Aggregation of Transitions in Marking Graph Generation Based on Maximality Semantics for Petri Nets

Djamel Eddine Saïdouni, Nabil Belala and Messaouda Bouneb
Computer Science Dept., University of Mentouri, 25000 Constantine, Algeria
saidounid@hotmail.com, nabil.belala@umc.edu.dz, bounebmessaouda@hotmail.com

Abstract

In this paper, we propose an operational semantics to build maximality-based labeled transition systems (MLTS) from Place/Transition Petri nets while performing aggregation of equivalent derivations of transitions according to maximality bisimulation relation. We show that generated MLTS are equivalent to MLTS generated without aggregation. As illustration, we apply results on a ticket reservation system.

Keywords: Maximality-based labeled transition systems, Maximality bisimulation, Aggregation of transitions, Petri nets.

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 system properties specified by the Petri net (model checking, bisimilarity, conformance testing, etc. [7, 17, 8]). However, the labeled transition system model cannot distinguish between sequential and parallel execution of transitions. On the other hand, this model considers the assumption of structural and temporal atomicity of actions. Nevertheless, this assumption is not always accepted in reality. For instance, an instruction written in a high level programming language is interpreted as an interruptible sequence of machine instructions and each machine instruction is of non-null duration.

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 [9, 10, 11, 1, 6, 5, 12, 13, 14, 18, 20, 24, 27]. Among these semantics, we can cite the maximality semantics which has been defined independently on Petri nets and event structures by Devillers and Vogler [13, 14, 29]. In this context, maximality bisimulation relation has been defined and proved to be the coarsest relation preserved by action refinement and supporting action duration. 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.

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. The main interest of maximality-based labeled transition system model is that it can be implemented and used in verification [21].

In order to take advantage of the different results developed around the model of maximality-based labeled transition systems, we proposed in [22] an operational generation method of maximality-based labeled transition systems for Places/Transitions 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 recalled in Section 3, is valid for Petri nets with cyclic behavior.

As example, consider the Petri net of Figure 1.(a). By applying the approach of [22], corresponding maximality-based labeled transition system of this Petri net is given by Figure 1.(b).

FIGURE 1: Operational semantics of a Petri net in terms of MLTS

At first, recall that a maximality-based labeled transition system is a graph labeled on both states and transitions. Each state is labeled by a set of event names. Each event name identifies the start of execution of an action (eventually under execution) which occurred before this state. This action is said to be *potentially under execution* in this state. A transition between two states s_i and s_j is labeled by a 3-uple (M, a, x) (denoted ${}_M a_x$) where x is the event name identifying the start of execution of the action a and M denotes the set of event names representing the causes of the action a . Elements of M belong to state s_i . Occurrence of this transition terminates actions identified by M , thus, the set of event names corresponding to state s_j is that of s_i from which we subtract the set M and add the event name x . Formal definition of a maximality-based labeled transition system will be given in Section 2.2.

In the initial state (state s_1) of the maximality-based labeled transition system of Figure 1.(b), no action is running, from where the association of the empty set with this state. From state s_1 , actions a and b can start their execution independently, their starts are respectively identified by event names x and y . a and b can be launched in any order. The set $\{x\}$ (resp. $\{y\}$) in state s_2 (resp. s_3) stipulates that the action a (resp. b) are potentially under execution in this state. $\{x, y\}$ in s_4 shows that actions a and b can be executed simultaneously.

Note that when the system is in state s_2 , while the action a has not been terminated yet, the only evolution concerns the start of b . However, when a terminates, we can start the action b caused by a or the action b which is independent from the end of a . Resulting states are respectively s_4 and s_5 . We can observe that from state s_5 , the start of b is always possible. However, the same ending constraint of a is imposed for the execution of b at the level of state s_4 . Note that causal dependence between execution of b across from the action a is captured by the consumption of the produced token coming from the transition t_1 during the firing of t_2 in the Petri net.

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, such as we noted above, 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, is it possible to omit the derivations $s_2 \rightarrow s_5 \rightarrow s_6$ in the maximality-based labeled transition system? In other words, did the maximality-based labeled transition system of Figure 1.(c) preserve the behavior of the Petri net of Figure 1.(a)?

In this paper, we propose an operational semantics for generating maximality-based labeled transition systems. This semantics performs aggregation of derivations according to the idea explained in the previous example. We show that generated maximality-based labeled transition system is equivalent to that generated by operational semantics proposed in [22] modulo the maximality bisimulation relation recalled in Section 2.2.

In Section 5, we show through an example that the proposed approach significantly reduces the size of the maximality-based labeled transition system. Finally, Section 6 gives some conclusions of this work. Proofs can be found in [23].

2. PRELIMINARIES

2.1. Petri nets related definitions [4, 5]

- 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 $W : ((S \times T) \cup (T \times S)) \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$ 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 ${}^\bullet x$ is defined by ${}^\bullet x = \{y \in S \cup T | W(y, x) \neq 0\}$ and the post-set $x^\bullet$ is defined by $x^\bullet = \{y \in S \cup T | W(x, y) \neq 0\}$.
- The *marking* of a Petri net (S, T, W) is defined as a function $M : S \rightarrow \mathbb{N}$. 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 $\mathcal{A}$ is a finite set; we suppose that $\tau \notin \mathcal{A}$ (τ will indicate invisible action, or *silent action*).
- The labeling of a Petri net $N = (S, T, W)$ is a function $\lambda : T \rightarrow \mathcal{A} \cup \{\tau\}$. If $\lambda(t) \in \mathcal{A}$ 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 $a \in \mathcal{A}$ 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 $\sigma = M_0 t_1 M_1 t_2 \dots$ is an occurrence sequence iff $M_{i-1}[t_i]M_i$ for $1 \leq i$. A sequence $t_1 t_2 \dots$ is a transition sequence starting with M iff there is an occurrence sequence $M t_1 M_1 t_2 \dots$. If a finite sequence $t_1 t_2 \dots t_n$ leads from M to M' , we write $M[t_1 t_2 \dots t_n]M'$. The set of reachable markings of a marked Petri net (S, T, W, M_0) is defined as $[M_0] = \{M | \exists t_1 t_2 \dots t_n : M_0[t_1 t_2 \dots t_n]M\}$.

2.2. Maximality-based labeled transition systems [10, 20]

Definition 2.1 Let $\mathcal{M}$ be a countable set of event names, a maximality-based labeled transition system of support $\mathcal{M}$ 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.
 - α 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) . ($\lambda : T \rightarrow Act$).
- $\psi : S \rightarrow 2^{\mathcal{M}}$ is a function which associates to each state the finite set of maximal event names present in this state.
- $\mu : T \rightarrow 2^{\mathcal{M}}$ 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.
- $\xi : T \rightarrow \mathcal{M}$ is a function which associates to each transition the event name identifying its occurrence.

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

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

- Let $mlds = (\Omega, \lambda, \mu, \xi, \psi)$ a maximality-based labeled transition system such that $\Omega = \langle S, T, \alpha, \beta, s_0 \rangle$. $t \in T$ 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 $s \xrightarrow{E a x} s'$.
- Let $f : E \rightarrow F$ be a function of domain $Dom(f) = E$ and codomain $Cod(f) = F$, and let D (respectively C) be a subset of E (respectively of F). Restrictions of f with respect to its domain and codomain are defined by:
 - $f \upharpoonright D = \{(x, y) \in f | x \in D\}$
 - $f \downharpoonright C = \{(x, y) \in f | y \in C\}$

- $\mathfrak{F} \subseteq 2^{\mathcal{M} \times \mathcal{M}}$ is the set of all bijective functions between subsets of $\mathcal{M}$.
- Id_A is the identity function on elements of a set A .

Definition 2.2 Let $mlts_1 = (\Omega_1, \lambda_1, \mu_1, \xi_1, \psi_1)$ and $mlts_2 = (\Omega_2, \lambda_2, \mu_2, \xi_2, \psi_2)$ be two maximality-based labeled transition systems such that $\Omega_1 = \langle S_1, T_1, \alpha_1, \beta_1, s_{10} \rangle$ and $\Omega_2 = \langle S_2, T_2, \alpha_2, \beta_2, s_{20} \rangle$. $mlts_1$ and $mlts_2$ are said to be maximally bisimilar, noted $mlts_1 \approx_m mlts_2$, if there is a relation $\mathfrak{R} \subseteq S_1 \times S_2 \times \mathfrak{F}$ with

1. $(s_{10}, s_{20}, \emptyset) \in \mathfrak{R}$. Initial states of $mlts_1$ and $mlts_2$ 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 $(s_1, s_2, f) \in \mathfrak{R}$ then
 - (a) $Dom(f) \subseteq \psi(s_1)$ and $Cod(f) \subseteq \psi(s_2)$.
 - (b) If $s_1 \xrightarrow{E a_x} s_1'$ then there is $s_2 \xrightarrow{F a_y} s_2'$ such that
 - i. $\forall (u, v) \in f$, if $u \notin E$ then $v \notin F$
 - ii. $(s_1', s_2', f') \in \mathfrak{R}$ with $f' = (f \upharpoonright ((\psi(s_1') - \{x\}))) \downarrow ((\psi(s_2') - \{y\}) \cup \{(x, y)\})$
 - (c) If $s_2 \xrightarrow{F a_y} s_2'$ then there is $s_1 \xrightarrow{E a_x} s_1'$ such that
 - i. $\forall (u, v) \in f$, if $v \notin F$ then $u \notin E$
 - ii. $(s_1', s_2', f') \in \mathfrak{R}$ with $f' = (f \upharpoonright ((\psi(s_1') - \{x\}))) \downarrow ((\psi(s_2') - \{y\}) \cup \{(x, y)\})$

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 2.(a). With the launch of the transition t_1 , it is clear that the

FIGURE 2: Marked Petri net

firings of transitions t_2 and t_3 are conditioned by the 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 2.(c).

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 $\mathcal{FT}$, while bound tokens set will be denoted by $\mathcal{BT}$. Hence, we obtain the succession of markings of Figure 3.

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 4.(a).

By a firing of the transition t_1 , we obtain the marked Petri net of Figure 4.(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 4.(c).

FIGURE 3: Free tokens and bound tokens in a marking

FIGURE 4: Tokens linking

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 5.

 FIGURE 5: Firing succession of t_1

In the configuration C_2 , the set of bound tokens in s_2 is $\mathcal{BT} = \{t_{1_x}\}$ while the set of bound tokens in s_2 of the configuration C_3 is $\mathcal{BT} = \{t_{1_x}, t_{1_y}\}$. 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 6.(a).

With the firing of the transition t_1 , we obtain the derivation of Figure 6.(b).

The right side $\mathcal{BT}$ of the place s_2 contains two tokens bound to the firing t_{1_x} , i.e. $\mathcal{BT} = \{t_{1_x}, t_{1_x}\}$. Since $\mathcal{BT}$ is a set, we consider that a bound token is a tuple (n, t, x) of $\mathbb{N} \times T \times \mathcal{M}$, 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

FIGURE 6: Petri net with an output arc of a weight greater than 1

$\mathcal{BT} = \{n_1 t_{1x_1}, n_2 t_{2x_2}, \dots\}$ the set (possibly empty) of bound tokens. In the previous example, $\mathcal{BT} = \{2t_{1x}\}$. Therefore, the marking of a place s is a pair $(\mathcal{FT}, \mathcal{BT})$ where $\mathcal{FT}$ 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 $M : S \rightarrow \mathbb{N} \times 2^{\mathbb{N} \times T \times \mathcal{M}}$. Among others, the marking $M(s)$ of a place $s \in S$ is a pair $(\mathcal{FT}, \mathcal{BT})$ such that $\mathcal{FT} \in \mathbb{N}$ and $\mathcal{BT} \in 2^{\mathbb{N} \times T \times \mathcal{M}}$ denote respectively 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 $M(s) = (\mathcal{FT}, \mathcal{BT})$ such that $\mathcal{BT} = \{n_1 t_{1x_1}, \dots, n_m t_{mx_m}\}$ then $|M(s)| = \mathcal{FT} + |\mathcal{BT}|$ with $|\mathcal{BT}| = \sum_{i=1}^m n_i$ is the cardinal of the bound tokens set in s . Sometimes, we use $\mathcal{FT}(s)$ and $\mathcal{BT}(s)$ 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 7.(a).

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 7.

FIGURE 7: Event names identifying consumed tokens

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 $\psi(M) = \bigcup_{s \in S} \bigcup_{i=1}^{ms} xs_i$ such that $M(s) = (\mathcal{FT}, \mathcal{BT})$ with $\mathcal{BT} = \{(ns_1, ts_1, xs_1), \dots, (ns_{ms}, ts_{ms}, xs_{ms})\}$.
- Let $N \subset \mathcal{M}$ be a non-empty finite set of event names, $makefree(N, M)$ is defined recursively by:
 - $makefree(\{x_1, x_2, \dots, x_n\}, M) = makefree(\{x_2, \dots, x_n\}, makefree(\{x_1\}, M))$
 - $makefree(\{x\}, M) = M'$ such that for all $s \in S$, if $M(s) = (\mathcal{FT}, \mathcal{BT})$ then

- * If there is $(n, t, x) \in \mathcal{BT}$ then $M'(s) = (\mathcal{FT} + n, \mathcal{BT} - \{(n, t, x)\})$ (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 $\text{enabled}(M)$.
- 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) . $M_1 \subseteq M_2$ iff $\forall s \in S$, if $M_1(s) = (\mathcal{FT}_1, \mathcal{BT}_1)$ and $M_2(s) = (\mathcal{FT}_2, \mathcal{BT}_2)$ then $\mathcal{FT}_1 \leq \mathcal{FT}_2$ and $\mathcal{BT}_1 \subseteq \mathcal{BT}_2$ such that the relation $\subseteq$ is extended to bound tokens sets as follows:
 - $\mathcal{BT}_1 \subseteq \mathcal{BT}_2$ iff $\forall (n_1, t, x) \in \mathcal{BT}_1, \exists (n_2, t, x) \in \mathcal{BT}_2$ 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 $M_1 \subseteq M_2$. 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 $M_1(s) = (\mathcal{FT}_1, \mathcal{BT}_1)$ and $M_2(s) = (\mathcal{FT}_2, \mathcal{BT}_2)$ then $M_3(s) = (\mathcal{FT}_3, \mathcal{BT}_3)$ with
 - $\mathcal{FT}_3 = \mathcal{FT}_2 - \mathcal{FT}_1$
 - $\forall (n_1, t, x) \in \mathcal{BT}_1, (n_2, t, x) \in \mathcal{BT}_2$ if $n_1 \neq n_2$ then $(n_2 - n_1, t, x) \in \mathcal{BT}_3$.
- $\text{Min}(M, t) = \{M' | M' \in \mathcal{M}\}$ and M' is minimal for the firing of t .
- Let $\mathcal{M}$ be a set. The function $\text{get} : 2^{\mathcal{M}} - \{\emptyset\} \rightarrow \mathcal{M}$ is a function which satisfies $\text{get}(E) \in E$ for any $E \in 2^{\mathcal{M}} - \{\emptyset\}$.
- Given a marking M , a transition t and an event name $x \notin \psi(M)$, $\text{occur}(t, x, M) = M'$ such that $\forall s \in S$, if $M(s) = (\mathcal{FT}, \mathcal{BT})$ then $M'(s) = (\mathcal{FT}, \mathcal{BT}')$ with $\mathcal{BT}' = \mathcal{BT} \cup \{W(t, s), t, x\}$ if $W(t, s) \neq 0$ and $\mathcal{BT}' = \mathcal{BT}$ 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) , $t \in \text{enabled}(M)$ then for all $M'' \in \text{Min}(M, t)$, $E = \psi(M'')$ and $M''' = \text{makefree}(E, M - M'')$; the following derivation is possible: $M \xrightarrow{E, t_x} M'$ (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 .
- $x = \text{get}(\mathcal{M} - \psi(M'''))$ and
- $M' = \text{occur}(t, x, M''')$.

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:

- $Mg = \langle Sg, Tg, \alpha, \beta, M_0 \rangle$ 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 $\beta((M, E, t_x, M')) = M'$
- $\psi : Sg \rightarrow 2^{\mathcal{M}}$ 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 $\psi(M_0) = \emptyset$. 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 $\psi(\beta(d)) = \psi(M') = (\psi(M) - \mu(d)) \cup \{\xi(d)\}$, 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 ($\mathcal{BT}$ 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 $\Sigma_1 = (S_1, T_1, W_1, M1_0, \lambda_1)$ and $\Sigma_2 = (S_2, T_2, W_2, M2_0, \lambda_2)$ be two labeled systems. Σ_1 and Σ_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 8.(a) (this example is from [13]). By applying the proposed approach, the corresponding maximality-based labeled transition systems are given by Figure 8.(b). The reader may easily check that these two systems are maximally bisimilar.

FIGURE 8: Maximally bisimilar systems

4. GENERATION OF MARKING GRAPH WITH AGGREGATION OF TRANSITIONS

4.1. Operational maximality semantics for Petri nets with aggregation of transitions

Usually, marking graph is generated in the same manner as the previous approach, thus we keep the same basic definitions, but to achieve our goal we must change the semantics of the function Min . In this case, a minimal marking for the firing of a transition t is considered as an element of the set $Min(M, t)$ only if for each place of this marking, bound tokens are only taken in the case when the free part does not satisfy the pre-condition of this transition. Therefore, we can ensure that a transition t will be executed sequentially after a transition t' if it cannot be executed independently with this same transition t' .

Formally, $Min(M, t)$ is the set of markings $M' \in M$ such that for any state s where $M(s) = (\mathcal{FT}, \mathcal{BT})$, $M'(s)$ is defined as follows:

$$M'(s) = \begin{cases} (w(s, t), \emptyset) & \text{if } \mathcal{FT} \geq w(s, t) \\ (\mathcal{FT}, \mathcal{BT}') & \text{otherwise} \end{cases}$$

with $\mathcal{BT}' \in \mathcal{BT}$ and $|\mathcal{BT}'| = w(s, t) - \mathcal{FT}$

4.2. Maximality bisimulation relation on transitions

In this section, we present a maximality bisimulation relation according to Petri net transitions instead of actions.

Definition 4.1 Let $Mark$ be a set of markings, T a set of transitions and $\rightarrow$ a derivation relation between markings as defined in Definition 3.3.

- A relation $\mathcal{R} \subseteq 2^{Mark} \times 2^{Mark} \times \mathfrak{F}$ is said a maximality bisimulation relation according to a set of transitions T iff $\forall (M_i, M'_i, Id_{A_i}) \in \mathcal{R}$, $A_i \subseteq \psi(M_i)$ and $A_i \subseteq \psi(M'_i)$,
 1. If $M_i \xrightarrow{E_i t_{ix}} M_j$ then $\exists M'_j \xrightarrow{E'_j t_{iy}} M'_j / \forall u \in A_i$ if $u \notin E_i$ then $u \notin E'_i$ and for $z = get(\mathcal{M} - ((\psi(M_i) - E_i) \cup (\psi(M'_i) - E'_i))) : (M_j[z/x], M'_j[z/y], Id_{A_{i+1}}) \in \mathcal{R} / A_{i+1} = (A_i - E_i) \cup \{z\}$.
 2. If $M'_i \xrightarrow{E'_i t_{iy}} M'_j$ then $\exists M_j \xrightarrow{E_i t_{ix}} M_j / \forall u \in A_i$ if $u \notin E'_i$ then $u \notin E_i$ and for $z = get(\mathcal{M} - ((\psi(M_i) - E_i) \cup (\psi(M'_i) - E'_i))) : (M_j[z/x], M'_j[z/y], Id_{A_{i+1}}) \in \mathcal{R} / A_{i+1} = (A_i - E'_i) \cup \{z\}$.
- Two labeled systems $\Sigma_1 = (S_1, T, W_1, M_0^1, \lambda_1)$ and $\Sigma_2 = (S_2, T, W_2, M_0^2, \lambda_2)$ are said to be maximally bisimilar according to T , noted $\Sigma_1 \approx_m^T \Sigma_2$, iff there exists a maximality bisimulation relation $\mathcal{R}$ according to T such that $(M_0^1, M_0^2, \emptyset) \in \mathcal{R}$.

- $M_1 \approx_m^T / f M_2$ denotes the fact that $(M_1, M_2, f) \in \mathfrak{R}$.

Corollary 4.1 $\approx_m^T \subseteq \approx_m$

Proposition 4.1 Let M be a marking with $A \subseteq \psi(M)$ then

1. $M \approx_m^T / Id_{\psi(M)} M$
2. $M \approx_m^T / Id_A M$
3. If $B = \psi(M) - A$ then $makefree(B, M) \approx_m^T / Id_A M$

Proposition 4.2 Let Σ be a system with the initial marking M_0 , and let σ be a sequence of transitions such that $M_0[\sigma > M$ and $M_0[\sigma > M'$ with $D \subseteq \psi(M)$, $C \subseteq \psi(M')$ and $f : D \rightarrow C$ a bijection, then

$$M \approx_m^T / f M' \implies makefree(\psi(M) - D, M) \approx_m^T / f makefree(\psi(M') - C, M')$$

Proposition 4.3 Let $f = Id_A$ such that $A \subseteq \psi(M)$. If $M \xrightarrow{E_1 t_x} M_1$ and $M \xrightarrow{E_2 t_y} M_2$ such that $A \cap E_1 \subseteq A \cap E_2$ then for $z = get(\mathcal{M} - ((\psi(M) - E_1) \cup (\psi(M) - E_2)))$ and $B = (A - E_2) \cup \{z\}$ we have $M_1[z/x] \approx_m^T / Id_B M_2[z/y]$.

Theorem 4.1 Let $\Sigma = (S, T, W)$ be a labeled system and let $mlts_1$ (resp. $mlts_2$) a maximality-based labeled transition system generated from Σ by using the maximality semantics (resp. maximality semantics with aggregation of transitions), then $mlts_1$ and $mlts_2$ are maximally bisimilar for the set of transitions T .

5. CASE STUDY

In order to illustrate the interest of the proposed approach, we study in this section an example of processes synchronization, namely ticket reservation system. We show the important reduction rate, and system properties preservation.

Let us consider a reservation system of plane tickets. To buy a ticket, We pass generally by two ticket-windows. A first window of type R (Reservation) allows reserving a place in a flight and to establish the plane ticket. A second window of type C (Cash desk) allows to take and collect the money and to give the ticket to the customer. This agency has a waiting room, three ticket-windows of type R and two of type C . While arriving, the customers pass initially in waiting room. As soon as a window of type R is free, a customer can present himself at the window and proceed to the reservation. Once this operation finished, he waits until a ticket-window of type C is free to carry out the payment and the withdrawal of his ticket.

FIGURE 9: Petri net specifying reservation system

System specification is given by Petri net of Figure 9. In order to show the reduction rate, we use at first the generating method of a maximality-based labeled transition system presented in Section 3, then we apply generation approach with reduction. This is achieved by varying the number of customers. Obtained results are summarized in Table 1.

6. CONCLUSION

In this paper, we proposed an operational method for generating reduced maximality-based labeled transition systems associated to Petri nets, based on aggregation of redundant transitions, and we showed that generated maximality-based labeled transition systems are equivalent to those generated by operational semantics proposed in [22] modulo the maximality bisimulation relation, i.e. degree of parallelism and system properties to be verified are preserved. As

Number of clients	Before reduction		After reduction		Reduction rate	
	States	Transitions	States	Transitions	States	Transitions
1	4	3	4	3	0%	0%
2	17	28	13	18	24%	36%
3	62	153	40	81	35%	47%
4	212	665	129	326	39%	51%
5	696	2572	423	1233	39%	52%
6	2202	9100	1376	4436	38%	51%
7	6658	29686	4303	14926	35%	50%
8	19135	89929	12801	47000	33%	48%

TABLE 1: Generation results

we have seen in the case study, the proposed approach reduces significantly the size of the maximality-based labeled transition system.

By considering maximality semantics, correctness properties relative to systems specified by a Petri net can be checked on maximality-based labeled transition systems which corresponds 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 graph.

In [3], an on-the-fly generation algorithm of maximality-based labeled transition systems, reduced modulo α -equivalence was proposed and integrated in FOCOVE¹ environment. We think that this algorithm adapts easily to the context of Petri nets.

REFERENCES

- [1] Aceto, L. and Hennessy, M. (1991), Adding action refinement to finite process algebra, in J. L. Albert, B. Monien and M. R. Artalejo, eds, 'ICALP'91', Vol. 510 of *LNCS*, Springer-Verlag, pp. 506–519.
- [2] Arnold, A. (1992), *Systèmes de transitions finis et sémantique des processus communicants*, Masson, Paris.
- [3] Benamira, A. and Saïdouni, D. E. (2006), Consideration of the covering steps in the maximality-based labeled transition systems, in 'Proceedings of International Arab Conference on Information Technology (ACIT'2006)', Al-Yarmouk University, Irbid, Jordan.
- [4] Best, E. and Devillers, R. (1987), 'Sequential and concurrent behaviour in Petri net theory', *Theoretical Computer Science* **55**, 87–136.
- [5] Best, E., Devillers, R., Kiehn, A. and Pomello, L. (1991), 'Concurrent bisimulations in Petri nets', *Acta Informatica* **28**, 231–264.
- [6] Boudol, G. and Castellani, I. (1988), 'Concurrency and atomicity', *TCS* **59**, 1–60.
- [7] Clarke, E. M., Emerson, E. A. and Sistla, A. P. (1986), 'Automatic verification of finite state concurrent systems using temporal logic specifications', *ACM Transactions on Programming Languages and Systems* **8**(2), 244–263.
- [8] Cleveland, R. and Hennessy, M. (1993), 'Testing equivalence as a bisimulation equivalence', *Formal Aspects of Computing* **5**, 1–20.
- [9] Courtiat, J. P. and Saïdouni, D. E. (1994), Action refinement in LOTOS, in A. Danthine, G. Leduc and P. Wolpe, eds, 'Protocol Specification, Testing and Verification (PSTV'93)', North-Holland, pp. 341–354.
- [10] Courtiat, J. P. and Saïdouni, D. E. (1995), Relating maximality-based semantics to action refinement in process algebras, in D. Hogrefe and S. Leue, eds, 'IFIP TC6/WG6.1, 7th Int. Conf. on Formal Description Techniques (FORTE'94)', Chapman & Hall, pp. 293–308.
- [11] Darondeau, P. and Degano, P. (1993), 'Refinement of actions in event structures and causal trees', *TCS* **118**, 21–48.
- [12] Degano, P. and Gorrieri, R. (1991), Atomic refinement in process description languages, in A. Tarlecki, ed., 'Mathematical Foundations of Computer Science', Vol. 520 of *LNCS*, Springer-Verlag, pp. 121–130.
- [13] Devillers, R. (1992a), Maximality preservation and the ST-idea for action refinement, in G. Rozenberg, ed., 'Advances in Petri Nets', Vol. 609 of *LNCS*, Springer-Verlag, pp. 108–151.
- [14] Devillers, R. (1992b), 'Maximality preserving bisimulation', *TCS* **102**, 165–183.

¹<http://www.focove.new.fr>

- [15] Dijkstra, E. W. (1971), 'Hierarchical ordering of sequential processes', *Acta Informatica* **1**(2), 115–138.
- [16] Emerson, E. A. (1990), Temporal and modal logic, in J. van Leeuwen, ed., 'Handbook of Theoretical Computer Science : Formal Models and Semantics, Ch. 16', Vol. B, Elsevier, pp. 995–1072.
- [17] Fernandez, J. C. and Mounier, L. (n.d.), A tool set for deciding behavioral equivalences, pp. 23–42.
- [18] Janssen, W., Poel, M. and Zwiers, J. (1991), Action systems and action refinement in the development of parallel systems, in J. C. M. Baeten and J. F. Groote, eds, 'CONCUR'91', Vol. 527 of *LNCS*, Springer-Verlag, pp. 298–316.
- [19] Park, D. (1981), Concurrency and automata on infinite sequences, in 'GI', Vol. 104 of *LNCS*, Springer-Verlag, pp. 167–183.
- [20] Saïdouni, D. E. (1996), Sémantique de maximalité: Application au raffinement d'actions en LOTOS, PhD thesis, LAAS-CNRS, 7 av. du Colonel Roche, 31077 Toulouse Cedex France.
- [21] Saïdouni, D. E. and Belala, N. (2005), 'Using maximality-based labeled transition system model for concurrency logic verification', *The International Arab Journal of Information Technology (IAJIT)* **2**(3), 199–205. ISSN: 1683-3198.
- [22] Saïdouni, D. E., Belala, N. and Bouneb, M. (2008), Using Maximality-based Labeled Transition System as a Model for Petri Nets, Technical report, Submitted for publication, Computer Science Dept., University of Mentouri, 25000 Constantine, Algeria.
- [23] Saïdouni, D. E., Belala, N. and Bouneb, M. (2008), Aggregation of transitions in marking graph generation based on maximality semantics for Petri nets (extended version), Technical report, Computer Science Dept., University of Mentouri, 25000 Constantine, Algeria.
- [24] Saïdouni, D. E. and Courtiat, J. P. (1994), Syntactic action refinement in presence of multiway synchronization, in D. Andrews, J. Groote and C. Middelburg, eds, 'Workshop on Semantics of Specification Languages (SoSL'93)', Workshops in Computing, Springer-Verlag, Utrecht, pp. 289–303.
- [25] Saïdouni, D. E. and Courtiat, J. P. (2003), Prise en compte des durées d'action dans les algèbres de processus par l'utilisation de la sémantique de maximalité, in 'CFIP'2003', Hermes, France.
- [26] Saïdouni, D. E. and Labbani, O. (2003), Maximality-based symbolic model checking, in 'ACS/IEEE International Conference on Computer Systems and Applications', Tunisia.
- [27] van Glabbeek, R. J. (1990), The refinement theorem for ST-bisimulation semantics, in 'IFIP Working Conference on Programming Concepts and Methods', North-Holland.
- [28] van Glabbeek, R. J. and Vaandrager, F. W. (1987), Petri net models for algebraic theories of concurrency, in 'PARLE'87, Vol. II (Parallel Languages)', Vol. 259 of *LNCS*, Springer-Verlag, pp. 224–242.
- [29] Vogler, W. (1993), 'Bisimulation and action refinement', *TCS* **114**, 173–200.