proposition4.1 we remove definitively M_7 and building a step as Fig.13. In this iteration, we have $M_{10} =_\alpha M_{11}$ and we substitute the graph by σ={x_9/x_{15},x_{13}/x_{16}, x_{14}/x_{15}, x_5/x_{16}}. So, we obtain as result the Fig.13 with S={M_0, M_3}and S'={M_5,M_8,M_{10}}.

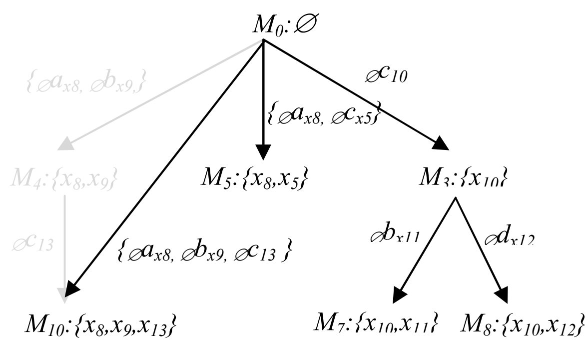


Figure 12. msg after fifth iteration

7. We select and remove M_5 from S'. By Definition2.2, we have X={M_{12}} and from Proposition4.1 we remove definitively M_5 , in this cases, we have $M_{10} =_\alpha M_{12}$ and we substitute the graph by σ={x_{15}/y,x_{11}/y,x_8/x,x_{17}/x,x_{10}/z,x_{16}/z}. So, we remove definitively M_{10} or M_{12}., we obtain as result the Fig. 14 with S={M_0, M_3}and S'={M_8,M_{10}}.
8. We select and remove M_8 from S'. We have any transition enabled with S={M_0,M_3,M_8}and S'={M_{10}}.
9. We select and remove M_{10} from S'. We have any transition enabled with S={M_0,M_3,M_8,M_{10}}and S'=Ø. So, this iteration is the last iteration with the Fig.14 is as MSG of Petri net of Fig. 7.

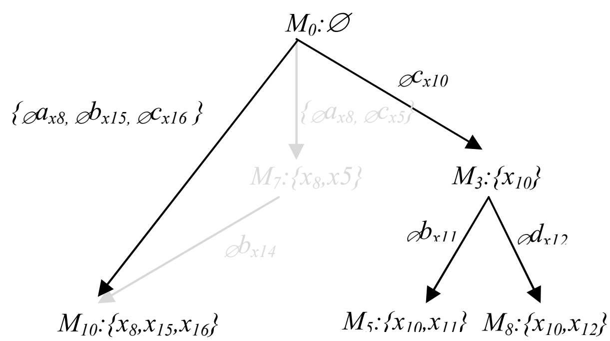


Figure 13. msg after sixth iteration

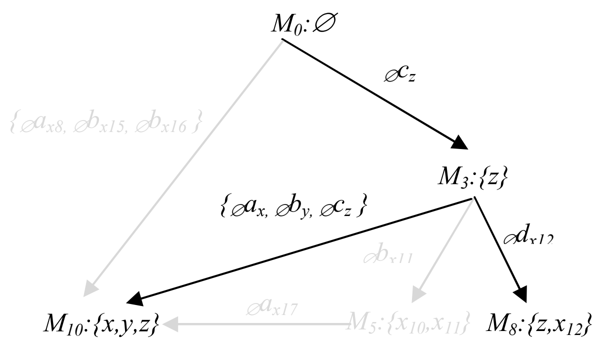


Figure 14. msg after seventh iteration

V. Development and Discussion

A. Development

We have implemented the Algorithm 4.1 as the system of Fig. 15, in which we have two modules:

1. The Graphic-editor module (for Safety-Petri nets and for MSG) is developed with use MDA approach; hence, we propose two meta-model, the first for safety-Petri nets and the second for MSG.
2. The Generator of MSG take as input a safety-Petri net description as a XML file and we give as result a MSG as XML file.

B. Discussion and limitations

In [23], we have developed a tool in which we can build a MSG from LOTOS description and we present also two studied systems with an aim of confirming the fact that it is very difficult to know as a preliminary which is the partial order approach most effective in term of graph built size, this study consists in comparing the ratio of reduction by our technique with the step graphs "CSG", the persistent sets "Pset" and persistent step graphs "PSG". In the present contribution, we have the same conclusion.

We note here as limitations, using MLTS as semantic model, the reduction with the presence of differed conflict is possible and moreover is important through a maximal even concept. But, this concept lowers this technique on time comparing with CSG technique. Since, reducing a MLTS with *n* transitions to MSG we most generate the *n* transitions possible of the MLTS and replace on the fly any sequence of transitions by a step associated, such that represents a minimal path which is determined by an independency relation dynamically calculated from this sequence, which do the necessity of generate all transitions of the MLTS, e.g., to generate the MSG of Fig. 7 we have generated 7 transitions but in the reduced graph we have only 3 steps.

To avoid the generation of all transitions of MLTS with taken the important ratio of reduction, we combine the using of the calculus structural (from Petri net specification) and dynamic of independency relation (from MLTS semantic model).

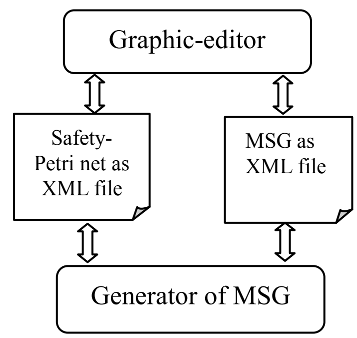


Figure 15. MSG-PetriNets tool

VI. Conclusion

This paper is a contribution to the state space combinatorial explosion problem for Safety-Petri nets. We proposed reduction of MLTS through partial order semantics (by elimination/steps). The MLTS is indeed a model which made possible the consideration of the branches, therefore the reduction is important. The reduced graph is a complete graph preserving the general properties (deadlock states and liveness).

The building of MSG is based on α-equivalent, so we must define the α-equivalent over Petri net in order to generate a MSG for Petri nets. In the other hand, it should be interesting the present contribution in term of specific properties preserving like observational equivalence and failure semantics. It is also interesting to study the equivalence relations over MSGs, and the extension of those to take into account time, like it was already made for MLTSs [21], [22], [25].

References

[1]. Valmari A., « Error detection by reduced reachability graph generation », In Proceedings of Application and theory of Petri Nets. Springer Verlag, LNCS, 1988.

[2]. Valmari A., « Stubborn sets for reduced state space generation », In Proceedings of the Tenth International Conference on Application and Theory of Petri Nets, volume II, Bohn, 1989.

[3]. Valmari A., « A stubborn attack on state explosion », In proceedings of CAV'90, pages 25-42. ACM, DIMACS volume3, 1990.

[4]. GodeFroid P., « Using partial orders to improve automatic verification methods », In proceedings of CAV'90, pages 321-34. ACM, DIMACS volume 3,1990.

[5]. Godefroid P., Wolper P., « A partial approach to model cheking », In proceedings 6th symp. On logic in Computer Science, Volume 531. pages 406-415, Amsterdam, July 1991.

[6]. Godefroid P., Wolper P. « Using partial orders for the efficient verification of deadlock free-dom and safety properties », Formal Methods in Systems Design. 2(2):149-164. April 1993.

[7]. Wolper P., Godefroid P., « Partial-Order methods for temporal verification », In Proceedings of Concur's93. LNCS 575,1993.

[8]. Ribet P. O., Vérification formelle de Systèmes, Contribution à la réduction de l'explosion combinatoire. Thèse de Doctorat, LAAS-CNRS 7 av. du Colonel Roche, 31077 Toulouse Cedex France, 2005.

[9]. West C. H., « Protocol Verification by Random State Exploration », in PSTV VI, pages 233-242, 1986

[10]. Vernadat F., Azéma P., Michel F., « Covering step graph », In Proceedings of Application and Theory of Petri Nets 96. Springer Verlag, LNCS 1091, 1996.

[11]. Magniette F., Pilard L., Rozoy B., " Model-Checking et Produit Synchronisé", in Modélisation des Systèmes Reactifs MSR, pages 213-224, Metz 2003.

[12]. Vernadat F., Michel F., « Covering step graph preserving failure semantics », In Proccedings of Appplication and Theory of Petri nets 97. Springer Verlag, LNCS, 1997.

[13]. Courtiat J. P., Saidouni D. E., « Relating maximality-based semantics to action refinement in process algebras », In D. Hogrefe and S. Leue, editors, IFIP TC/WG6.1, 7th Int. Cof of Formal Description Techniques(FORTE'94) pages 293-308. Chapman &Hall,1995.

[14]. Saidouni D.E., Sémantique de maximalité: Application au raffinement d'actions dans LOTOS. Thèse de Doctorat, LAAS, Université Paul Sabatier Toulouse, Mars 1996

[15]. Belala N., Saidouni D. E., « Non-Atomicity in Timed Models », In Proceedings of ACIT'2005. Amman, Jordan, December 2005.

[16]. Saidouni E. and Belala N., Using Maximality-Based Labeled Transition System Model for Concurrency Logic Verification, The International Arab Journal of Information Technology (IAJIT), vol. 2, no. 3, pp. 199-205,2005.

[17]. Saidouni D. E., Ghenai A., « Intégration des refus temporaries dans les graphes de refus », in proceeding of NOTERE '2006, Toulouse, France, 2006.

[18]. Layeb A. and Saidouni E., Quantum Genetic Algorithm for Binary Decision Diagram Ordering Problem,International Journal of Computer Science and Network Security, vol. 7, no. 9, pp. 130-135, 2007.

[19]. Layeb A. and Saidouni E., A Quantum Genetic Algorithm with Hill Climbing Algorithm for Max 3-SAT Problems, in Proceedings of International Conference on Intelligent Computing (ICIC'2008) China, to Appear in LNCS. Springer-Verlag, 2008.

[20]. Saidouni D. E., Benamira A, Belala N., and Arfi F.,"FOCOVE: Formal Concurrency Verification Environment for Complex Systems". American Institute of Physics AIP Conference proceedings, Vol 1019:375-380, 2008.

[21]. Saıdouni D. E, Belala N., and Bouneb M.: Aggregation of transitions in marking graph generation based on maximality semantics for Petri nets. In Proceedings of the Second International Workshop on Verification and Evaluation of Computer and Communication Systems (VECoS'2008), University of Leeds, UK. eWiC Series, The British Computer Society (BCS), July, 2-3rd 2008. ISSN: 1477-9358

[22]. Saïdouni D. E., Belala N., and Bouneb M., "Maximality-based structural operational semantics for Petri nets", in Proceedings of 2nd Mediterranean Conference on Intelligent Systems and Automation on (CISA'09). Tunisia.

[23]. Benamira A. and Saidouni D. E., «Graphe de pas maximaux : une solution pour la réduction des systèmes de transitions étiquetées maximales » In Proceedings of the Third International Conference on Computer Science and its Applications (CIIA'11) Saida, Algeria, December 13-15, 2011. ceur-ws.org/Vol-825/

[24]. Saidouni D. E., , Matmat R. and Tabib N. «A Distributed Algorithm for MLTS Generation with Aggregation of Transitions» In Proceedings of the Third International Conference on Computer Science and its Applications (CIIA'11) Saida, Algeria, December 13-15, 2011. ceur-ws.org/Vol-825/

[25]. Hachichi H., Kitouni I., Bouaroudj K., and Saidouni D.E., "A graph transformation approach for testing timed systems," accepted by The 18th International Conference on Information and Software Technologies, Kaunas, Lithuania, the proceedings which will be published as a volume of Springer-Verlag CCIS, September,13- 14th 2012, in press.

[26]. Reisig W., Petri Nets: An Introduction, volume 4 of EATCS Monographs in Theoretical Com- puter Science. Springer, May 1985

[27]. Milner R. « Communication and Concurrency », volume 92 of *LNCS*. Springer Verlag, 1980.

[28]. Bolognesi T., Brinksma E., Introduction to the ISO specification language LOTOS, Computer Networks and ISDN Systems, 14:25-59, 1987.

[29]. Mazurckiewicz A., « Trace theory. In Petri Nets: Applications and Relationships to Other Model of Concurrency », Advances in Petri nets 1986, Part II; Proceedings of an advanced Course, pages 279-324. Springer Verlag, LNCS 255,1986.