Using Maximality-Based Labelled Transition System as a Model for Petri Nets

Djamel Eddine Saidouni, Nabil Belala, and Messaouda Bouneb
Computer Science Department, University of Mentouri, Algeria

Abstract: *This work deals with the specification and the verification of concurrent systems. Our goal is to exploit an implementable model, namely the maximality -based labelled transition system, which permits expressing true-concurrency in a natural way without splitting actions on their start and end events. To do this, we give an operational semantics to build maximality-based labelled transition systems for place/transition Petri nets.*

Keywords: *Maximality-based labelled transition systems, maximality bisimulation, Petri nets.*

Received December 18, 2008; accepted June 16, 2009

1. Introduction

The model of Petri nets is very attractive, not only by its graphical aspect, but also by its ability to capture parallel behavior of systems. One of verification approaches of a Petri net is to generate its marking graph in which nodes represent system states, and arcs represent transitions taking the system from one state to another. After its generation, the marking graph can be seen as a labeled transition system [2]. The generated labeled transition system is used for the verification of the properties of the system specified by the Petri net (model checking, bisimilarity, conformance testing, *etc*. [8, 18, 9]). However, the labeled transition system model ignores parallel execution of transitions. To clarify things, consider the example of the two Petri nets of Figures 1 (a) and (b). The Petri net in Figure 1 (a) represents a system that could execute the actions a and b in parallel, while the Petri net of Figure 1(b) represents a system executing actions a and b sequentially. Nevertheless, these two nets have a single semantic representation given by labeled transition system of Figure 1 (c).

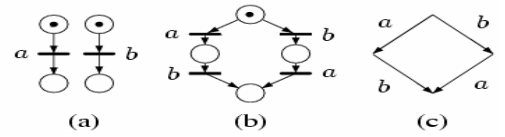


Figure 1. Adapted spiral model [1] for a reflective approach to learning and teaching.

With the assumption that the firing of each transition corresponds to the execution of an indivisible action with null duration (structural and temporal atomicity of actions), the model of labeled transition systems is suitable for the interpretation of concurrent systems behaviors. However, in reality, this assumption of temporal and structural atomicity is not always accepted.

To accept the verification results, realization constraints must be taken into account by both the specification and the used semantic model.

Taking into account the non-atomicity of actions in a system has been deeply studied in the literature through the definition of several semantics supporting the concept of action refinement [10, 11, 12, 1, 7, 5, 13, 14, 15, 19, 23, 27, 29]. Among these semantics, we can cite the maximality semantics which has been defined independently on petri nets and event structures by Devillers and Vogler [14, 15, 30]. In this context, maximality bisimulation relation has been defined and proved to be the coarsest relation preserved by action refinement. In underlying semantic models of Petri nets and event structures, a system with infinite behavior needs an infinite set of events, which makes the underlying structures interesting just for the theoretical point of view [14, 15, 30].

Dealing with implementability, another model named maximality-based labeled transition system has been defined in the literature and used for expressing the semantics of process algebras with the hypothesis that actions are not necessary atomic, i.e., actions are abstractions of finite processes and may elapse on time [11, 23, 28, 25]. The main interest of maximality-based labeled transition system model is that it can be implemented and used in verification [24].

In order to take advantage of the different results developed around the model of maximality-based labeled transition systems, we propose an operational generation method of maximality-based labeled transition systems for place/transition petri nets. As result, maximality bisimulation relations defined on maximality-based labeled transition systems are

extended to petri nets. The proposed approach, which will be presented in section 3, is valid for Petri nets containing recursive behaviors.

2. Preliminaries

2.1. 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 $W:((S\times T)\cup(T\times S))\lozenge N=\{0, 1, 2,...\}$ is the weight function [4, 5]. 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., $|S\cup T|\in N$.

- For $x\in S\cup T$, the pre-set x is defined by $x=\{y\in S\cup T | W(y,x)\neq 0\}$ 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 $M:S\lozenge N$. A marking is generally represented graphically by putting tokens in places.
- 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 $\tau\notin A$ (τ will indicate invisible action, or *silent action*).
- The labeling of a Petri net $N=(S,T,W)$ is a function $\lambda:T\lozenge A\cup\{\tau\}$. 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 λ 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 $\sigma=M_0t_1M_1t_2...$ 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 $[M_0>=\{M \mid \exists t_1t_2...t_n : M_0[t_1t_2...t_n>M\}$.

2.2. Maximality-Based Labelled Transition Systems

A maximality-based labeled transition system is a labeled graph in which transitions are events that represent the beginning of execution of actions [11, 23]. A transition is therefore labeled with the name of the corresponding action. A state is labeled by all the actions that are potentially under execution at this state, these actions are said to be maximal. Given that several actions of the same name can be executed in parallel (auto-concurrency), we identify each start of an action by a distinct identifier, called event name. event names are selected from a countable set denoted by M. As illustration, let us consider the petri net of Figure 2 (a). In his example, only the transitions t_1 and t_2 can be executed in parallel, which corresponds to the parallel execution of two actions of the same name a. For example, we admit that the starts of the two actions a are identified by event names x and y and the start of b by the event name z. Section 3 presents a method for choosing event names allowing their reuse.

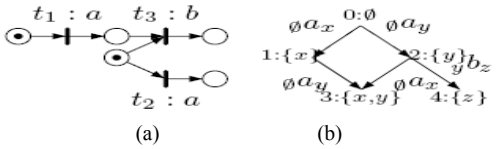


Figure 2. Maximality-based labeled transition system.

In the same maximality-based labeled transition system; this allows treating systems with recursive behaviors. The maximality-based labeled transition system which represents the semantics of the Petri net of Figure 2 (a) is given in Figure 2 (b). In the initial state, no action has begun its execution; the initial state is then labeled by the empty set. Starting from this state, each action a can start, from where the transitions identified by event names x and y. The state 1, labeled by the set $\{x\}$, means that the action a is potentially running at this state. The transition identified by event name y corresponds to the start of the other action a. The state 3, labeled by the set $\{x,y\}$, shows that the two actions a may be executed *simultaneously*, while the state 2, labeled by the set $\{y\}$, shows that in this state only the action a may be under execution. From the state 2, two scenarios are possible: either the second action a begins its execution, leading to the state 3; or it is the action b which starts. It is clear that the action b cannot starts until the end of the first execution of the action a. This causality between the executions of a and b is captured by the set $\{y\}$ associated with the transition leading the system from the state 2 to the state 4. In the resulting state, only the action b may be under execution, from where the labeling of this state by the set $\{z\}$.

Definition 2.1: 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

$\Omega=<S, T, \alpha, \beta, s_0>$ 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.
- and β 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 Ω.
- (Ω,λ) is a transition system labeled by the function λ on an alphabet *Act* called support of (Ω,λ). $(\lambda: T \Diamond Act)$.
- $\psi: S \Diamond 2^M$ is a function which associates to each state the finite set of maximal event names present in this state.
- μ: $T \Diamond 2^M$ is a function which associates to each transition the finite set of event names corresponding to actions that have already begun their execution and of which the end of execution enables this transition.
- $\xi: T \Diamond M$ is a function which associates to each transition the event name identifying its occurrence.

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

Notation 2.1: In what follows, we use the following notations:

- Let $mlts=(\Omega,\lambda,\mu,\xi,\psi)$ be a maximality-based labeled transition system such that $\Omega=<S,T,\alpha,\beta,s_0>$, $t \in T$ is a transition for which $\alpha(t)=s$, $\beta(t)=s'$, $\lambda(t)=a$, $\mu(t)=E$ and $\xi(t)=x$. The transition t will be noted $s \text{--}_E a_x \Diamond s'$.
- Let f: $E \Diamond F$ be a function of domain $Dom(f)=E$ and codomain $Cod(f)=F$, and let D (resp. C) be a subset of E (resp. of F). Restrictions of f w.r.t its domain and codomain are defined by:
  - $f\lceil D=\{(x,y) \in f | x \in D\}$
  - $f\lfloor C=\{(x,y) \in f | y \in C\}$
- $\Im \subseteq 2^{M \times M}$ is the set of all bijective functions between subsets of M.

Definition 2.2: let $mlts1=(\Omega1,\lambda1,\mu1,\xi1,\psi1)$ and $mlts2=(\Omega2,\lambda2,\mu2,\xi2,\psi2)$ be two maximality-based labelled transition systems such that $\Omega1=<S1,T1,\alpha1,\beta1,s1_0>$ and $\Omega2=<S2,T2,\alpha2,\beta2,s2_0>$. $mlts1$ and $mlts2$ are said to be *maximally bisimilar*, noted $mlts1 \approx_m mlts2$, if there is a relation $\Re \subseteq S1 \times S2 \times \Im$ with

1. $(s1_0,s2_0,\varnothing) \in \Re$. Initial states of $mlts1$ and $mlts2$ are related by the relation. Since the sets of maximal events in initial states are empty, the function relating these two sets is empty.
2. If $(s1,s2,f) \in \Re$ then
   - $Dom(f) \subseteq \psi(s1)$ and $Cod(f) \subseteq \psi(s2)$.
   - If $s1 \text{--}_E a_x \Diamond s1'$ then there is $s2 \text{--}_F a_y \Diamond s2'$ such that
   - $\forall (u,v) \in f$, if $u \notin E$ then $v \notin F$
3. $(s1',s2',f') \in \Re$ with $f' = (f\lceil(\psi(s1') - \{x\})) (\psi(s2') - \{y\}) \cup \{(x,y)\}$
   - If $s2 \text{--}_F a_y \Diamond s2'$ then there is $s1 \text{--}_E a_x \Diamond s1'$ such that
   - $\forall (u,v) \in f$, if $v \notin F$ then $u \notin E$

   $(s1',s2',f') \in \Re$ with $f' = (f\lceil(\psi(s1') - \{x\})) \lfloor (\psi(s2') - \{y\}) \cup \{(x,y)\}$

3. Petri Nets and Maximality Semantics

In this section, we introduce through simple examples useful notations and functions for the definition of marking graph associated to a labeled system in a maximality-based approach.

Consider the example of the marked Petri net of Figure 3 (a). With the launch of the transition t_1, it is clear that the firings of transitions t_2 and t_3 are conditioned by end of the action related to t_1. To capture this causal dependence between firings of transitions, we consider that tokens produced by the firing of the transition t_1 are bound to this transition, namely the token in place s_2 and the token in place s_3. We can see that, in the initial state, the token in s_1 is not bound to any transition; this token is called *free* in this state. In the case when t_2 would be fired, it could be argued that the action associated with the firing of t_1 has finished its execution. As a result, the token in s_3 will become free. Resulting marking after the firing of the transition t_2 is given in Figure 3 (c).

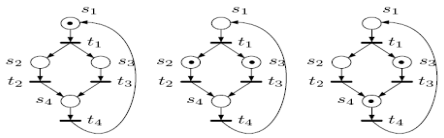


Figure 3. Market petri net.

To distinguish between free and bound tokens in a place, we can imagine that a place is composed of two separated parts. The left part contains free tokens while the right one will contain bound tokens. In a place, the number of free tokens will be denoted by *FT*, while bound tokens set will be noted *BT*. Hence, we obtain the succession of markings of Figure 4.

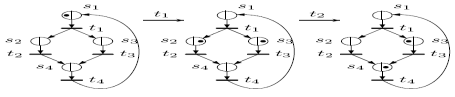
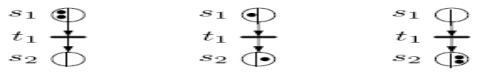


Figure 4. Free and bound tokens in a marking.

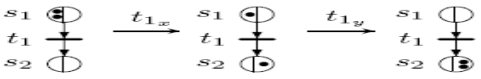
A question that arises is how to bind a token with a transition? to answer the question, we consider the marked petri net of Figure 5 (a). By a firing of the

transition t_1, we obtain the marked Petri net of Figure 5 (b). Starting from this marking, we can see that t_1 is enabled. The firing of this transition will lead to the configuration of Figure 5 (c).



(a) (b) (c)

Figure 5. Tokens linking.



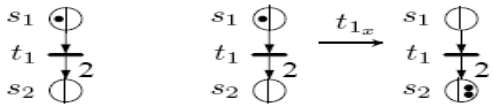
(c_1) (c_2) (c_3)

Figure 6. Firing succession of t_1.

The two tokens of the place s_2 are bound. Indeed, one is bound to the first firing of the transition t_1 while the second one is bound to the second firing of the same transition (two actions associated with t_1 can be executed in parallel). To remove this ambiguity, each firing will be identified by an event name. As a result, binding a token may be characterized by both the transition that produced it and the event name identifying the firing of this transition. The succession of firings of the above example is depicted by Figure 6..

In the configuration C_2, the set of bound tokens in s_2 is $BT=\{t_{1x}\}$ while the set of bound tokens in s_2 of the configuration C_3 is $BT=\{t_{1x}, t_{1y}\}$. The event name x refers to the first firing of the transition t_1 while y refers to the second firing of this transition.

Another problem concerns tokens which are bound to the same transition. To see that, consider the Petri net of Figure 7 (a).



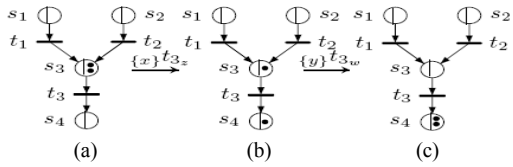
(a) (b)

Figure 7. Net with an output arc of a weight greater than 1.

With the firing of the transition t_1, we obtain the derivation of Figure 7 (b). The right side BT of the place s_2 contains two tokens bound to the firing t_{1x}, i.e., $BT=\{t_{1x}, t_{1x}\}$. Since BT is a set, we consider that a bound token is a tuple (n, t, x) of N×T×M, also denoted nt_x, where n is the number of instances, t is the transition that produce this token and x is the event name associated to the firing of t. We denote by BT= $\{n_1t_{1x1}, n_2t_{2x2}, \ldots\}$ the set (possibly empty) of bound tokens. In the previous example, $BT=\{2t_{1x}\}$. Therefore, the marking of a place s is a pair (FT, BT) where FT is the number of free tokens in s.

Definition 3.1: Let $N=(S, T, W)$ be a Petri net, the marking of N is a function $M:S\Diamond \mathrm{N}\times 2^{\mathrm{N}\times T\times \mathrm{M}}$. Among others, the marking $M(s)$ of a place $s\in S$ is a pair (FT, BT) such that $FT\in$N and $BT\in 2^{\mathrm{N}\times T\times \mathrm{M}}$ denote resp. the number of free tokens and the set (possibly empty) of bound tokens in the place s. In what follows, a Petri net with a marking will be called *configuration*. $|M(s)|$ denote the total number of tokens in a place s. If $M(s)=(FT, BT)$ such that $BT=\{n_1t_{1x1},\ldots,n_mtm_{xm}\}$ then $|M(s)|=FT+|BT|$ with $|BT|=\sum_{i=1,\ldots,m} n_i$ is the cardinal of the bound tokens set in s. Sometimes, we use $FT(s)$ and $BT(s)$ to denote the parts of the marking $M(s)$ of the place s.

Now, we consider the identification of consumed tokens by a firing of a transition. To do this, let us consider the Petri net of Figure 8 (a).



(a) (b) (c)

Figure 8. Event names identifying consumed tokens.

We assume that one token of s_3 is bound to the firing of t_1 (t_{1x}) and the other one is bound to the firing of t_2 (t_{2y}). Among bound tokens in s_3, we want to know the consumed token in the first firing of t_3 and that consumed in the second firing of the same transition. This information is essential to know, in each configuration, the actions (associated with transitions) which have finished their execution. To do this, we associate at the level of a firing the event names identifying bound tokens consumed by this firing. This gives us firing sequence of Figure 8.

In the following paragraph, we give some preliminary definitions that will enable us to propose a generation method of a marking graph in the context of the maximality semantics.

3.1. Preliminary Definitions

Definition 3.2: let (S, T, W) be a Petri net with a marking M:

- 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 ψ will be used to calculate this set, it can be defined as $\psi(M)=\cup_{s\in S}\cup_{i=1,\ldots,ms} xs_i$ such that $M(s)=(FT, BT)$ with $BT=\{(ns_1, ts_1, xs_1),\ldots,(ns_{ms}, ts_{ms}, xs_{ms})\}$.
- Let N⊂M be a non-empty finite set of event names, *makefree*(N, M) is defined recursively by: *makefree* $(\{x_1, x_2, \ldots,x_n\}, M)$= *makefree*$(\{x_2,\ldots,x_n\}$,
- *Makefree* $(\{x_1\}, M))$ *makefree* $(\{x\}, M)$= M' such that for all $s\in S$, if $(s)=(FT, BT)$ then:
- If there is $(n, t, x)\in BT$ then $M'(s) = (FT+n, BT-\{(n,t,x)\})$ (Conversion of n bound tokens identified by the event name x to free tokens).
- Otherwise, $M'(s)= M(s)$.

- 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).
- The marking M is said to be minimal for the firing of the transition t iff $|M(s)|=W(s, t)$ for all $s \in S$.
  Let M_1 and M_2 be two markings of the Petri net (S, T, W). $M_1 \in M_2$ 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 $BT_1 \in BT_2$ such that the relation $\in$ is extended to bound tokens sets as follows: $BT_1 \in BT_2$ iff $\forall (n_1, t, x) \in BT_1$, $\exists (n_2, t, x) \in BT_2$ such that $n_1 \leq n_2$.
- Let M_1 and M_2 be two markings of the Petri net (S, T ,W) such that $M_1 \in M_2$. 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 $\forall (n_1,t,x) \in BT_1$, $(n_2,t,x) \in BT_2$, if $n_1 \neq n_2$ then $(n_2-n_1,t,x) \in BT_3$.
- $Min(M,t)=\{M' | M' \in M\}$ and M' is minimal for the firing of t.
- Let M be a set. The function $get: 2^M-\{\varnothing\} \Diamond M$ is a function which satisfies $get(E) \in E$ for any $E \in 2^M-\{\varnothing\}$.
- 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.

3.2. Construction of Marking Graph

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 M_0 and the transitions between states are labeled according to the derivation rule of Definition 3.3.

Definition 3.3: let M be a reachable marking of the marked Petri net (S,T,W,M_0), $t \in enabled(M)$ then for all $M'' \in Min(M,t)$, $E=\psi(M'')$ and $M'''=makefree(E,M-M'')$; the following derivation is possible: $M \text{--}_E t_x \Diamond M'$ (also denoted by $(M,_E t_x,M')$) such that

- 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.
- $x = get(M-\psi(M'''))$ and
- $M' = occur(t,x,M''')$.

3.3. Properties

Proposition 3.1: let $\Sigma=(S,T,W,M_0,\lambda)$ be a labeled system and Mg its marking graph built according to Definition 3.3, then the structure $\Sigma_{mlts}=(Mg,\lambda,\mu,\xi,\psi)$ is a maximality-based labeled transition system with:

- $Mg=<Sg,Tg,\alpha,\beta,M_0>$ 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.
- $Tg=\{(M,_E t_x,M')\}$ such that $M,M' \in Sg$ and $(M,_E t_x,M')$ is a valid derivation.
- For $(M,_E t_x,M') \in Tg$ we have $\alpha((M,_E t_x,M'))=M$ and $\beta((M,_E t_x,M'))=M'$.
- $\psi : Sg \Diamond 2^M$ is the function defined in Definition 3.2.

For $d=(M,_E t_x,M') \in Tg$ we put $\lambda(d)=\lambda(t)$, $\mu(d)=E$ and $\xi(d)=x$.

Proof: let us observe at first that the initial marking M_0 contains only free tokens, therefore $\psi(M0)=\varnothing$. On the other hand, for $d=(M,_E t_x,M') \in Tg$, according to Definition 3.3, $\mu(d)=E \subseteq \psi(\alpha(d))=\psi(M)$, $\xi(t) \notin \psi(M)-\mu(d)$ and $\psi(\beta(t))=\psi(M')=(\psi(M)-\mu(d)) \cup \{\xi(d)\}$, then the result.

Proposition 3.2: given a marked Petri net (S, T, W, M_0), then the set of sequences of generated transitions in an interleaving approach is the same to the set of sequences generated in a maximality-based approach.

Proof: derive directly from the fact that in a maximality-based approach, the firing condition of a transition takes into account only the number of tokens in places and not the nature of these tokens. Note that information related to the origin of tokens (BT set) is used only to decorate the graph with event names used in the generation of the maximality-based labeled transition system.

Definition 3.4: let $\sum_1=(S_1,T_1,W_1,M1_0,\lambda_1)$ and $\sum_2=(S_2,T_2,W_2,M2_0,\lambda_2)$ be two labeled systems. $\sum_1$ and $\sum_2$ are said to be *maximally bisimilar* iff their respective maximality-based labeled transition systems are maximally bisimilar.

Example 3.1: consider the example of the two labeled systems Sys_1 and Sys_2 in Figure 9 (this example is from [14]). By applying the proposed approach, the corresponding maximality-based labeled transition systems are given by Figure 10. The reader may easily check that these two systems are maximally bisimilar.

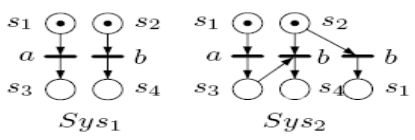


Figure 9. Maximally bisimilar systems.

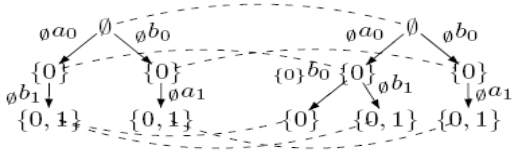


Figure 10. Maximally bisimilar MLTSs.

4. Conclusion

In this paper, we proposed an operational method of generation of maximality-based labeled transition systems associated to Petri nets. Thus, correctness properties relative to systems specified by a Petri net can be checked on maximality-based labeled transition systems which correspond to this Petri net. In addition, maximality-based labeled transition system structure includes information on the parallel execution of actions. This structure enables us to more easily express properties relating to the parallel execution of actions without splitting actions in their start and end events. We note that splitting actions in start and end events to capture the parallel execution of the actions will contribute directly to the problem of combinatorial explosion of the state paragraph.

In [3], an on-the-fly generation algorithm of maximality-based labeled transition systems, reduced modulo α-equivalence, relating to behavior expressions written in basic LOTOS process algebra [6] was proposed and integrated in FOCOVE environment [26]. We think that this algorithm adapts easily to the context of Petri nets.

A future work consists on aggregating equivalent derivations of transitions according to maximality bisimulation relation. As illustration, consider the Petri net of Figure 11 (a). By applying the proposed approach, corresponding maximality-based labeled transition system of this Petri net is given by Figure 11 (b).

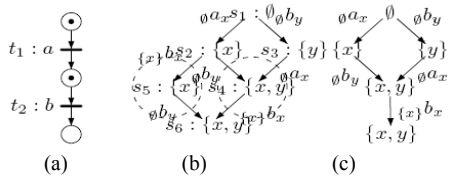


Figure 11. Operational semantics of a Petri net in terms of MLTS.

Notice that from state s_2, transitions leading respectively to states s_4 and s_5 are due to the firing of the same transition t_2. In the first firing, the token of the initial marking is used whereas in the second firing, the used token is that produced by the firing of t_1. On the other hand, the derivation by *b* leading to state s_4 is not conditioned by the end of the action *a*, while the derivation leading to state s_5 is conditioned by the end of *a*. Therefore, we think that it possible to omit the derivations $s_2 \lozenge s_5 \lozenge s_6$, *i.e;* the maximality-based labeled transition system of Figure 11 (c) preserves the behavior of the Petri net of Figure 11 (a).

In another work, we think that we can use quantum evolutionary and genetic algorithms for reducing maximality-based labeled transition systems according to behavioral relations in the same way as in [20, 22, 21].

References

[1] Aceto L. and Hennessy M., "Adding Action Refinement to Finite Process Algebra", *in Proceedings of ICALP'91*, Springer-Verlag, pp. 506-519, 1991.

[2] Arnold A., *Systèmes de Transitions Finis et Sémantique des Processus Communicants*, Masson, Paris, 1992.

[3] Benamira A. and Saïdouni E., "Consideration of the Covering Steps in the Maximality-Based Labeled Transition Systems", *in Proceedings of International Arab Conference on Information Technology (ACIT'2006)*, Jordan, pp. 19-21, 2006.

[4] Best E. and Devillers R., "Sequential and Concurrent Behaviour in Petri Net Theory," *Theoretical Computer Science*, vol. 55, no. 1, pp. 87-136, 1987.

[5] Best E., Devillers R., Kiehn A., and Pomello L., "Concurrent Bisimulations in Petri Nets," *Acta Informatica*, vol. 28, pp. 231-264, 1991.

[6] Bolognesi T. and Brinksma E., "Introduction to the ISO Specification Language LOTOS," *Computer Journal of Computer Networks and ISDN Systems*, vol. 14, no. 1, pp. 25-59, 1987.

[7] Boudol G. and Castellani I., "Concurrency and Atomicity," *Computer Journal of TCS*, vol. 59, no., pp. 1-60, 1988.

[8] Clarke M., Emerson E., and Sistla P., "Automatic Verification of Finite State Concurrent Systems Using Temporal Logic Specifications," *Computer Journal of ACM Transactions on Programming Languages and Systems*, vol. 8, no. 2, pp. 244-263, 1986.

[9] Cleveland R. and Hennessy M., "Testing Equivalence as a Bisimulation Equivalence," *Computer Journal of Formal Aspects of Computing*, vol. 5, no. 3, pp. 1-20, 1993.

[10] Courtiat P. and Saïdouni E., "Action Refinement in LOTOS," *in Proceedings of Protocol Specification, Testing and Verification (PSTV'93)*, Holland, pp. 341-354, 1994.

[11] Courtiat P. and Saïdouni E., "Relating Maximality-Based Semantics to Action Refinement in Process Algebras," *in the 7th International Conférence on Formal Description Techniques (FORTE'94)*, pp. 293-308, Chapman & Hall, 1995.

[12] Darondeau P. and Degano P., "Refinement of Actions in Event Structures and Causal Trees," *Computer Journal of TCS*, vol. 118, no. 4, pp. 21-48, 1993.

[13] Degano P. and Gorrieri R., "Atomic Refinement in Process Description Languages," *in Proceedings of Mathematical Foundations of Computer Science*, LNCS, vol. 520, no. 2, pp. 121-130, Springer-Verlag, 1991.

[14] Devillers R., "Maximality Preservation and the ST-Idea for Action Refinement," *Computer Journal of Advances in Petri Nets*, vol. 609, no. 3, pp. 108-151, Springer-Verlag, 1992.
[15] Devillers R., "Maximality Preserving Bisimulation", *Computer Journal of TCS*, vol. 102, no. 6, pp. 165-183, 1992.
[16] Dijkstra W., "Hierarchical ordering of Sequential Processes," *Computer Journal of Acta Informatica*, vol. 1, no. 2, pp. 115-138, 1971.
[17] Emerson A., "Temporal and Modal Logic," *in Handbook of Theoretical Computer Science: Formal Models and Semantics,* vol. B, no. 5, pp. 995-1072, 1990.
[18] Fernandez J. and Mounier L., "A Tool Set for Deciding Behavioral Equivalences," *in Proceedings of the 2nd International Conference on Concurrency Theory*, UK, pp. 23-42, 1991.
[19] Janssen W., Poel M., and Zwiers J., "Action Systems and Action Refinement in the Development of Parallel Systems," *in Proceedings of CONCUR'91*, LNCS, vol. 527, no. 9, pp. 298-316. Springer-Verlag, 1991.
[20] Layeb A. and Saïdouni 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.
[21] Layeb A. and Saïdouni E., "A New Quantum Evolutionary Local Search Algorithm for Max 3-SAT Problem," *in Proceedings of the 3rd International Workshop on Hybrid Artificial Intelligence Systems (HAIS'08),* Spain, 2006.
[22] Layeb A. and Saïdouni 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.
[23] Saïdouni E., Sémantique de Maximalité : Application au Raffinement d'Actions en LOTOS, *PhD Thesis*, du Colonel Roche, France, 1996.
[24] Saïdouni 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.
[25] Saïdouni D. and Belala N., "Actions Duration in Timed Models," *in Proceedings of International Arab Conference on Information Technology (ACIT'2006)*, Jordan, pp. 19-21, 2006.
[26] Saïdouni E., Benamira A., Belala N., and Arfi F., "FOCOVE: Formal Concurrency Verification Environment for Complex Systems," *in Mediterranean Conference on Intelligent Systems and Application (CISA'2008)*, *in American Institute of Physics Conference Proceedings*, pp. 375-380.
[27] Saïdouni E. and Courtiat P., "Syntactic Action Refinement in Presence of Multiway Synchronization," *in Workshop on Semantics of Specification Languages (SoSL'93)*, Utrecht, pp. 289-303, 1994.
[28] Saïdouni E. and Courtiat P., "Prise en Compte des Durées d'action Dans les Algèbres de Processus Par L'utilisation de la Sémantique de Maximalité," *in Proceedings of CFIP'2003*, France, pp.301-310, 2003.
[29] Van J., "The Refinement Theorem for ST-Bisimulation Semantics," *in Proceedings of IFIP Working Conference on Programming Concepts and Methods*, Holland, pp. 22-25, 1990.
[30] Vogler W., "Bisimulation and action refinement", *Computer Journal of TCS*, vol. 114, no. 3, pp. 173-200, 1993.

Djamel Eddine Saidouni obtained his BEng degree from University of Mentouri Constantine, Algeria, in 1990. He prepared his DEA in communicating systems. He obtained his PhD in theoretical computer science and concurrency from the University of Paul Sabatier, Toulouse, France in 1996.

Nabil Belala obtained his BEng degree from University of Mentouri Constantine, Algeria, in June 2002, and his MSc degree in computer science at the University of Mentouri, Constantine. From October 2005, he started his PhD thesis on timed models and their use in formal specification and verification of real-time systems.

Messaouda Bouneb obtained her BEng degree from University of Mentouri Constantine, Algeria, in June 2005. In February 2009, she obtained her MSc degree in computer science at the University of El Arbi Ben-Mhidi Oum El-Bouaghi, Algeria.