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).
In this paper, we propose an operational semantics for generating maximality-based labeled transition systems. This of Figure 1.(a)?

A transition between two states (under execution) which occurred before this state. This action is said to be therefore, is it possible to omit the derivations produced by the firing of t\textsubscript{1} in state s\textsubscript{5}. The set of event names corresponding to state s\textsubscript{5} is that of s\textsubscript{4} from which we substract 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\textsubscript{4}) 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\textsubscript{4}, 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\textsubscript{2} (resp. s\textsubscript{3}) stipulates that the action a (resp. b) are potentially under execution in this state. {x, y} in s\textsubscript{4} shows that actions a and b can be executed simultaneously.

Note that when the system is in state s\textsubscript{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\textsubscript{4} and s\textsubscript{5}. We can observe that from state s\textsubscript{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\textsubscript{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\textsubscript{1} during the firing of t\textsubscript{2} in the Petri net.

Notice that from state s\textsubscript{2}, transitions leading respectively to states s\textsubscript{4} and s\textsubscript{5} are due to the firing of the same transition t\textsubscript{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\textsubscript{1}. On the other hand, such as we noted above, the derivation by b leading to state s\textsubscript{4} is not conditioned by the end of the action a, while the derivation leading to state s\textsubscript{5} is conditioned by the end of a. Therefore, is it possible to omit the derivations s\textsubscript{2} \rightarrow s\textsubscript{4} \rightarrow s\textsubscript{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].

**FIGURE 1:** Operational semantics of a Petri net in terms of MLTS
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 \(W : ((S \times T) \cup (T \times S)) \rightarrow \mathbb{N} = \{0, 1, 2, \ldots\}\) is the weight function. Graphically, transitions of \(T\) are represented by rectangles, places of \(S\) by circles and weight function by arrows associated with their weights. We suppose that all nets are finite, i.e., \(|S \cup T| \in \mathbb{N}\).

- For \(x \in S \cup T\), the pre-set \(\cdot x\) is defined by \(\cdot x = \{y \in S \cup T | W(y, x) \neq 0\}\) 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 \(M : S \rightarrow \mathbb{N}\). 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 \(A\) is a finite set; we suppose that \(\tau \notin A\) (\(\tau\) will indicate invisible action, or silent action).

- The labeling of a Petri net \(N = (S, T, W)\) is a function \(\lambda : T \rightarrow A \cup \{\tau\}\). If \(\lambda(t) \in A\) then \(t\) is said to be observable or external; at the opposite, \(t\) is silent or internal.

- A sequence \(\sigma = (S, T, W, M_0, \lambda)\) is a labeled system iff \((S, T, W, M_0)\) is a marked Petri net and \(\lambda\) is a labeling function of \((S, T, W)\).

- An action \(a \in A\) of a system \(\Sigma = (S, T, W, M_0, \lambda)\) is auto-concurrent in a marking \(M\) iff \(M\) concurrently enables two observable transitions \(t_1\) and \(t_2\) (not necessarily distinct) such that \(\lambda(t_1) = \lambda(t_2) = a\).

- A sequence \(\sigma = M_0t_1M_1t_2\ldots\) is an occurrence sequence iff \(M_{i-1}[t_i]M_i\) for \(1 \leq i \leq n\). A sequence \(t_1t_2\ldots\) is a transition sequence starting with \(M\) iff there is an occurrence sequence \(M_1M_2\ldots\) If a finite sequence \(t_1t_2\ldots t_n\) leads from \(M\) to \(M'\), we write \(M[t_1t_2\ldots t_n]M'\). The set of reachable markings of a marked Petri net \((S, T, W, M_0)\) is defined as \(\{M_0\} = \{M|\exists t_1t_2\ldots t_n : M_0[t_1t_2\ldots t_n]M\}\)

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

**Definition 2.1** Let \(M\) be a countable set of event names, a maximality-based labeled transition system of support \(M\) is a tuple \((\Omega, \mu, \xi, \psi)\) with:

- \(\Omega = (S, T, \alpha, \beta, \varsigma_0)\) 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.
  - \(\alpha\) and \(\beta\) are two applications of \(T\) in \(S\) such that for all transition \(t\) we have: \(\alpha(t)\) is the origin of the transition and \(\beta(t)\) its goal.
  - \(\varsigma_0\) is the initial state of the transition system \(\Omega\).

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

- \(\psi : S \rightarrow 2^M\) is a function which associates to each state the finite set of maximal event names present in this state.

- \(\mu : T \rightarrow 2^M\) 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.

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

such that \(\psi(\varsigma_0) = \emptyset\) 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 \(mlts = (\Omega, \mu, \xi, \psi)\) a maximality-based labeled transition system such that \(\Omega = (S, T, \alpha, \beta, \varsigma_0), t \in T\) 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 \(s \xrightarrow{\alpha, \beta, \lambda, \mu, \xi, E} s'\).

- 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 \(F\)). Restrictions of \(f\) with respect to its domain and codomain are defined by:
  - \(f[D] = \{(x, y) \in f|x \in D\}\)
  - \(f[C] = \{(x, y) \in f|y \in C\}\)
• $\mathcal{F} \subseteq 2^{M \times M}$ is the set of all bijective functions between subsets of $M$.

• $Id_{\mathcal{A}}$ is the identity function on elements of a set $\mathcal{A}$.

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 $\Omega_1 = (S_1, T_1, \alpha_1, \beta_1, s_1) \omega_0$ and $\Omega_2 = (S_2, T_2, \alpha_2, \beta_2, s_2) \omega_0$. $mlts_1$ and $mlts_2$ are said to be maximally bisimilar, noted $mlts_1 \approx_m mlts_2$, if there is a relation $R \subseteq S_1 \times S_2 \times \mathcal{F}$ with

1. $(s_1, s_2, \emptyset) \in R$. 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 $(s_1, s_2, f) \in R$ then
   a. $\text{Dom}(f) \subseteq \psi(s_1)$ and $\text{Cod}(f) \subseteq \psi(s_2)$.
   b. If $s_1 \xrightarrow{\alpha_{s_1}} s_1'$ then there is $s_2 \xrightarrow{\alpha_{s_2}} s_2'$ such that
      i. $\forall (u, v) \in f$, if $u \notin E$ then $v \notin E$
      ii. $(s_1', s_2', f') \in R$ with $f' = (f[\psi(s_1') - \{x\}] - \{y\}) \cup \{(x, y)\}$
   c. If $s_2 \xrightarrow{\alpha_{s_2}} s_2'$ then there is $s_1 \xrightarrow{\alpha_{s_1}} s_1'$ such that
      i. $\forall (u, v) \in f$, if $v \notin E$ then $u \notin E$
      ii. $(s_1', s_2', f') \in R$ with $f' = (f[\psi(s_1') - \{x\}] - \{y\}) \cup \{(x, y)\}$

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 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 $FT$, while bound tokens set will be denoted by $BT$. 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).
Aggregation of Transitions in Marking Graph Generation for Petri Nets

FIGURE 3: Free tokens and bound tokens in a marking

(a)  
(b)  
(c)

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 $BT = \{t_1_x\}$ while the set of bound tokens in $s_2$ of the configuration $C_3$ is $BT = \{t_1_x, t_1_y\}$. 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 $BT$ of the place $s_2$ contains two tokens bound to the firing $t_{1x}$, i.e. $BT = \{t_{1x}, t_{1y}\}$. Since $BT$ is a set, we consider that a bound token is a tuple $(n, t, x)$ of $\mathbb{N} \times T \times M$, 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
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.

**Definition 3.2** Let \( N = (S, T, W) \) be a Petri net, the marking of \( N \) is a function \( M : S \rightarrow \mathbb{N} \times 2^{N \times T \times M} \). Among others, the marking \( M(s) \) of a place \( s \in S \) is a pair \((FT, BT)\) such that \( FT \in \mathbb{N} \) and \( BT \in 2^{N \times T \times M} \) 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 \( M(s) = (FT, BT) \) such that \( BT = \{n_1 t_{1x_1}, \ldots, n_m t_{xm} \} \) then \(|M(s)| = |FT| + |BT| \) with \(|BT| = \sum_{i=1}^{m} n_i \) 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 7.(a).

We assume that one token of \( s_3 \) is bound to the firing of \( t_1 \left(t_{1x_1}\right) \) and the other one is bound to the firing of \( t_2 \left(t_{2x_2}\right) \). 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.

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 3.3** 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 \( \psi \) will be used to calculate this set, it can be defined as \( \psi(M) = \bigcup_{s \in S} \bigcup_{i=1}^{m} s x_i \) such that \( M(s) = (FT, BT) \) with \( BT = \{(n_{s_1}, t_{s_1}, x_{s_1}), \ldots, (n_{s_m}, t_{s_m}, x_{s_m})\} \).
- Let \( N \subset M \) be a non-empty finite set of event names, \( \text{makefree}(N, M) \) is defined recursively by:
  - \( \text{makefree}(\{x_1, x_2, \ldots, x_n\}, M) = \text{makefree}(\{x_2, \ldots, x_n\}, \text{makefree}(\{x_1\}, M)) \)
  - \( \text{makefree}(\{x\}, M) = M' \) such that for all \( s \in S \), if \( M(s) = (FT, BT) \) then
3.2. Construction of marking graph

Let $\Sigma = (S, T, W, M_0, \lambda)$ be a labeled system. The marking graph $M_g$ labeled by $\lambda$ associated to $\Sigma$ is a graph in which the states are defined by all reachable markings from the initial marking $M_0$ and the transitions between states are labeled according to the derivation rule of Definition 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: $M \xrightarrow{E \cdot t} M'$ (also denoted by $(M, E \cdot t, M')$) 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 $t$.
- $x = \text{get}(M - \psi(M'''))$ and
- $M' = \text{occurrence}(t, x, M''')$.

3.3. Properties

Proposition 3.1 Let $\Sigma = (S, T, W, M_0, \lambda)$ be a labeled system and $M_g$ its marking graph built according to Definition 3.3, then the structure $\Sigma_{\text{mbs}} = (M_g, \alpha, \beta, M_0)$ is a maximality-based labeled transition system with:

- $M_g = (S_g, T_g, \alpha, \beta, M_0)$ is the marking graph associated to $\Sigma$ such that
  - $S_g$ is the set of states defined by the set of reachable markings from the initial marking $M_0$.
  - $T_g = \{ (M, E \cdot t, M') \}$ such that $M, M' \in S_g$ and $(M, E \cdot t, M')$ is a valid derivation.
  - For $(M, E \cdot t, M') \in T_g$ we have $\alpha((M, E \cdot t, M')) = M$ and $\beta((M, E \cdot t, M')) = M'$.
  - $\psi : S_g \rightarrow 2^M$ is the function defined in Definition 3.2.
  - For $d = (M, E \cdot t, M') \in T_g$ 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 $d = (M, E \cdot t, M') \in T_g$, 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.
In this section, we present a maximality bisimulation relation according to Petri net transitions instead of actions.

### 4.2. Maximality bisimulation relation on transitions

**Definition 4.1**

Let $\Sigma_1 = (S_1, T_1, W_1, M_{10}, \lambda_1)$ and $\Sigma_2 = (S_2, T_2, W_2, M_{20}, \lambda_2)$ be two labeled systems. $\Sigma_1$ and $\Sigma_2$ 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 $S_{sys1}$ and $S_{sys2}$ 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.

![Maximally bisimilar systems](image)

**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 $M(s) = (FT, BT)$, $M'(s)$ is defined as follows:

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

with $BT' \subseteq BT$ and $|BT'| = w(s, t) - FT$

#### 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 $\mathcal{R} \subseteq 2^{Mark} \times 2^{Mark} \times \mathfrak{H}$ is said a maximality bisimulation relation according to a set of transitions $T$ iff for any $(M, M', A_i, A_i') \in \mathcal{R}$, $A_i \subseteq \psi(M')$ and $A_i' \subseteq \psi(M')$,

  1. If $M, M' \xrightarrow{E_i} M_j$, then $\exists M_{\psi(M)} \xrightarrow{E_i} M_j$ and for $z = \psi(M) - ((\psi(M_i) - E_i) \cup E_i')$.

  2. If $M, M' \xrightarrow{E_i} M_j$, then $\exists M_{\psi(M)} \xrightarrow{E_i} M_j$ for $z = \psi(M) - ((\psi(M_i) - E_i) \cup E_i')$.

- Two labeled systems $\Sigma_1 = (S_1, T, W_1, M_{10}, \lambda_1)$ and $\Sigma_2 = (S_2, T, W_2, M_{20}, \lambda_2)$ are said to be maximally bisimilar according to $T$, noted $\Sigma_1 \approx_T \Sigma_2$, iff there exists a maximality bisimulation relation $\mathcal{R}$ according to $T$ such that $(M_i, M_i', \emptyset) \in \mathcal{R}$.

---

*International Workshop on Verification and Evaluation of Computer and Communication Systems*  
*VECoS 2008*
M_1 \approx_{m}^{T} / f M_2$ denotes the fact that $(M_1, M_2, f) \in R$.

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 $\text{makefree}(B, M) \approx_{m}^{T} / Id_{A} M$

Proposition 4.2 Let $\Sigma$ be a system with the initial marking $M_0$, and let $\sigma$ 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

$M \approx_{m}^{T} / f M' \implies \text{makefree}(\psi(M) - D, M) \approx_{m}^{T} / f \text{makefree}(\psi(M') - C, M')$

Proposition 4.3 Let $f = Id_A$ such that $A \subseteq \psi(M)$. If $M \frac{E_1}{m} \frac{E_2}{m} M_1$ and $M \frac{E_1}{m} \frac{E_2}{m} M_2$ such that $A \cap E_1 \subseteq A \cap E_2$ then for $z = \text{get}(M - ((\psi(M) - E_1) \cup (\psi(M) - E_2)))$ 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 $\text{mlts}_1$ (resp. $\text{mlts}_2$) a maximality-based labeled transition system generated from $\Sigma$ by using the maximality semantics (resp. maximality semantics with aggregation of transitions), then $\text{mlts}_1$ and $\text{mlts}_2$ 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.

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
Aggregation of Transitions in Marking Graph Generation for Petri Nets

<table>
<thead>
<tr>
<th>Number of clients</th>
<th>Before reduction</th>
<th>After reduction</th>
<th>Reduction rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>States</td>
<td>Transitions</td>
<td>States</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>17</td>
<td>28</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>62</td>
<td>153</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>212</td>
<td>665</td>
<td>129</td>
</tr>
<tr>
<td>5</td>
<td>696</td>
<td>2572</td>
<td>423</td>
</tr>
<tr>
<td>6</td>
<td>2202</td>
<td>9100</td>
<td>1376</td>
</tr>
<tr>
<td>7</td>
<td>6658</td>
<td>29686</td>
<td>4303</td>
</tr>
<tr>
<td>8</td>
<td>19135</td>
<td>89929</td>
<td>12801</td>
</tr>
</tbody>
</table>

**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 \(\alpha\)-equivalence was proposed and integrated in FOCOVE\(^1\) environment. We think that this algorithm adapts easily to the context of Petri nets.

REFERENCES


\(^1\)http://www.focove.new.fr


