CONSIDERATION OF THE COVERING STEPS IN THE MAXIMALITY-BASED LABELED TRANSITIONS SYSTEMS

Adel Benamira and Djamel-Eddine Saidouni

LIRE Laboratory, University of Mentouri, 25000 Constantine, Algeria.
a.benamira@yahoo.fr , saidounid@hotmail.com

ABSTRACT

Partial order approaches seek to solve the state space combinatorial explosion problem by tackling one of its causes namely the parallelism representation by interleaving execution of actions. This paper proposes the joint use of the covering steps and the maximality semantics, as a partial order approach for the resolution of this problem..

Keywords: *Formal verification, Partial order semantic, Maximality-based semantic, Covering step graph.*

1. INTRODUCTION

This work enters in the frame of 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. 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][5][6][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] (Figure 1.(*b*)). Another alternative was proposed in [9][10][11]. This approach consists of regrouping independent events in only one step (Figure 1.(*c*)). 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].

Two principal limitations may be distinguished in partial order approaches. The first one concerns the generic factor of the approach. In fact, the calculated independency relation is structural. Consequently, several kinds of independent transitions may not be taken into account, for example the case of the structural conflict [13] (Figure 2). In other words, dynamic parallelism is not considered in this approach.

In [14] an attempt was proposed for the consideration of dynamic conflict relation. However, this solution is specific to Petri nets.

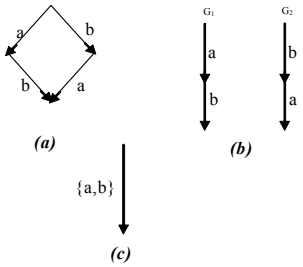

Figure 1: Transition Systems of the behavior expression $a|||b$

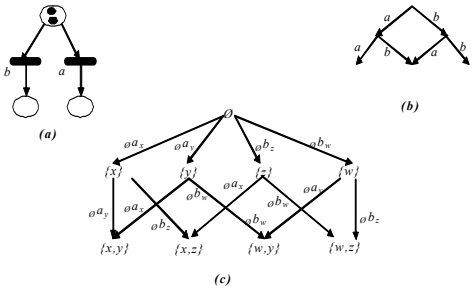

Figure 2: Structural conflit

The second limitation is summarized in the powerlessness to exploit all the independency relations. For example, 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 (Figure 3) where its strong presence decreases the reduction ratio. In deed, branches are not considered any more in the possible reductions.

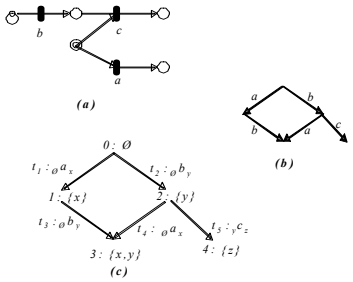

Figure 3: Differed conflit

In this paper, we propose a reduction method which combines the use of the MLTS model (Maximality-based Labeled Transitions Systems) [15] and the covering steps method that will make it possible to answer the limits quoted above. Note that the MLTS

model has been used in work relating to the specification and the verification of reactive systems [16][17][18].

The MLTS model can be used as a semantic representation of systems behaviors. Hence, various specification models may be used (RdP [13], CCS [19], LOTOS [20] …); for that, it is enough to define semantics in MLTS term for each one. Let us take for example the MLTS of Figure 3.(*c*) representing the behavior of Figure 3.(*a*). In the initial state, no action is in execution. Transition t1 (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.

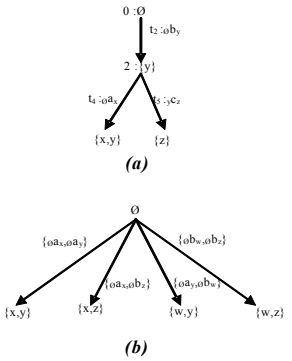

Figure 4: Recuced graphs

Maximal events concept allows to take into account dynamic parallelism, which allows the exploitation of full independency relations between actions during the reduction operation. For example, Figure 3.(*c*) may be reduced, as a result we obtain the MLTS represented by Figure 4.(*a*). Also, the reduced graph of Figure 2.(*c*) is represented by Figure 4.(*b*).

The paper is organized as follows. Section 2 recalls some definitions related to the maximality-based semantics of Basic Lotos as presented in [21][15]. In section 3, α-equivalence relation is presented and an on the fly reduction algorithm is given. Section 4 introduces our reduction method based on the joint use of the maximality-based semantics and the covering steps approach. In Section 5, we discuss the obtained results.

2. MAXIMALITY SEMANTICS

2.1. THE INTUITION OF THE MAXIMALITY SEMANTICS [21][15]

The semantics of a concurrent system can be characterized by the set of the system states and the transitions by which the system passes from a state to another. In the approach based on the maximality, the transitions are events which represent only the beginning of actions execution. To distinguish each action execution, an identifier is associated to its beginning. In a state, an event is said maximal if it corresponds to the beginning of the execution of an action which can possibly be always being carried out in this state.

To illustrate the maximality principle, let us consider the lotos behavior expressions *E≡a;stop|||b;stop* and *F≡a;b;stop[]b;a;stop*. In the initial state, no action was started its execution, therefore the set of maximal events is empty, from where following initial configurations associated to *E* and *F* are $_{\emptyset}[E]$ and $_{\emptyset}[F]$. From the configuration $_{\emptyset}[E]$, starting the execution of actions a and b leads to the following transitions:

$$_{\phi}[E] \xrightarrow{\phi\, a_x} {}_{m\,\{x\}}[stop] \,|||\, {}_{\phi}[b;stop] \xrightarrow{\phi\, b_y} {}_{m\,\{x\}}[stop] \,|||\, {}_{\{y\}}[stop]$$

x (respectively y) being the event name identifying the beginning of the action *a* (respectively *b*). Since nothing can be concluded about the termination of both actions *a* and *b* in the configuration $_{\{x\}}[stop] \,|||\, {}_{\{y\}}[stop]$, *x* and *y* are then maximal in this configuration. Let us note that *x* is also maximal in the intermediate state represented by the configuration $_{\{x\}}[stop] \,|||\, {}_{\emptyset}[b;stop]$.

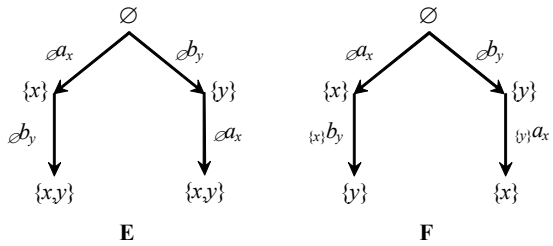

Figure 5: MLTS structures of *E* and *F*

For the initial configuration, associated to the behavior expression *F*, the following transition is possible: $_{\phi}[F] \xrightarrow{\phi\, a_x} {}_{m\,\{x\}}[b;stop]$. As previously, *x* identifies the beginning of the action a and it is the unique maximal event name in the configuration $_{\{x\}}[b;stop]$. It is clear that, within sight of the action prefixing operator semantics, the beginning of the action b is possible only if the action a terminates its execution. Consequently, *x* does not remain maximal any more when the action *b* begins its execution; the unique maximal event in the resulting configuration is *y* which identifies the beginning of execution of action *b*. Thus the following derivation $_{\{x\}}[b;stop] \xrightarrow{\{x\}\, b_y} {}_{m\,\{x\}}[stop]$.

The configuration $_{\{y\}}[stop]$ is different from the configuration $_{\{x\}}[stop] \,|||\, {}_{\{y\}}[stop]$, because the first has only one maximal event (identified by *y*), whereas the second has two (identified by *x* and *y*). The derivation structures of the behavior expressions E and F obtained by the application of the maximality semantics principle are represented in Figure 5. These structures are called Maximality-based Labeled Transition System (MLTS).

2.2. RELATED DEFINITIONS

In this section, we recall some related definitions of Basic lotos. The complete presentation of Basic lotos

maximality-based operational semantics may be fond in [21][15].

- *Syntax of Basic LOTOS*: Let *PN* be the set of processes ranged over by P and let $\mathcal{G}$ be the set of gates ranged over by g. A particular observable action $\delta \notin \mathcal{G}$ is used to notify the successful termination of the processes. L indicates any subset of $\mathcal{G}$, the internal action is noted by i. The set of all actions is indicated by $Act = \mathcal{G} \cup \{i,\delta\}$. $\mathcal{B}$, ranged over by *E, F, ...* denotes the set of behavior expressions whose syntax is:

  $E ::= Stop \mid exit \mid E[L] \mid g;E \mid i;E \mid E[\,]E$
  $E|[L]|E \mid hide\ L\ in\ E \mid E>>E \mid E[>E$

  Given a process *P* which have the behavior *E*, the definition of *P* is expressed by *P:=E*.
- The set of event names is a countable set indicated by *M*. This set is ranged over by *x,y,....* M,N,... indicate finite subsets of *M*. The set of atoms of support *Act* is $Atm = 2^{M}_{fn} \times Act \times M$, 2^{M}_{fn} being the set of finite parts of *M*. For $M \in 2^{M}_{fn}$, $x \in M$ and $a \in Act$, the atom (M,*a*,*x*) will be noted ${}_{M}a_{x}$. The choice of an event name can be done in a deterministic way by the use of any function $get: 2^{M}-\{\emptyset\} \rightarrow M$ satisfying $get(M) \in M$ for any $M \in 2^{M}-\{\emptyset\}$.
- Configuration : The set $\mathcal{C}$ of configurations of Basic LOTOS behavior expressions is the smallest set defined inductively as follows:
  - $\forall E \in \mathcal{B}, \forall M \in 2^{M}_{fn} : {}_{M}[E] \in \mathcal{C}$
  - $\forall P \in PN, \forall M \in 2^{M}_{fn} : {}_{M}[P] \in \mathcal{C}$
  - If $\mathcal{E} \in \mathcal{C}$ then $hide\ L\ in\ \mathcal{E} \in \mathcal{C}$
  - If $\mathcal{E} \in \mathcal{C}$ and $F \in \mathcal{B}$ then $\mathcal{E}>>F \in \mathcal{C}$
  - If $\mathcal{E}, \mathcal{F} \in \mathcal{C}$ then $\mathcal{E}$ op $\mathcal{F} \in \mathcal{C}$
    op $\in \{[\,], |||, \|, |[\,]|, [>\}$
  - If $\mathcal{E} \in \mathcal{C}$ and $\{a_1,a_2, \ldots, a_n\}, \{b_1,b_2, \ldots, b_n\} \in 2^{M}_{fn}$ then $\mathcal{E}[a_1/b_1,a_2/b_2,\ldots,a_n/b_n]$

  Given a set $M \in 2^{M}_{fn}$, ${}_{M}[\ldots]$ is called embedding operation. This operation is distributive over the operations [],|[L]|, hide, [> and the renaming gates operation. We also admit that ${}_{M}[E>>F] \equiv {}_{M}[E]>>F$. A configuration is known as canonical if it cannot be reduced any more by the distribution of the embedding operation on the other operators. Thereafter, we suppose that all configurations are in canonical form.

  Any canonical configuration is under one of the following forms ($\mathcal{E}$ and $\mathcal{F}$ being canonical Configurations):

  ${}_{M}[stop] \mid {}_{M}[exit] \mid {}_{M}[a;E] \mid {}_{M}[P] \mid \mathcal{E}[\,]\mathcal{F} \mid$
  $hide\ L\ in\ \mathcal{E} \mid \mathcal{E}>>F \mid \mathcal{E}[>\mathcal{F} \mid \mathcal{E}[a_1/b_1,a_2/b_2,\ldots,a_n/b_n]$
- The function $\psi : \mathcal{C} \rightarrow 2^{M}_{fn}$, which determines the set of event names in a configuration, is defined inductively by:

  $\psi({}_{M}[E]) = M$
  $\psi(\mathcal{E}\,[\,]\mathcal{F}) = \psi(\mathcal{E}) \cup \psi(\mathcal{F})$
  $\psi(\mathcal{E}\,|[L]|\,\mathcal{F}) = \psi(\mathcal{E}) \cup \psi(\mathcal{F})$
  $\psi(\mathcal{E} >> F) = \psi(\mathcal{E})$
  $\psi(hide\ L\ in\ \mathcal{E}) = \psi(\mathcal{E})$
  $\psi(\mathcal{E}\,[>\mathcal{F}) = \psi(\mathcal{E}) \cup \psi(\mathcal{F})$
  $\psi(\mathcal{E}\,[b_1/a_1,\ldots,b_n/a_n]) = \psi(\mathcal{E})$
- Let *E* be a configuration; *E* \N indicates the configuration obtained by removing the set of event names N from the configuration *E*. *E* \N is defined inductively as follows:

  ${}_{M}[E])\backslash N = {}_{M-N}[E]$
  $(\mathcal{E}\,[\,]\mathcal{F})\backslash N = \mathcal{E}\backslash N\,[\,]\,\mathcal{F}\backslash N$
  $(\mathcal{E}\,|[L]|\,\mathcal{F})\backslash N = \mathcal{E}\backslash N\,|[L]|\,\mathcal{F}\backslash N$
  $(hide\ L\ in\ \mathcal{E})\backslash N = hide\ L\ in\ \mathcal{E}\backslash N$
  $(\mathcal{E}>>\mathcal{F})\backslash N = \mathcal{E}\backslash N >> \mathcal{F}$
  $(\mathcal{E}[>\mathcal{F})\backslash N = \mathcal{E}\backslash N\,[>\mathcal{F}\backslash N$
  $(\mathcal{E}[b_1/a_1,\ldots,b_n/a_n])\backslash N = \mathcal{E}\backslash N\,[b_1/a_1,\ldots,b_n/a_n]$
- The set of substitution functions of event names is noted *Subs* (i.e. $Subs = M \rightarrow 2^{M}_{fn}$); $\sigma, \sigma_1, \sigma_2, \ldots$ are elements of Subs. Given $x,y,z \in M$ and $M \in 2^{M}_{fn}$, then
  - The application of σ to *x* will be written by σx
  - The substitution identity function ι is defined by $\iota x = \{x\}$
  - $M\sigma = \cup_{x \in M} \sigma x$;
  - $\sigma[y/z]$ is defined by

    $$\sigma[y/z]x = \begin{cases} \{y\} & \text{if } z=x \\ \sigma x & \text{otherwise} \end{cases}$$

  Let σ be a substitution function, the simultaneous substitution of all occurrences of *x* in $\mathcal{E}$ by σx, is defined recursively on the configuration $\mathcal{E}$ as follows:

  ${}_{M}[\mathcal{E}])\sigma = {}_{M\sigma}[\mathcal{E}]$
  $(\mathcal{E}[\,]\mathcal{F})\sigma = \mathcal{E}\sigma\,[\,]\mathcal{F}\sigma$
  $(\mathcal{E}|[L]|\,\mathcal{F})\sigma = \mathcal{E}\sigma\,|[L]|\,\mathcal{F}\sigma$
  $(hide\ L\ in\ \mathcal{E})\sigma = hide\ L\ in\ \mathcal{E}\sigma$
  $(\mathcal{E}>>\mathcal{F})\sigma = \mathcal{E}\sigma >> F$
  $(\mathcal{E}[>\mathcal{F})\sigma = \mathcal{E}\sigma\,[>\mathcal{F}\sigma$
  $(\mathcal{E}[b_1/a_1,\ldots,b_n/a_n])\sigma = \mathcal{E}\sigma[b_1/a_1,\ldots,b_n/a_n]$

3. α-EQUIVALENT RELATION [15]

The purpose of this relation it to put in correspondance MLTSs describing the same behavior of which the only difference resides in the choice of event names.

For example, both MLTSs of Figure 6 describe the same behavior (the parallel execution of actions *a* and *b*), we can obtain the MLTS of Figure 6.(*a*) from that of Figure 6.(*b*) by substituting event names *e* by *x* and event name *z* by *y*.

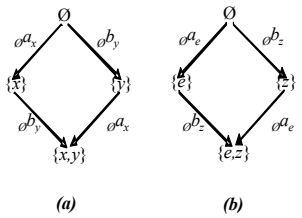

Figure 6: Two MTLSs α-equivalent

Definition 1 α-reduction

Let $=_\alpha$ be the smallest relation over MLTS such as $S=_\alpha T$ iff

- $S \cong T$,or
- $S \cong \sum_{i\in I} {}_{M_i} a_{ix_i} T_i,\ T \cong \sum_{j\in J} {}_{M_j} a_{jx_j} T_j$ and
- $\Psi(S) = \Psi(S)$, and there is a bijection $f{:}I\rightarrow J$ such as, for any $i \in I$, $M=M_{f(i)}, a^i=a^{f(i)}$, and
  - ✓ $x_i=x_{f(i)}$ and $T_i=_\alpha T_{f(i)}$
  - ✓ $x_{f(i)} \notin \Psi(T_i)$ and $T_i[x_{f(i)}/x_i]=_\alpha T_{f(i)}$

3.1. REDUCTION MODULO THE α-EQUIVALENCE RELATION

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 Figure 7.(*a*) represents the behavior of the Lotos expression *a;d;stop|[d]|b;d;stop*, it was generated by the direct application of the lotos maximality-based operational semantics [15]. Both sub-MLTSs S_1 and S_2 of Figure 7.(*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 $S_1\sigma_1 \equiv S_2\sigma_2$. To remove such a redundancy, we must, initially, apply the substitution function $\sigma_1 \cup \sigma_2$ to the MLTS of Figure 7.(*a*), group the start stats of $S_1$and S_2, and then, we remove $S_1\sigma_1$ or $S_2\sigma_2$. As a result we obtain the MLTS of Figure 7.(*b*)

3.2. ON THE FLY α-REDUCED MLTS GENERATION ALGORITHM

What we come to see is applied on an already generated MLTS; however, our goal is the on the fly generation of the α-reduced MLTS. The alternative consists to verify for each generated configuration if it is α-equivalent with previously generated configuration. In this case, a substitution function is applied to the MLTS, and a configuration is removed.

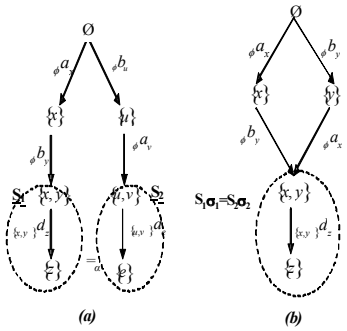

Figure 7: α-reduction

Definition 2

α-equivalence is recursively defined over the configurations as follows:

${}_M[E]=\alpha\ {}_N[E]$, if there exists σ where Mσ=Nσ

$E\ |[L]|\ F\ =\alpha\ E'|[L]|\ F'$, iff $E =\alpha\ E'$ and $F =\alpha\ F'$

E op $F =\alpha\ E'$ op F', iff $E=\alpha\ E'$ and $F =\alpha\ F'$,

op={';','[],'[>','>>'}

hide L in $E =\alpha$ hide L in E', iff $E =\alpha\ E'$

$M[P]=\alpha\ N[P]$, if there exists σ1,σ2 where Mσ1=Nσ2

To simplify the construction of the substitution function, we have associated, for each maximal event, its suitable action. In the example of Figure 8, configurations 1 and 2 are α-equivalent, the events (a,1) and (*a*,4) (resp (*b*,3)and (*b*,2)) represent the same event, for instance (*a*,5) (resp (*b*,6)). We can consider the substitution function σ={(*a*,5)/(*a*,1), (*a*,5)/(*a*,4),(*b*,6)/(*b*,2), (*b*,6)/(*b*,3)}.

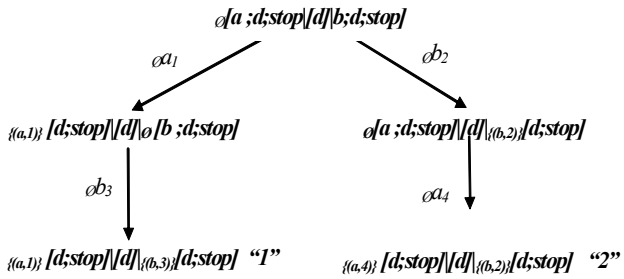

Figure 8 : MLTS in generation phase

Algorithm 3 allows us, starting from a behavior expression, to generate on the fly an α-reduced MLTS. The construction is based on the configurations where for each new configuration, we extract new configurations and new transitions, by using the rules of the maximality-based operational semantics [21][15].

A configuration is known as new if it is not α-equivalent with any other previously generated configuration.

Algorithm 3 "On the fly MLTS generation algorithm"
Data: LOTOS behavior expression;
Results: an MLTS α-reduced;
Var:
Confs_List: list of untreated configurations;
Confs_treated_List: list of already treated configurations;
Sub: list of couple list (action,event) representing a same event (Function σ).
Begin
build initial configuration;
initialize the list of *Confs_List* configurations by the initial configuration;
While *Confs_list* Non empty **Do**
Sub←Ø
select and remove an element Conf de *Confs_List*;
Treat Conf configuration;
add Conf to the *Confs_treated_List* list;
add the new resulting configurations to *Confs_List*;
add the resulting transitions to the MLTS;
Substitute MLTS, *Confs_List* and *Confs_treated_List* by using Sub;
EndWhile
End.
The description of this algorithm is given in [22].

4. REDUCTION BASED ON THE MAXIMALITY SEMANTICS

Inspiring from the covering steps technique, we do not consider all possible interleaving. On the other hand, we build, under certain conditions, a step allowing directly reaching the final state which would have been reached by each interlaced sequence. Figure 9 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 Figure 9.(*b*) is the step graph of the MLTS of Figure 9.(*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 [23]. It will proved that our approch preserves deadlock states and liveness property. Its on the fly generation is thus possible.

4.1. PRELIMINARY DEFINITIONS

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

- *Events sequence* : Let *Atm* be an set of atoms and *M* a set of event names , <_> is the mapping *Atm* →*M*, inductively defined by:
  - ✓ $< \epsilon >=_{def} \epsilon$
  - ✓ $<_M a_x.p>=_{def} x.<p>$
- *Support of a transitions sequence :* || || is *a* mapping $T \rightarrow \rho(T)$ defined by*:*
  - ✓ $|| \epsilon ||=_{def} \emptyset$
  - ✓ $||u.w||=_{def}\{u\} \cup ||w||$

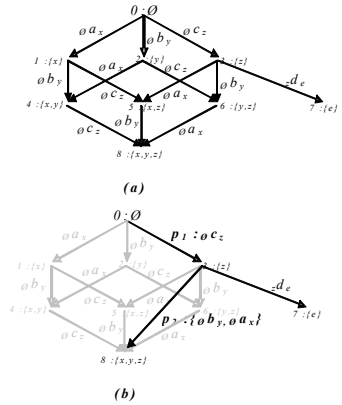

Figure 9 : A MLTS and its maximum steps graph

- Extension of Mazurckiewicz's trace to MLTS : Let $G=<S,s_0,T,\psi,\mu,\xi>$ be a MLTS. $U._M a_{x\cdot N} b_y.V$ and $U._N b_{y\cdot M} a_x.V$ are two paths of *G*. Let ≈ be the relation defined on $T^* \times T^*$ by $< U._M a_{x\cdot N} b_y.V >\approx< U._N b_{y\cdot M} a_x.V >$ if $x \notin N$ and $y \notin M'$, by construction, ≈ is reflexive and symmetric. The trace equivalence ≡ can be defined by the transitive closing of the relation ≈, Equivalence classes of ≡ are called traces. [<*w*>] the trace generated by *w*.
- *Maximal path* : Let $G=<S,s_0,T,\psi,\mu,\xi>$ be a MLTS and $w \in T^*$, *w* is a maximal path $\exists s, s' \in S, s \overset{w}{\Rightarrow} s'$:
  - ✓ $\|<w>\| \in \psi(s')$
  - ✓ $(s \nrightarrow) \vee (\exists t \in T) : wt$ is not a maximal path)
- *Minimal path* : Let C_s be a maximal paths set associated to the state *s*.

$$Min(C_s)=\{c : / \nexists c' \in C_s : \|< c' >\| \subset \|< c >\|\}$$

- *Maximal paths equivalence* : Two maximal paths *w* and *w'* are equivalent, noted $w \approx_c w'$, if $s \overset{w}{\Rightarrow} s'$ implies that $s \overset{w'}{\Rightarrow} s'$. It is a particular case of the relation of Mazurckiewicz's trace equivalence in which all events are independent.
- *Maximal step* : Let $T=<S,s_0,T,\psi,\mu,\xi>$, and $w \in T^*$, $\|w\|$ defines a step iff $\exists s, s' \in S, w \in T^*, s \overset{w}{\Rightarrow} s'$ such as $\forall e \in \|< w >\|, e \in \psi(s')$

Property 2 :
Let w and w′ be two maximal paths:
1. if $w \approx_c w'$ then [<*w*>]≡[<*w'*>].
2. A maximal path is a finite path.
3. The transitions of a maximal path constitute a step of transitions.

- Extension of the accessibility relation to the maximal transitions steps : Let $\rightarrow_p$ be an extension of → to the maximal steps, and *w* be

a maximal path $s \overset{w}{\Rightarrow} s'$. The associated step is $\overset{\|w\|}{\rightarrow}_p$.

For example, for the initial state of Figure 3.(*c*) the possible maximal paths are

$C_{\emptyset}=[\emptyset b_y; \emptyset b_y.\emptyset a_x; \emptyset b_y.\emptyset a_x]$, where $\emptyset b_y.\emptyset a_x$ and $\emptyset b_y.\emptyset a_x$ are two equivalent paths. $\emptyset b_y$ is the small path of $C_{\emptyset}$.

The suggested reduction method consists to replace all equivalent maximal paths by only one path. This path should be replaced by a maximal step. At the end, the built graph is a maximal steps graph.

4.2. MAXIMAL PATHS PRESERVING

Maximal paths preserving consists, for each state, to take into account only its transitions which cover all its maximal paths traces. We note the MLTS which preserves the maximal paths by MLTS^{op} (Definition 4 illustration, in Figure 3.(*c*), transition t_1 outgoing from state 0 preserves the trace [*x.y*], whereas the other transition t_2 preserves the traces [*x.y*] and [*y*]. Consequently, the second transition should be considered. Figure 4.(*a*) represents the MLTS^{op} corresponding to the MLTS of Figure 3.(*c*).

Definition 4 "MLTS ^{op}"

Let T=<$S,s_{\emptyset},T,\psi,\mu,\xi$> and T'=<$S,s_{\emptyset},T',\psi',\mu',\xi'$> be two MTLSs. T' is the MLTS^{op} of T iff:

1. $\forall s' \in S' : s' \in S$,
2. $\forall t' \in T' : t' \in T$ and
3. $\exists s \in S', s \xrightarrow{{}_M a_x} s' \in T, \forall s'' \in S', \forall w \in T^* : s' \overset{w}{\Rightarrow} s''$
$\Rightarrow \left\{ \exists w' \in T'^{*5}, s \overset{w'}{\Rightarrow} s'' : [<_M a_x.w>] = [<w'>] \right\}$

Condition 1 imposes that any state of T' is a state of T. Condition 2 imposes that any transition of T' is a transition of T, and Condition 3 expresses a condition of cover via Mazurkiewcz's traces between the sequences of the MLTS and those associated T'. Consequently, T' is a complete graph of the initial T which preserves deadlock states and liveness property.

Proposition 5

Let T be an MLTS and T' be the MLTS^{op} of T, then

1. T' preserves the maximal paths of T.
2. DeadLock (T) =DeadLock (T').
3. for any event *e* of T′, *e* is accessible from any state of T′(known as alive) iff *e* is accessible from any state of T.

Proof. See [24]

As an illustration, The behavior expression *(a;stop[]b;stop)|||(c;stop[]d;stop)* presents two independent conflicts sets. Figure 10.(*a*) introduces its associated α-reduced MLTS generated by the LotoSTEM tool [22].

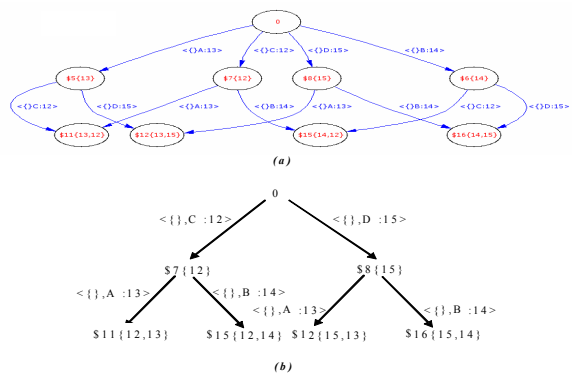

Figure 10 : A MLTS and its MLTS^{op}

Figure 10.(*b*) represents its corresponding MLTS^{op}. For example, paths $\emptyset A_{13}.\emptyset C_{12}$ and $\emptyset C_{12}.\emptyset A_{13}$ of Figure 10.(*a*) are covered in Figure 10.(*b*) by the path $\emptyset C_{12}.\emptyset A_{13}$. Remark that deadlock states $11, $12,$15 and $16 are preserved.

4.3. MAXIMAL STEPS GRAPH

Intuitively, MLTS^{op} is a Maximal Steps Graph (MSG) in which each maximal paths are replaced by their corresponding step. As an illustration, the MSG of Figure 11 was obtained by replacing the maximal paths set{$\emptyset C_{12}.\emptyset A_{13};\emptyset C_{12}.\emptyset B_{14};\emptyset D_{15}.\emptyset A_{13};\emptyset D_{15}.\emptyset B_{14}$} of Figure 10.(*b*) by their corresponding step set.

Definition 6

Let T=<$S,s_{\emptyset},T,\psi,\mu,\xi$>be an MLTS$_{op}$, T'=<$S,s_{\emptyset},\Xi,\psi,\mu',\xi'$> is an MSG of T iff:

1. $\forall s' \in S' : s' \in S$,
2. $\forall t' \in \Xi$: *t'* is a step, where $\|t'\|$ constitute a maximal path in T.
3. $\exists s \in S', s \xrightarrow{{}_M a_x} s' \in T, \forall s'' \in S', \forall w \in T^* :$

$s' \overset{w}{\Rightarrow} s''$

$\Rightarrow \left\{ \exists w' \in \Xi^*, s \overset{w'}{\Rightarrow}_p s'' : [<_M a_x.w>] = [<w'>] \right\}$

Such as

- $\xi' \in 2^T_{fn} \rightarrow 2^M_{fn}$

  $\xi(\varepsilon) =_{def} \varepsilon$

  $\xi'(\{t\} \cup E) =_{def} \xi(t) \cup \xi'(E)$
- $\xi(\xrightarrow{{}_M a_x}) =_{def} x$
- $\mu' : 2^T_{fn} \rightarrow 2^M_{fn}$

  $\mu'(\varepsilon) =_{def} \varepsilon$

  $\mu'(\{t\} \cup E) =_{def} \mu(t) \cup \mu'(E)$
- $\mu(\xrightarrow{{}_M a_x}) =_{def} M$

Where for any step $s \xrightarrow{E}_p s'$, the following conditions are satisfied:

- $\psi(s')=(\psi(s)\backslash\mu'(E)) \cup \xi'(E)$

- $\xi'(E) \notin \psi(s)$-$\mu'(E)$
- $\mu'(E) \subseteq \psi(s')$

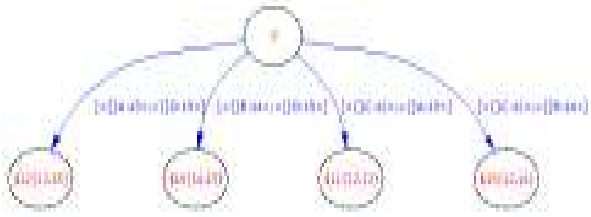

Figure11 : Maximal Steps Graph

Proposition 7 the maximal steps graph preserves deadlock states and liveness property.
Proof. See[24]

4.4. ON THE FLY MAXIMAL STEP GRAPH GENERATION

On the fly maximal step graph generation (Algorithm 8) is the direct extension of the algorithm 3. The difference resides in the line #, which consists in checking for each developed transition, if it can form part of a maximal step (Condition 1) or it is itself a step. LotosGPM [24] is the implementation of this extension.

Algorithm 8 "maximal step graph generation"
Data: LOTOS behavior expression;
Results: An MSG;
Begin
build the initial configuration;
initialize the list of configurations
Confs_List by the initial configuration;
While *Confs_List* Non empty **do**
select and remove an element Conf of *Confs_List*;
Treat Conf configuration;
add Conf to the list of already treated configurations;
add the new resulting configurations to *Confs_List*;
add the resulting transitions to the MLTS;
Substitute MLTS, *Confs_List* and *Confs_treated_List* by using Sub;
Build the maximal transitions steps ..... #
EndWhile
EndAlgo.

Condition 9

Let T=<S,s_o,T,ψ,μ,ξ> be an MSG in generation (in the state s'), and let $s \xrightarrow{t}_p s' \in \Xi$:for all $t : s' \xrightarrow{t}$ generated, if $pt \in Min(\text{Cs})$ then pt is a step of MSG else t is a step.

5. RESULTS

We present in this section 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 size of the MSG with that of the other approaches quoted in this paper. Used tools are:

1. Tina[25]: to generate the step graphs "CSG", the persistent sets "Pset" and persistent step graphs "PSG".
2. Lotostem 2.0: to generate MLTSs
3. LotosGPM: to generate MSGs.

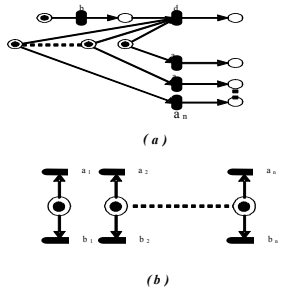

Figure 12 : Studied systems

	States Graph		CSG		PSet		PSG	
n	T	S	T	S	T	S	T	S
4	81	33	81	33	42	22	34	18
6	449	129	449	129	206	72	194	66
8	2305	513	2305	513	1042	266	1026	258
10	11265	2049	11265	2049	5142	1036	5122	1026

	MLTS		MSG	
n	T	S	T	S
4	81	33	3	4
6	449	129	3	4
8	2305	513	3	4
10	11265	2049	3	4

Table 1

Such as T:transition ant S: states.

	States Graph		CSG		PSet		PSG	
n	T	S	T	S	T	S	T	S
4	64	16	16	2	26	11	16	2
6	384	64	64	2	52	22	64	2
8	2048	256	256	2	86	37	256	2

	MLTS		MSG	
n	T	S	T	S
4	216	81	16	17
6	2916	729	64	65
8	20475	6651	256	257

Table 2

The system of Figure 12.(*a*) illustrates a case where MSGs are more effective than the other graphs. Table.1 summarizes the results obtained in a number of transitions and a number of states according to the number of transitions which can be drawn in parallel (*n*). We note that the size of the MSG remains the same one whatever n value. On the other hand, for the system of Figure 12.(*b*) we noticed that the size of an MSG is largely lower than that of the corresponding MLTS (Table.2); however, it is more important than that of the other graphs. In spite of these results, MSGs remain relatively privileged because they represent implicitly more information on the parallel execution of actions.

Proposition 10 CSG, PSet, PSG and MSG approaches are incomparable.
Proof. See [24].

6. CONCLUSION

This paper is a contribution to the state space combinatorial explosion problem. The most widespread partial order approaches are based on structural calculation of the independency relation of actions. We proposed a work based on the joint use of the MLTS model and the covering steps method. The MLTS is indeed a model which made possible the consideration of the branches. The reduced graph is a complete graph preserving the general properties (deadlock states and liveness).

As a perspective, it should be interesting this work 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 [17][26].

REFERENCES

[1] A. Valmari, "Error Detection by Reduced Reachability Graph Generation" *in Proceedings of Application and Theory of Petri Nets*, Springer Verlag, LNCS, 1988.

[2] A. Valmari, " 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] A. Valmari, " A Stubborn Attack on State Explosion", *in Proceedings of CAV'90*, volume 3,pages 25-42, 1990.

[4] P. GodeFroid, " Using Partial Orders to Improve Automatic Verification Methods", *in Proceedings of CAV'90*, volume 3, pages321-340, ACM, DIMACS, 1990.

[5] P. Godefroid and P. Wolper, " A Partial Approach to Model Cheking", *in Proceedings 6th Symp. On Logic in Computer Science*, volume 531, pages 406-415, Amsterdam 1991.

[6] P. Godefroid and P. Wolper, " Using Partial Orders for the Efficient Verification of Deadlock Free-Dom and Safety Properties", *in Formal Methods in Systems Design*, pages 2(2):149-164, 1993.

[7] P. Wolper and P. Godefroid, " Partial-Order Methods for Temporal Verification", *in Proceedings of Concur's93*, volume LNCS 575, 1993.

[8] P. O. Ribet, *Vérification Formelle de Systèmes, Contribution À la Réduction de L'explosion Combinatoire*, Phd thesis, LAAS-CNRS, Toulouse France 2005.

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

[10] F. Vernadat, P. Azéma and F. Michel, " Covering Step Graph", *in Proceedings of Application and Theory of Petri Nets 96*, volume LNCS 1091, Springer Verlag, 1996.

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

[12] F. Vernadat and F. Michel, " Covering Step Graph Preserving Failure Semantics", *in Proccedings of Appplication and Theory of Petri Nets 97*, Springer Verlag, LNCS, 1997.

[13] W. Reisig, "Petri Nets", *in EATCS Monofraphs on Theoretical Computer Science*, Springer Verlag, Berlin, Heidelberg, New York, Tokyo, 1985.

[14] P. O. Ribet and F. Vernadat, *Graphes de Pas Couvrants Vers Une Relation de Conflit Dynamique*, technical report 02065, LAAS, 2002.

[15] D. E. Saidouni, *Sémantique de Maximalité: Application Au Raffinement D'actions Dans LOTOS*, Phd thesis, LAAS, Univerity of Paul Sabastier, Toulouse ,1996.

[16] D. E. Saidouni and N. Belala, Straightforward adaptation of interleaving-based solutions for true concurrency-based logic verification approaches, *in Proceedings of International Conference on Complex Systems (CISC'2004*), Univerity of Jijel, Algeria, 2004.

[17] N. Belala and D. E. Saidouni, " Non-Atomicity in Timed Models", in Proceedings of ACIT'2005, 2005.

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

[19] R. Milner, Communication and Concurrency, volume 92 of LNCS, Springer Verlag 1980.

[20] T. Bolognesi and E. Brinksma, " Introduction to the ISO Specification Language LOTOS", volume 14, Computer Networks and ISDN Systems, 1987.

[21] J. P. Courtiat and D. E. Saidouni, " 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.

[22] D. E. Saidouni and A. Benamira, *La Alpha-Réduction À la Volée Des Systèmes de Transitions Étiquetées Maximales*, technical report, LIRE Laboratory, University of Mentouri, Constantine, Algeria, 2004.

[23] A. Mazurckiewicz, " Trace Theory. In Petri Nets: Applications and Relationships to Other Model of Concurrency", *in Advances in Petri Nets 1986, Part {II;} Proceedings of an Advanced Course*, pages 279-324, Springer Verlag, LNCS 255, 1986.

[24] A. Benamira and D. E. Saidouni, *Contribution À la Résolution de L'explosion Du Graphe D'état Par L'utilisation Conjointe Des Graphes de Pas Couvrants et de la Sémantique de Maximalité*, technical report, LIRE Laboratory, University of Mentouri, Constantine, Algeria, 2005.

[25] *Tina : Time Petri Net Analyzer*, The Tina Toolbox is Property of LAAS-CNRS, 7, avenue du Colonel Roche,31077, Toulouse, France.

[26] N. Belala, *Formalisation Des Systèmes Temps-Réel Avec Durées D'actions*, Master' thesis, LIRE Laboratory, University of Mentouri, Constantine, Algeria, 2005.