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 \( i = \{(a,b)\} \).

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].
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 t₁ (resp t₂) represents the beginning of execution (identified by event x (resp y)). 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 t₁; 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.

\[ A \text{. Petri Nets Related Definitions} \]

- A Petri net is a tuple \((S, T, W)\) where \(S\) is the set of places, \(T\) is the set of transitions such that \(S \cup T = \emptyset\), and \(W: (S \times T) \cup (T \times S) \rightarrow \mathbb{N} = \{0, 1, 2, \ldots \}\) 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. \(|S \cup T| \in \mathbb{N}\).

- For \(x \in S \cup T\), the pre-set \(*x\) is defined by \(*x = \{ y \in S \cup T | W(x, y) \neq 0 \}\) and the post-set \(x^*\) is defined by \(x^* = \{ y \in S \cup T | W(x, y) \neq 0 \}\).

- The marking of a Petri net \((S, T, W)\) is defined as a function \(M: S \rightarrow \mathbb{N}\). 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 \(M[t\rightarrow M']\).

- Two transitions \(t_1\) and \(t_2\) (not necessarily distinct) are concurrently enabled by a marking \(M\) iff \(M(s) \geq W(s, t_1) + W(t_2, s)\) 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 \(\emptyset \notin A\) (\(\emptyset\) will indicate invisible action, or silent action).

- The labeling of a Petri net \(N=(S, T, W)\) is a function \(\lambda : T \rightarrow A \cup \{\emptyset\}\). If \(\lambda(t) \in A\) 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 \(\lambda\) is a labeling function of \((S, T, W)\).

- An action \(a \in A\) 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 \(\sigma=M_0; M_1; M_2; \ldots\) is an occurrence sequence iff \(M_0; M_1; M_2; \ldots\) is a transition sequence starting with \(M_0\) there is an occurrence sequence \(M_0; M_1; M_2; \ldots\), if a finite sequence \(t_1; t_2; \ldots; t_n\) leads from \(M\) to \(M'\), we write \(M[t_1; t_2; \ldots; t_n] = M'\). The set of reachable markings of a marked Petri net \((S, T, W, M_0)\) is defined as \([M_0] = \{ M | \exists t_1; t_2; \ldots; t_n : M_0[t_1; t_2; \ldots; t_n] = M \}\).

\[ B \text{. Maximality-based labeled transition systems [13], [14]} \]

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:

- \(\Omega = \langle S, T, \alpha, \beta, s_0 \rangle\) 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.

- \(\alpha\) and \(\beta\) are two applications of \(T\) in \(S\) such that for all transition \(t\) we have: \(\alpha(t)\) is the origin of the transition and \(\beta(t)\) its goal.

- \(S_0\) is the initial state of the transition system \(\Omega\).

2. \((\Omega, \lambda)\) is a transition system labeled by the function \(\lambda\) on an alphabet \(Act\) called support of \((\Omega, \lambda)\). In the other word \(\lambda : T \rightarrow Act\).

3. \(\psi : \sum \rightarrow 2^{\Omega}\) is a function which associates to each state the finite set of maximal event names present in this state.

4. \(\mu : T \rightarrow \emptyset\) 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. \(\xi : T \rightarrow M\) is a function which associates to each transition the event name identifying its occurrence.

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

### 2 \(\alpha\)-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** “\(\alpha\)-equivalent”: Let \(=_{\alpha}\) be the smallest relation over MLTSs such as \(mlts_1 =_{\alpha} mlts_2\) if and only if:

- \(mlts_1 \equiv mlts_2\) (Isomorphism), or

\(mlts_1 \equiv \sum_{i \in I} M_i a_i x_i T_i, mlts_2 \equiv \sum_{j \in J} M_j a_j x_j T_j,\) and

- \(\psi(S) = \psi(T),\) and there is a bijection \(f : I \rightarrow J\) such as, for any \(i \in I\), \(M_i = M_{f(i)} a_i = a_{f(i)}\) and

\(x_i = x_{f(i)}\) and \(T_i = T_{f(i)}\),

\(x_{f(i)} \notin \psi(T_i)\) and \(T_i[x_{f(i)} / x_i] = T_{f(i)}\).

C. Safety-Petri Nets and Maximality Semantics

In [21], we have maximality semantics for Petri nets, the \(\alpha\)-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 Safety-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 \(M\) is the set of all event names identifying bound tokens in the marking \(M\).

Formally, the function \(\psi\) will be used to calculate this set, it can be defined as \(\psi(M) = \bigcup_{i \in S} \bigcup_{y = 1, \ldots, ms} x_i\) such that \(M(s) = (FT, BT)\) with \(BT = \{(t, x)\}\).

2. Let \(N \subseteq M\) be a non-empty finite set of event names, \(\text{makefree}(N, M)\) is defined recursively by:

- \(\text{makefree}\{x_1, x_2, \ldots, x_n\}, M) = \text{makefree}\{x_2, \ldots, x_n\},\)

- \(\text{makefree}\{x\}, M) = M'\) such that for all \(s \in S\), if \(M(s) = (FT, BT)\) then:

\(\text{If there is } (t, x) = BT \text{ then } M'(s) = (FT + 1, 0)\) (Conversion of \(BT\) bound tokens identified by the event name \(x\) to free tokens).

\(\text{Otherwise, } M'(s) = M(s)\).

3. Let \(t\) be a transition of \(T\); \(t\) is said to be enabled by the marking \(M\) iff \(M(s) \geq W(t, s)\) for all \(s \in S\). The set of all transitions enabled by the marking \(M\) will be noted \(\text{enabled}(M)\).
4. The marking $M$ is said to be minimal for the firing of the transition $t$ if $|M(s)|=W(s,t)$ for all $s \in S$.

5. Let $M_1$ and $M_2$ be two markings of the Petri net $(S,T,W)$. $M_1 \in E M_2 \iff \forall s \in S$, if $M_1(s) = (F T_1, B T_1)$ and $M_2(s) = (F T_2, B T_2)$ then $F T_1 \leq F T_2$ and $B T_1 \leq B T_2$.

6. Let $M_1$ and $M_2$ be two markings of the Petri net $(S,T,W)$ such that $M_1 \in E M_2$. The difference $M_2 - M_1$ is a marking $M_3$ such that for all $s \in S$, if $M_1(s) = (F T_1, B T_1)$ and $M_3(s) = (F T_2, B T_2)$ then $M_3(s) = (F T_3, B T_3)$ with $F T_3 = F T_2 - F T_1$ and if $(t,x) \in B T_2$ then $(t,x) \in B T_1$.

7. $\text{Min}(M,t) = \{M|M^{\leq}M\}$ and $M'$ is minimal for the firing of $t$.

8. Let $M$ be a set. The function $\text{get}(M, \{\emptyset\} \rightarrow M$ is a function which satisfies $\text{get}(E) \in E$ for any $E \in 2^M - \{\emptyset\}$.

9. Given a marking $M$, a transition $t$ and an event name $x \notin E(M)$, $\text{occur}(t,x,M) = M'$ such that for all $s \in S$, if $M(s) = (F T, B T)$ then $M'(s) = (F T, B T')$ with $B T' = B T \cup \{W(t), (t,x)\}$ if $W(t,x) \neq \emptyset$ and $B T = B T'$ 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 $\lambda$ associated to $\Sigma$ is a graph in which the states are defined by all reachable markings from the initial marking $M_0$ and the transitions between states are labeled according to the derivation rule of Definition 2.2.

Definition 2.2 Let $M$ be a reachable marking of the marked Petri net $(S, T, W, M_0, \lambda), t \in enabled(M)$ then for all $M^* \in \text{Min}(M,t), E = \psi(M^*)$ and $M^* = \text{makefree}(E, M - M^*)$;

the following derivation is possible: $M \rightarrow M'$ (also denoted by $(M, t, M')$) 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. $x = \text{get}(M, \psi(M^*))$ and
3. $M' = \text{occur}(t, x, M^*)$.

Proposition 2.1 Let $\Sigma = (S, T, W, M_0, \lambda)$ be a labeled system and $Mg$ its marking graph built according to Definition 2.2, 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 $\Sigma$ such that:
   - $Sg$ is the set of states defined by the set of reachable markings from the initial marking $M_0$,
   - $Tg = \{(M, t, M')\}$ such that $M, M' \in Sg$ and $(M, t, M')$ is a valid derivation.
   - $\alpha$ is the relation defined on $Tg$ such that $\alpha((M, t, M')) = M$ and $\beta((M, t, M')) = M'$.
5. $\psi : Sg \rightarrow 2^M$ is defined as of MLTS.
6. For $d = (M, t, M') \in Tg$ 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 interleaved 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 Mazurkiewicz’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):

Events sequence $:\_\_ >$ is a function inductively defined by:

- $\_\_ > = \text{def} \varepsilon$
- $\_\_ a, p > = \text{def} x \cdot p >$

Support of a transitions sequence $: || |$ is a function is defined as follows:

- $|| c > = \text{def} c$
- $|| u, w || = \text{def} \{u \} \sqcup || w ||$

Extension of Mazurkiewicz’s trace to MLTS:

Let $mlts = \langle S, s_0, T, \psi, \mu, \xi >$ be a MLTS. $U_{\lambda}a_{\lambda}b, V$ and $U_{\lambda}b_{\lambda}a, V$ are two paths of $mlts$. Let $\approx$ be the relation defined on $T^* \times T^*$ by $< U_{\lambda}a_{\lambda}b, V > \approx < U_{\lambda}b_{\lambda}a, V >$ 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 $w$.

Maximal path: Let $mlts = \langle S, s_0, T, \psi, \mu, \xi >$ be a MLTS and $w \in T^*$. $w$ is a maximal path

- $\exists s, s' \in S, s \Rightarrow s' : || w || \subseteq \psi(s')$ and $s' \rightarrow$
- $\forall (\exists t \in T : w$ is not a maximal path)

Minimal path: Let $C_i$ be a maximal paths set associated to the state $s$. $\text{Min}(C_i) = \{c \mid \forall e \in C_i, \langle c \rightarrow e < c \rangle < \langle c > \}$.

Maximal paths equivalence: Two maximal paths $w$ and $w'$ are equivalent, noted $w \approx w'$, if and only if:

$s \Rightarrow s'$ implies that $s \Rightarrow s'$. It is particular case of the relation of Mazurkiewicz’s trace equivalence in which all events are independent.
8. Maximal step: Let \( \text{mlts} = (S, s_0, T, \psi, \mu, \xi) \), and \( w \in T^* \), \( ||w|| \) defines a step if and only if:
\[
\exists s, s' \in S, w \in T^*, s \xrightarrow{w} s': \forall e \in <w> , e \in \psi(s') .
\]

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 \( s \xrightarrow{w} s' \).

The associated step is \( \xrightarrow{w} \).

10. Maximality-based Step Graph: Let \( \text{mlts} = (\Omega, \lambda, \mu, \xi, \psi) \) such that \( \Omega = (S, T, a, \beta, s_0) \) be MLTS,
\( \text{msg} = (\Omega, \lambda', \mu', \xi', \psi) \) such that \( \Omega' = (S', \Xi, a, \beta, s_0) \) is a MSG of \( \text{mlts} \) if and only if:
\[
\begin{align*}
1. & \forall s' \in S': s' \in S, \\
2. & \forall t' \in \Xi : t' \text{ is a step, where } ||t'|| \text{ constitute a maximal path in } \text{mlts}, \\
3. & \forall s \in S', s \xrightarrow{\mu_a} s' \in T ,
\end{align*}
\]

\[\forall s'' \in S', \forall w \in T^*, s' \xrightarrow{w} s'' \]
\[\Rightarrow \{ \exists w' \in \Xi, s \xrightarrow{p} s' : [<M, a, w>] = [<w'>] \}\]

Such that:
- \( \zeta' : 2^T \rightarrow 2^\mathbb{M} \):
  \[
  \begin{align*}
  \zeta'(t) & = \text{def } t, \\
  \zeta'(t \cup E) & = \text{def } \zeta'(t) \cup \zeta'(E).
  \end{align*}
  \]
- \( \mu' : 2^T \rightarrow 2^\mathbb{M} \):
  \[
  \begin{align*}
  \mu'(t) & = \text{def } t, \\
  \mu'((t \cup E) & = \text{def } \mu(t) \cup \mu'(E).
  \end{align*}
  \]

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 \( \zeta'(E) \subseteq \psi(s) - \mu'(E) \cup \mu'(E) \subseteq \psi(s') \).

**Proposition 3.1:** Let \( s \xrightarrow{w} s_1 \) and \( s \xrightarrow{w'} s_2 \), if \( S_1 \) and \( S_2 \) are \( \alpha \)-equivalents then \( w \equiv w' \).

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

**Proposition 3.3:** Let \( s \xrightarrow{w} s_1 \) and \( s \xrightarrow{w'} s_2 \) such that \( S_1 \) and \( S_2 \) are \( \alpha \)-equivalents, if \( u \in \text{Min}(C) \) such that \( w = u \), 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 Elimation of the superfluous interleaving, which released when we detect two states \( \alpha \)-equivalent (see Definition 4.1), so we have diamond in which all branches are Mazurkiewicz’s trace equivalent (see Proposition3.1). By Proposition3.3 we can eliminate all the superfluous interleaving and take only the branch \( w = u \), the branch \( w \) preserves deadlock states and liveness property.

In the algorithm, we represent the state \( s \) by the marking \( M_s \).

**Proposition 4.1:** Let \( \text{msg} \) be a MSG in generation in state \( s' \), and let \( s \xrightarrow{p} s' \) step of \( \text{msg} \): for any transition generated from this state \( s' \xrightarrow{t} s'' \), we have:
Either $pt$ is step, we replace $s\xrightarrow{pt}p$ by $s\xrightarrow{i}p$.

Definition 4.1: Let $\Sigma=(S,T,W, M_0, \lambda)$ be a labeled system. The $\alpha$-equivalence relation is recursively defined over configurations as follows:
- $M(s)=M'(s)$ iff $FT(s)=FT'(s)$.
- $(t,s)\in BT(s)$ such that $(t,s)\in BT(s)$.
- $M_a M'$ iff $\forall s\in S$, $M(s)=M'(s)$.

Algorithm 4.1 "basic on the fly maximal step graph generation "

Require: $R$ be a safety-Petri net ;
Ensure: $msg=(\Omega,\lambda,\xi,\psi)$ such that $\Omega=S, \Sigma, \alpha, \beta, s_0$ ;
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 $T'=\text{enabled}(s)=\cup\{s\rightarrow s_j\}; X=\cup\{s_j\}$
5 For each $s'' \xrightarrow{i} p$ Do
6 For each $t_j$ de $T$ Do Build step w.r.t Proposition 4.1 ;
7 For each $s_i$ state of $X$ $\alpha$-equivalent with $s''$ of $S$ Do implement the Proposition 3.3
8 Insert all new states $s_i$ of $X$ modulo $\alpha$-equivalent in $S'$ ;
9 Endwhile
Fin.Algo.

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

![Figure 7. Differed conflict.](image)

1. By initialization $S'=\{M_0\}$, we take from line 3 $S=\{M_0\}$. By Definition 2.2, we have Fig. 8, with $S'=X=\{M_1,M_2,M_3\}$ and any states $\alpha$-equivalent. With
- $M_0=[(1,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),0,0,(a,x_1)]$.
- $M_1=[(1,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),0,0),(1, (a,x_1))].$
- $M_2=[(0,\emptyset),(0,\emptyset),(1, \emptyset),(0,\emptyset),(b,x_2),(0,0),(0,0)].$
- $M_3=[(0,\emptyset),(1, \emptyset),(0, \emptyset),(0,\emptyset),(0,\emptyset),(0,0)].$

We have any modification by the application of Proposition 4.1.

![Figure 8. msg after first iteration](image)

2. In the second iteration, we select and remove $M_1$ from $S'$ to $S$, so $S'=\{M_0,M_1\}$. By Definition 2.2, we have fig.9(a), with $X=\{M_4,M_5\}$ and $S'=\{M_6,M_7,M_8,M_9\}$. Any states $\alpha$-equivalent. But, we build two steps by the use of Proposition 4.1, so we remove definitively $M_1$ from the graph. We obtain as result the Fig.9(b). With
- $M_4=[(1,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),(b,x_3)].$
- $M_5=[(0,\emptyset),(0,\emptyset),(1,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),(a,x_1)]].$

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 Proposition 4.1 we remove definitively $M_2$ and we build two steps. But in this iteration, we have $M_4=M_5$, so, we remove definitively $M_6$ (or $M_4$) and we substitute the graph by $\sigma'=x_7/x_8,x_1/x_2,x_3/x_4/x_5$. We obtain as result the Fig.10 with $S=\{M_0\}$ and $S'=\{M_6,M_7,M_8,M_9\}$. With
- $M_6=[(1,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),(a,x_1)].$
- $M_7=[(0,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),(a,x_1)].$

![Figure 10. Refined conflict.](image)
6. We select and remove $M_7$ from $S'$. By Definition 2.2, we have $X=\{M_6, M_9\}$. We remove definitively $M_7$ and building a step as Fig.13. In this iteration, we have $M_{10} = M_{11}$ and we substitute the graph by $\sigma = x_9/x_{10}, x_{16}, x_{14}, x_{15}, x_{16}$. So, we obtain as result the Fig.13 with $S=\{M_0, M_3\}$ and $S' = \{M_4, M_9, M_{10}\}$.

7. We select and remove $M_8$ from $S'$. By Definition 2.2, we have $X=\{M_{12}\}$ and from Proposition 4.1 we remove definitively $M_8$ and building a step as Fig.14. We have $M_{10} = M_{12}$ and we substitute the graph by $\sigma = x_{13}/y, x_{13}, y_{17}/x_{15}/x_{10}, x_{10}/z, x_{10}/z$. So, we obtain as result the Fig. 14 with $S=\{M_0, M_3\}$ and $S' = \{M_4, M_8, M_{10}\}$.
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).

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 \( \alpha \)-equivalent, so we must define the \( \alpha \)-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


[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.


