DOI 10.1007/s00165-015-0341-3
BCS © 2015
Formal Aspects of Computing

Formal Aspects
of Computing

A reduced maximality labeled transition system generation for recursive Petri nets

Messaouda Bouneb[1,2], Djamel Eddine Saidouni[2] and Jean Michel Ilie[3]

[1] Department of Mathematics and Computer Sciences, El Arbi ben Mhidi University, Oum el bouaghi, Algeria
[2] MISC Laboratory, Department of Fundamental Informatics and its Applications, University Constantine 2, Abdelhamid Mehri, Constantine, Algeria
[3] Department of Computer Sciences, Pierre and Marie Curie University, Paris, France

Abstract. In Saidouni et al. (Maximality semantic for recursive Petri net. Europeen conference on modelling and simulation (ECMS'13) pp 544–550, 2013) a maximality operational semantics has been defined for the recursive Petri net model. This operational semantics generates a true concurrency structure named maximality-based labeled transition systems (MLTS). This paper proposes an approach that generates an on-the-fly reduced MLTS modulo a maximality bisimulation relation. The interest of the approach is shown using an example concerning the woodshop cutting system.

Keywords: Maximality labeled transition systems, Maximality bisimulation, Recursive Petri nets

1. Introduction

Formal methods are techniques for rigorously reasoning about systems using a mathematical basis, in order to demonstrate their validity relatively to certain specifications. These methods provide an assurance of the absence of bugs in systems. Their improvement has motivated many researchers in computer science. For this purpose, several formal methods have been proposed to specify, analyze, verify and synthesize concurrent systems. A concurrent system can be seen as a set of cooperative or competitive processes. Managing these systems behavior is provided by various tools implementing synchronous communication, mutual exclusion, etc. It is obvious that concurrent systems cannot be described by only referring to their initial and final states. Adequate description should represent several kinds of behavior such as possible states of the system, causally dependent tasks, parallelism, synchronization and resource sharing. Formal methods are based on specification and semantics models able to describe concurrent systems. Specification models are used by specifiers for describing their systems; while semantics models interpret the specifications and are used for checking systems properties. One of the most elegant models for specifying concurrent systems is the Petri net model. This model is based on a graphical representation, which facilitates its use.

Correspondence and offprint requests to: M. Bouneb, e-mail: bounebm.univ@gmail.com

One approach for checking Petri net specifications consists of generating its marking graph in which nodes represent system states, and arcs represent transitions moving the system from one state to another. After its generation, the marking graph can be seen as a labeled transition system [Arn92]. The generated labeled transition system is then used for verifying system properties (model checking, bisimilarity, conformance testing, etc) [Ces86, Clh93]. 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. 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 [Ach91, Bdk91, Boc88, Cos94, Cos95, Dad89, Dad91, Dad93, Deg91, Dev92a, Dev92b, Dev93, Dij71, Jpz91, Sai96, Van90]. Two main reasons argue the use of these semantics; the first is that they allow the hierarchical system design by the refinement of actions; the second is that they allow the characterization of parallel execution of non-atomic actions.

Among these semantics, we can cite the maximality semantics which has been defined independently for Petri net and event structure models by Devillers and Vogler [Dev92a, Dev92b, vog93]. In this context, maximality bisimulation relation has been defined and shown to be the coarsest relation preserved by action refinement. It has also been shown that this relation is a congruence with the performance relation [Sac03]. In underlying semantics 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 from a theoretical point of view.

Dealing with implementability, another model, named maximality-based labeled transition system (MLTS) has been defined in the literature and firstly used for expressing the semantics of process algebras with the hypothesis that actions are not necessarily atomic, i.e. actions are abstractions of finite processes and may elapse in time [Sac94, Sai96]. The main interest of the maximality-based labeled transition system model is that it can be implemented and used in verification. Later on, these results have been extended to the Petri net model by defining a maximality operational semantics of the model in terms of MLTS [Sbb08a, Sbb09a, Sbb09b].

At first, we recall that a maximality-based labeled transition system is a graph labeled on both states and transitions (Fig. 1). Each state is labeled by a set of event names. Each event name identifies the start of execution of an action which occur before this state. This action is said potentially under execution in this state. A transition between two states s_i and s_j is labeled by a 3-uple (E,a,x) (noted ${}_{E}a_{x}$) where x is the event name identifying the start of execution of the action a and E identifies the set of event names representing the causes of the action a. Elements of E belong to the state s_i. The occurrence of this transition terminates the actions identified by E. Thus, the set of event names corresponding to the state s_j is that of s_i from which the set E is subtracted and the event name x is added. It should be noted that the set of event names associated to a state refer to actions potentially in execution at this state. Each event is maximal according to the causality relation between events. Thus, this set will be named a set of maximal event names. The formal definition of a maximality-based labeled transition system is given in Sect. 2 below.

As an example, let us consider the Petri net of Fig. 2a. By applying the approach of Saidouni, Belala and Bouneb [Sbb09a], the corresponding maximality-based labeled transition system of this Petri net is given by Fig. 2b.

In the initial state (state s_0) of the maximality-based labeled transition system of Fig. 2b, no action is running, from where the association of the empty set to this state. From the state s_0, 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 the state s_1 (resp. s_2) stipulates that the action a (resp. b) is potentially under execution in this state. The set $\{x, y\}$ in s_3 shows that actions a and b may be in execution simultaneously.

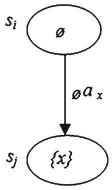
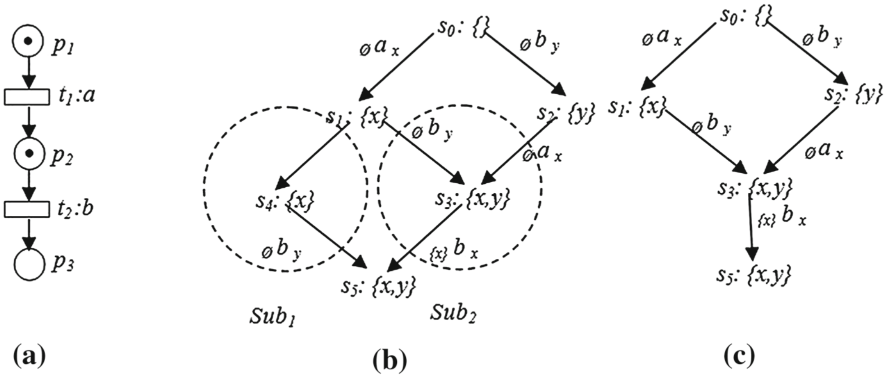


Fig. 1. Example of MLTS



(a) (b) (c)

Fig. 2. Petri net and its corresponding MLTS

It is worth to note that when the system is in the state s_1, while the action a has not been yet terminated, the only evolution concerns the start of b. However, when a terminates, the action b caused by a or the action b which is independent from the end of a can be started. Resulting states are s_3 and s_4 respectively. It can be observed that from the state s_4, 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 the state s_3. We note that causal dependence between the 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.

It is worth to note that from the state s_1 transitions leading respectively to states s_3 and s_4 are due to 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, as we noted above, the derivation by b leading to the state s_3 is not conditioned by the end of the action a, while the derivation leading to the state s_4 is conditioned by the end of a. As explained in [Sbb08b], it is possible to omit the derivations $s_1 \longrightarrow s_4 \longrightarrow s_5$ in the maximality-based labeled transition system. In other words, the maximality-based labeled transition system of Fig. 2c preserves the behavior of the Petri net of Fig. 2a; the degree of parallelism is preserved.

In [Sbb08b], the authors have proposed an operational semantics for generating maximality-based labeled transition systems for the Petri net model. This semantics performs aggregation of derivations according to the idea explained in the previous example. It has been shown that the generated maximality-based labeled transition system is equivalent to the one generated by the operational semantics proposed in [Sbb09a] modulo the maximality bisimulation relation.

The limits of the Petri net model has been highlighted for the specification of systems with dynamic structures, such as multi-agent systems. For this reason, recursive Petri nets have been defined [Sep99a, Sep99b]. Dynamic behaviors are considered through abstract transitions. In fact the firing of such transition represents the execution of a thread. The thread behavior is modeled by a recursive Petri net. Abstract transitions can be used for hierarchical design of dynamic systems. As abstract transitions represent activities, the association of a true concurrency semantics to the model becomes more appropriate than the use of interleaving semantics. For this purpose, in [Sbi13], a maximality operational semantics has been proposed for recursive Petri nets.

In this paper, we extend the maximality bisimulation relation to recursive Petri nets and propose an operational semantics for generating a reduced maximality-based labeled transition systems. This semantics performs aggregation of derivations according to the maximality bisimulation relation.

The paper is organized as follows: Sects. 2 and 3 recall the definition of maximality-based labeled transition systems and the maximality semantics of Petri net model. Sections 4 and 5 recall the definition and the maximality semantics of recursive Petri net model. In Sect. 6, we define an on-the-fly reduction method for recursive Petri nets which preserves the maximality bisimulation relation. In Sect. 7, through an example, we show that the proposed approach significantly reduces the size of the maximality-based labeled transition systems. Finally, Sect. 8 concludes the paper.

2. Maximality-based labeled transition system

Definition 2.1 Let $\mathcal{H}$ be a countable set of event names. Let $\mathbb{L}$ be an alphabet ranging over by $a, b, \ldots$. In practice a label is a name of an action. A maximality-based labeled transition system of support $\mathcal{H}$ is a fivefold $(\rho, \varphi, \mu, \xi, \theta)$ with: $\rho = \langle S, TR, \alpha, \beta, s_0 \rangle$ is a transition system such that:

- S is the set of states in which the system may be found, this set can be finite or infinite.
- TR is the set of transitions indicating the change of states which the system can do; this set can be finite or infinite.
- α and β are two applications of TR in S such that for any transition $tr \in TR$: $\alpha(tr)$ is the origin of the transition *tr* and $\beta(tr)$ is its goal.
- s_0 is the initial state of the transition system ρ.
- (ρ, φ) is a system of transitions labeled by the function φ on $\mathbb{L}$, called support of (ρ, φ). $(\varphi : TR \longrightarrow \mathbb{L})$.
- $\theta : S \longrightarrow 2^{\mathcal{H}}$ is a function which associates to each state a finite set of maximal event names, with the assumption that $\theta(s_0) = \emptyset$.[1]
- $\mu : TR \longrightarrow 2^{\mathcal{H}}$ is a function which associates to each transition a finite set of event names corresponding to the actions which began their execution and their terminations cause the execution of this transition.
- $\xi : TR \longrightarrow \mathcal{H}$ is a function which associates to each transition the event name identifying its occurrence.

Where each transition $tr \in TR$ satisfies the condition, $\mu(tr) \subseteq \theta(\alpha(tr))$, $\xi(tr) \notin \theta(\alpha(tr)) - \mu(tr)$ and $\theta(\beta(tr)) = (\theta(\alpha(tr)) - \mu(tr)) \cup \{\xi(tr)\}$.

The last condition avoids the consideration of imaginary systems. In fact:

- The condition $\mu(tr) \subseteq \theta(\alpha(tr))$ ensures that the execution of the transition tr is only conditioned by the termination of a subset of actions potentially in execution in the state $\alpha(tr)$.
- The condition $\xi(tr) \notin \theta(\alpha(tr)) - \mu(tr)$ ensures that the event name $\xi(tr)$ indexing the transition tr does not refer to any action remaining potentially in execution in the resulting state $\beta(tr)$.
- As the set of event names $\mu(tr)$ is related to actions such that their termination constitute a condition for the execution of the transition tr, then the condition $\theta(\beta(tr)) = (\theta(\alpha(tr)) - \mu(tr)) \cup \{\xi(tr)\}$ ensures that the set of maximal events in the state $\beta(tr)$ is the one in the state $\alpha(tr)$ from which the set $\mu(tr)$ is removed and the event name $\xi(tr)$ is added.

---

[1] $2^{\mathcal{H}}$ denotes the part sets of $\mathcal{H}$.

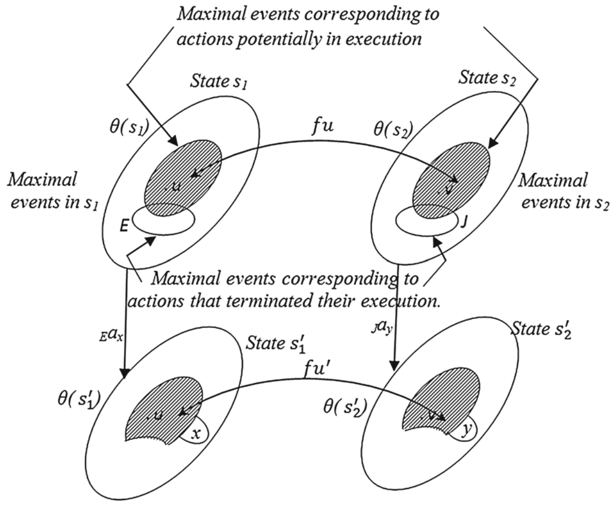


Fig. 3. Maximality bisimulation relation

Notation 2.1 In the sequel the following notations will be used:

- Let $mlts = (\rho, \varphi, \mu, \xi, \theta)$ be a maximality-based labeled transition system such that: $\rho = \langle S, TR, \alpha, \beta, s_0 \rangle$. $tr \in TR$ is a transition such that $\alpha(tr) = s$, $\beta(tr) = s'$, $\varphi(tr) = a$, $\mu(tr) = E$ and $\xi(tr) = x$. The transition tr will be noted $s \xrightarrow{{}_E a_x} s'$.
- Let $fu : E \longrightarrow J$ be a function of domain $Dom(fu) = E$ and codomain $Cod(fu) = J$, and let D (respectively C) be a subset of E (respectively of J). Restrictions of fu with respect to its domain and codomain are defined respectively by:
  - $fu\lceil D = \{(x, y) \in fu \mid x \in D\}$.
  - $fu\lfloor C = \{(x, y) \in fu \mid y \in C\}$.
- $\mathfrak{F}$ is the set of all bijective functions between subsets of $\mathcal{H}$.
- Id_A is the identity function on elements of the set A.

Definition 2.2 Let $mlts_1 = (\rho_1, \varphi_1, \mu_1, \xi_1, \theta_1)$ and $mlts_2 = (\rho_2, \varphi_2, \mu_2, \xi_2, \theta_2)$ be two maximality-based labeled transition systems such that $\rho_1 = \langle S_1, TR_1, \alpha_1, \beta_1, s1_0 \rangle$ and $\rho_2 = \langle S_2, TR_2, \alpha_2, \beta_2, s2_0 \rangle$. $mlts_1$ and $mlts_2$ are said to be maximally bisimilar, noted $mlts_1 \approx_m mlts_2$, if there is a relation $\mathfrak{Rel} \subseteq S_1 \times S_2 \times \mathfrak{F}$ with (see Fig. 3):

1. $(s1_0, s2_0, \emptyset) \in \mathfrak{Rel}$. 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 too.
2. If $(s_1, s_2, fu) \in \mathfrak{Rel}$ then:
   (a) $fu : C \longrightarrow D$ where $C \subseteq \theta(s_1)$ and $D \subseteq \theta(s_2)$.
   (b) If $s_1 \xrightarrow{{}_E a_x} s_1'$ then there is $s_2 \xrightarrow{{}_J a_y} s_2'$ such that:
      - $\forall (u, v) \in fu$, if $u \notin E$ then $v \notin J$
      - $(s_1', s_2', fu') \in \mathfrak{Rel}$ with $fu' = (fu\lceil(C - E)\lfloor(D - J)) \cup \{(x, y)\}$
   (c) If $s_2 \xrightarrow{{}_J a_y} s_2'$ then there is $s_1 \xrightarrow{{}_E a_x} s_1'$ such that:
      - $\forall (u, v) \in fu$, if $v \notin J$ then $u \notin E$
      - $(s_1', s_2', fu') \in \mathfrak{Rel}$ with $fu' = (fu\lceil(C - E)\lfloor(D - J)) \cup \{(x, y)\}$

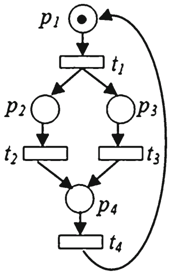


Fig. 4. Marked Petri net

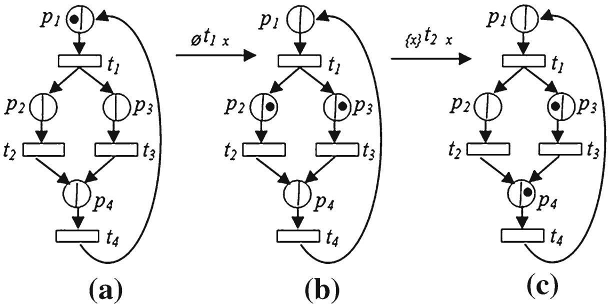


Fig. 5. Free and bound tokens in a marking

3. Maximality semantics for Petri nets

In this section we recall the maximality approach of place transition Petri nets, proposed in [Sbb08a, Sbb09a]. We introduce through a simple example useful notations and functions for the definition of a marking graph associated to a Petri net in a maximality-based approach.

Let us consider the example of the marked Petri net of Fig. 4. 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 the tokens produced by the firing of the transition t_1 are bound to this transition, namely the token in the place p_2 and that in the place p_3 (Fig. 5b). We can see that, in the initial state, the token in p_1 is not bound to any transition; this token is called free in this state, which explains the marked Petri net of Fig. 5a. 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 p_3 will become free. Resulting marking after the firing of the transition t_2 is given in Fig. 5c.

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 contains bound tokens. In a place, the number of free tokens will be denoted by $\mathcal{FT}$, while the bound token set will be noted $\mathcal{BT}$. So, each place is marked by $(\mathcal{FT}, \mathcal{BT})$. Hence, we obtain the succession of markings of Fig. 5. Each bound token identifies an action that is eventually being executed (this token corresponds to a maximal event). Also, each transition of the marking graph corresponds to the start of execution of an action which is identified by an event name. Since a weight of an edge linking a transition to a place may be greater than one, a firing transition may produce more than one bound token, the bound token is identified by a tuple (n, t, x) where n is the number of instances of a bound token, t is a fired transition producing this bound token and x is an event name identifying the transition firing in time. It is worth to note that the firing condition of a transition is only conditioned by the number of free and bound tokens in places.

Definition 3.1 A Petri net is a 4-tuple $\left(P, T, W^-, W^+\right)$ where:

- P : is a finite set of places.
- T : is a finite set of transitions such that: $P \cap T = \emptyset$.
- $W^- : P \times T \longrightarrow \mathbb{N}$ is the matrix of preconditions.
- $W^+ : P \times T \longrightarrow \mathbb{N}$ is the matrix of postconditions.

Definition 3.2 A labeled Petri net $\Sigma = \left(P, T, W^-, W^+, \lambda\right)$ is a Petri net in which all transitions are labeled by actions such that $\lambda : T \longrightarrow \mathbb{L}$ is a labeling function.

3.1. Operational maximality semantics for Petri nets

Preliminary definitions

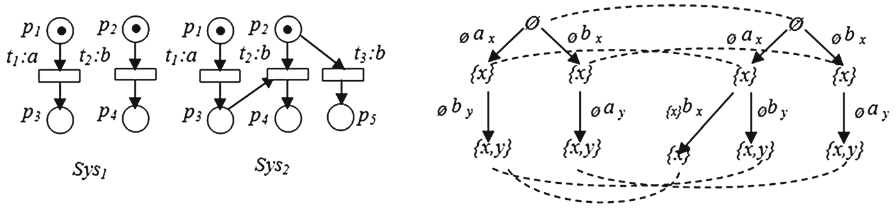
Let $\left(P, T, W^-, W^+\right)$ be a Petri net with a marking M:

- $\forall p \in P$, $M(p)$ is a pair $(\mathcal{FT}, \mathcal{BT})$ such that $\mathcal{FT} \in \mathbb{N}$ and $\mathcal{BT} = \{bt/bt \in \mathbb{N} \times T \times \mathcal{H}\}$ denote the number of free tokens and the set (possibly empty) of bound tokens in the place p, respectively.
- Let p be a place such that $M(p) = (\mathcal{FT}, \mathcal{BT})$ where $\mathcal{BT} = \{(n_1, t_1, x_1), \ldots, (n_m, t_m, x_m)\}$. The set of event names in p is given by a function $\delta^\bullet : P \longrightarrow 2^{\mathcal{H}}$, $\delta^\bullet(p) = \{x_1, x_2, \ldots, x_m\}$.
- 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 δ will be used to calculate this set and it can be defined as:
  $\delta : \{M : M \textit{ a marking of the Petri net}\} \longrightarrow 2^{\mathcal{H}}$ such that $\delta(M) = \cup_{p \in P}\, \delta^\bullet(p)$.
- Let $X \subset \mathcal{H}$ be a finite set of maximal event names of actions which terminated their execution. The operation of transforming bound tokens defined by X to free tokens in the marking M is defined by the inductive function *makefree* as follows:
  - $makefree\,(\{x_1, x_2, \ldots, x_m\}, M) = makefree\,(\{x_2, \ldots, x_m\}, makefree\,(\{x_1\}, M))$
  - $makefree\,(\{x\}, M) = M'$ such that for all $p \in P$, if $M(p) = (\mathcal{FT}, \mathcal{BT})$ then:
    - · If there is $(n, t, x) \in \mathcal{BT}$ then $M'(p) = (\mathcal{FT} + n, \mathcal{BT} - \{(n, t, x)\})$ (Conversion of n bound tokens identified by the event name x to free tokens).
    - · Otherwise, $M'(p) = M(p)$.
- $\mid M(p) \mid = \mathcal{FT} + \sum_{i=1}^{m} n_i$ such that $M(p) = (\mathcal{FT}, \mathcal{BT})$ with $\mathcal{BT} = \{(n_1, t_1, x_1), \ldots, (n_m, t_m, x_m)\}$.
- Let t be a transition of T; t is said to be enabled by the marking M iff $\mid M(p) \mid \geq W^-(p, t)$ for all $p \in P$. The set of all transitions enabled by the marking M will be noted $enabled(M)$.
- The marking M is said minimal for the firing of the transition t iff $\mid M(p) \mid = W^-(p, t)$ for all $p \in P$.
- Let M_1 and M_2 be two markings of the Petri net $\left(P, T, W^-, W^+\right)$. $M_1 \subseteq M_2$ iff $\forall p \in P$, if $M_1(p) = (\mathcal{FT}_1, \mathcal{BT}_1)$ and $M_2(p) = (\mathcal{FT}_2, \mathcal{BT}_2)$ 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 $\left(P, T, W^-, W^+\right)$ 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 $p \in P$, if $M_1(p) = (\mathcal{FT}_1, \mathcal{BT}_1)$ and $M_2(p) = (\mathcal{FT}_2, \mathcal{BT}_2)$ then $M_3(p) = (\mathcal{FT}_3, \mathcal{BT}_3)$ with $\mathcal{FT}_3 = \mathcal{FT}_2 - \mathcal{FT}_1$ and $\forall (n_1, t, x) \in \mathcal{BT}_1, (n_2, t, x) \in \mathcal{BT}_2$, if $n_1 \neq n_2$ then $(n_2 - n_1, t, x) \in \mathcal{BT}_3$.
- $Min\,(M, t) = \{M'/M' \subseteq M$ and M' is minimal for the firing of $t\}$.
- $get : 2^{\mathcal{H}} \longrightarrow \mathcal{H}$ is a function such that for any $A \in 2^{\mathcal{H}}$, $get\,(A) \in A$. The function get chooses in a unique manner an element of A (an event name).

3.1.1. Semantic rule

The operational semantics of labeled Petri nets allowing the generation of a maximality-based labeled transition systems is defined by:

$$\frac{t \in T \wedge t \in enabled\,(M_1)\,,\ M_3 \in Min\,(M_1, t)}{M_1 \xrightarrow{{}_{E}\,\lambda(t)_{x}} M_2}\ \text{such that:}$$



(a) **(b)**

Fig. 6. Maximally bisimilar systems

- $E = \delta(M_3)$, $M_4 = makefree(E, M_1 - M_3)$
- For any $p \in P$ with $M_4(p) = (\mathcal{FT}_4, \mathcal{BT}_4)$, $M_2(p) = (\mathcal{FT}_4, \mathcal{BT}_2)$ where: $\mathcal{BT}_2 = \mathcal{BT}_4 \cup \left\{ \left(W^+(p,t), t, x \right) / W^+(p,t) \neq 0 \right\}$
- $x = get(\mathcal{H} - (\delta(makefree(E, M_1))))$

Definition 3.3 Let $\Sigma_1 = \left(P_1, T_1, W_1^-, W_1^+, \lambda_1\right)$ and $\Sigma_2 = \left(P_2, T_2, W_2^-, W_2^+, \lambda_2\right)$ be two labeled Petri nets. Σ_1 and Σ_2 are said to be maximally bisimilar iff their associated maximality-based labeled transition systems are maximally bisimilar.

Example 3.1 Consider the example of the two labeled Petri nets Sys_1 and Sys_2 in Fig. 6a (this example is from [Dev92a]). By applying the semantic rule, the corresponding maximality-based labeled transition systems are given in Fig. 6b. The reader may easily check that these two systems are maximally bisimilar.

4. Recursive Petri nets

Recursive Petri Net model (RPN) has been proposed for specifying and analyzing dynamic systems [Sep99a, Sep99b, Dah09]. In a recursive Petri net, transitions are partitioned into two categories: elementary transitions and abstract transitions. Each abstract transition represents a thread which is modeled by its own Petri net. The structure of a recursive Petri net may be informally seen as a tree of Petri nets (denoting fatherhood relations), where each Petri net plays its own token game. In a recursive Petri net there is always ancestor Petri net which is the root of the tree. A Petri net father is the Petri net which contains abstract transitions, the firing of an abstract transition calls the execution of its own Petri net which is its son Petri net. Also a son Petri net may have in its structure abstract transitions, so it will be the father of Petri nets which model these abstract transitions. A son Petri net which has in its structure only elementary transitions cannot call the execution of any other Petri net, this son Petri net is a sheet of the tree.

The fatherhood relations between Petri nets of a recursive Petri net will be inherited at the semantics level in the same manner. However the semantics of recursive Petri nets is a dynamical tree of threads (threads are dynamically created and removed) [Sep99a]. If a thread fires an abstract transition, it consumes the input tokens of this transition and generates a new child (son thread) which begins its own token game with a predefined starting marking. When a thread finishes its execution or is preempted by an elementary transition, it produces the output tokens of the abstract transition which gave birth to it and dies (removed).

Formally, a recursive Petri net is defined as follows [Sep99a]:

Definition 4.1 A recursive Petri net is defined by $R = \left(P, T, I, W^-, W^+, \Omega, \gamma, K\right)$ such that:

- P is a finite set of places.
- T is a finite set of transitions such that $P \cap T = \emptyset$. T can be partitioned into two sets: elementary transitions T_{el}, and abstract transitions T_{ab}.
- $I = I_C \cup I_P$ is a finite set of indexes, it indicates the cut steps and preemptions. $I \subset \mathbb{N}$.

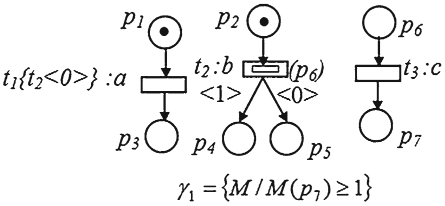


Fig. 7. Labeled recursive Petri net

- $W^- : P \times T \longrightarrow \mathbb{N}$ is the matrix of preconditions.
- $W^+ : P \times [T_{el} \cup (T_{ab} \times I)] \longrightarrow \mathbb{N}$ is the matrix of postconditions.
- $\Omega : T_{ab} \longrightarrow \mathbb{N}^P$ is a function which associates to each abstract transition an ordinary marking (starting marking).
- γ is a family indexed by the set of termination I_C. Each set is specified as an effective representation of semi linear set of final markings.
- $K : T_{el} \times T_{ab} \longrightarrow I_P$ is a partial function of control which allows the modeling of external preemption.

Definition 4.2 A labeled recursive Petri net $\Sigma = \left(P, T, I, W^-, W^+, \Omega, \gamma, K, \lambda\right)$ is a recursive Petri net in which all transitions are labeled by actions where $\lambda : T \longrightarrow \mathbb{L}$ is the labeling function.

The labeled recursive Petri net of Fig. 7 may be seen as a tree of Petri nets, this tree is formed by the Petri net R_1 and the Petri net R_2. In this labeled recursive Petri net, the ancestor Petri net is R_1. The abstract transition t_2 in R_1 is modeled by the Petri net R_2. The firing of the transition t_2 in R_1 gives birth to the execution of its own Petri net R_2. So, we say that the Petri net R_1 is the father Petri net of the Petri net R_2. We also say that R_2 is the son of R_1. In this example, we remark that the Petri net R_2 contains only elementary transitions, this Petri net is a sheet.

In the labeled recursive Petri net of Fig. 7, we have:

- $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$.
- $T = T_{el} \cup T_{ab}$ such that $T_{el} = \{t_1, t_3\}$ and $T_{ab} = \{t_2\}$.
- $I = I_C \cup I_P$ such that $I_C = \{1\}$ and $I_P = \{0\}$.
- W^-: is the matrix of preconditions:

	t_1	t_2	t_3
p_1	1	0	0
p_2	0	1	0
p_3	0	0	0
p_4	0	0	0
p_5	0	0	0
p_6	0	0	1
p_7	0	0	0

- W^+: is the matrix of postconditions:

	t_1	t_2		t_3
		$i=0$	$i=1$	
p_1	0	0	0	0
p_2	0	0	0	0
p_3	1	0	0	0
p_4	0	0	1	0
p_5	0	1	0	0
p_6	0	0	0	0
p_7	0	0	0	1

  Here we remark that when the abstract transition t_2 finishes its execution a token is added in the place p_4 (because the index of the cut step of this transition is 1). But, when the transition t_1 preempts the execution of the abstract transition t_2, a token is produced in the place p_5 (because the index of preemption of this transition is 0).
- $\Omega(t_2) = \langle 0, 0, 0, 0, 0, 1, 0 \rangle$, $\Omega(t_2)$ is the starting marking of the Petri net associated to the abstract transition t_2. In Fig. 7, $\Omega(t_2)$ is given beside the transition t_2. We remark that in the initial state of the labeled recursive Petri net there is no marking in the place p_6. When the abstract transition t_2 is fired a token is put in the place p_6 (according to the function Ω), this gives birth to its own thread represented by the execution of the Petri net R_2.
- $\gamma = \{\gamma_1\}$ with $\gamma_1 = \{M/ \mid M(p_7) \mid \geq 1\}$. γ_1 defines the termination condition of the execution of the thread corresponding to the abstract transition t_2. γ_1 means that this thread finishes its execution when there is at least one token in the place p_7.
- $K(t_1, t_2) = 0$, 0 is the index of preemption of the abstract transition t_2 by the elementary transition t_1. In Fig. 7, $t_1\, \{t_2\, \langle 0 \rangle\}$ means that the firing of the transition t_1 preempts the execution of the transition t_2 with an index of preemption equal to 0. We say here that the firing of the transition t_1 kills the thread associated to the abstract transition t_2.
- $\lambda(t_1) = a$, $\lambda(t_2) = b$, $\lambda(t_3) = c$.

5. Maximality semantics for recursive Petri nets

For explaining the intuition of maximality semantics of the recursive Petri net model, let us consider the labeled recursive Petri net of Fig. 8, in which t_2 is an abstract transition. The firing of the transition t_2 is caused by the end of execution of the action a linked to the transition t_1. The firing of this abstract transition gives birth to a new son thread associated with this abstract transition. The starting marking defined by $\Omega(t_2)$ will be prolonged to the marking of the son Petri net, this is interpreted by the addition of a token in the place p_5. For modelling the passing from the father Petri net to the son Petri net, and vice versa, we can consider two imaginary transitions labeled by *admitted* and *finished* respectively.

The execution of $admitted\,(b)$ is conditioned by the termination of a and is identified by the event name x. According to $\Omega(t_2)$, the execution of $admitted\,(b)$ is followed by the deposition of a token related to this action in the right part of the place p_5.

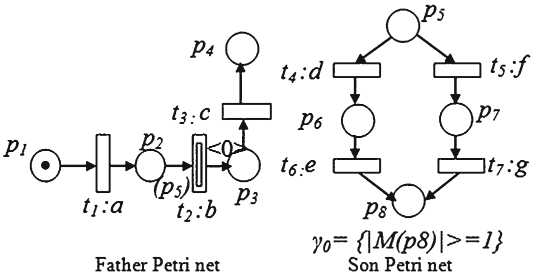


Fig. 8. Example of labeled recursive Petri net

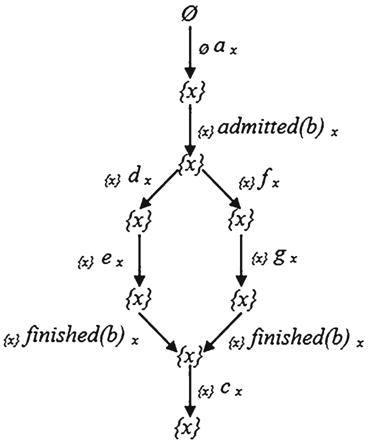


Fig. 9. Start and end of abstract transition

After the generation of a bound token in the place p_5, the son thread starts its own token game, this means that any transition that can be fired from this thread will immediately be executed. The son thread dies when the predicate of termination $\gamma_0 = \{| M(p_8) | \geq 1\}$ is satisfied. Indeed, this predicate will be satisfied when the transition t_6 or exclusively the transition t_7 deposits at least one token in the right part of the place p_8. When the predicate γ_0 becomes true, the action *finished* will be executed, it makes the return to the father thread. Indeed, this action represents the cut step of the son thread. The execution of the action *finished* (b) causes the emersion of the tokens defined by the post condition of the abstract transition in the right part of all the places which belong to the post set of this one. Just after the end of the execution of the abstract transition, the firing of the transition t_3 could happen. Figure 9 represents the maximality labeled transition system generated from this labeled recursive Petri net. We should note that the event x identifies the action *admitted* (b) as well as the start of the execution of the thread itself. Thus, it can be re-used within this thread. Once the son thread finishes, this event name can be re-used in the father thread.

5.1. Operational maximality semantics for recursive Petri nets

Preliminary definitions

- $\mathcal{THR}$: is the set of all threads.
- The state of execution of any abstract transition t is defined by its associated thread noted $\left(ST, (M)^{N}_{ref(t)}\right)$ such that:
  - $ref(t)$: is the Petri net corresponding to this thread. It describes the behavior of the abstract transition t.
  - M: is the marking of the Petri net $ref(t)$. Note that in the beginning of the execution of t the marking M is $\Omega(t)$.
  - ST: is the set of the son threads of this thread. Note that at the beginning of t the set ST is empty.
  - N: is the event name identifying this thread.
- The ancestor thread, noted $\left(\emptyset, (M_0)^{N}_{R}\right)$, is built from an initial marking M_0 of the ancestor Petri net R in the labeled recursive Petri net.
- $\psi : \mathcal{THR} \to 2^{\mathcal{H}}$: is the function which determines the event names in a thread. It is recursively defined by:
  - $\psi\left(\left(\emptyset, (M)^{N}_{R}\right)\right) = \delta(M)$
  - $\psi\left(\left(ST, (M)^{N}_{R}\right)\right) = \left(\cup_{i=1}^{n} \psi(TH_i)\right) \cup \delta(M)$ with $ST = \{TH_1, TH_2, \ldots, TH_n\}$.

- $\forall \left(ST, (M)_R^N\right) \in \mathcal{THR}$. $X \subset \mathcal{H}$ is a finite set of maximal event names of actions which terminated their executions. The operation of transforming bound tokens defined by X to free tokens in the thread $\left(ST, (M)_R^N\right)$ is defined by the inductive function $clean$ as follows:
  - $clean\left(X, \left(\emptyset, (M)_R^N\right)\right) = \left(\emptyset, (makefree\,(X, M))_R^N\right)$
  - $clean\left(X, \left(ST, (M)_R^N\right)\right) = \left(\{clean\,(X, TH_i)\,/\,TH_i \in ST\}, (makefree\,(X, M))_R^N\right)$.
- $\forall \left(ST, (M)_R^N\right) \in \mathcal{THR}$, $t \in T$ can be fired from this thread if and only if:
  - For each place p of R:$\mid M\,(p) \mid \geq W^-\,(p, t)$, or
  - $\exists\, TH' = \left(ST', (M')_{R'}^{N'}\right) \in ST$ such that: $\mid M'\,(p) \mid \geq W^-\,(p, t)$ for each place p of R'.
- The function $cutstep : \mathcal{THR} \times \gamma \to boolean$ is defined as follows:
  $\forall\, TH = \left(ST, (M)_R^N\right) \in \mathcal{THR}, \forall\, \gamma_i \in \gamma$ then:
  $$cutstep\,(TH, \gamma_i) = \begin{cases} true & if\ for\ each\ place\ p\ \ of\ R\ \ such\ that \mid M\,(p) \mid \geq n \in \gamma_i, \mid M\,(p) \mid \geq n \\ false & otherwise. \end{cases}$$
- $TH \mid t > TH'$ means that the execution of the transition t from TH leads to TH'.
- A sequence $TH_0 t_1 TH_1 t_2 ...$ is an occurrence sequence iff $TH_{i-1} \mid t_i > TH_i$ for $i \geq 1$. A sequence $\sigma = t_1 t_2 ...$ is a transition sequence starting with TH_0 iff there is an occurrence sequence $TH_0 t_1 TH_1 t_2 ...$ If a finite sequence $t_1 t_2 ... t_n$ leads from TH to TH', it is noted $TH \mid t_1 t_2 ... t_n > TH'$.

5.1.1. Semantic rules

The operational semantics of the labeled recursive Petri net model allowing the generation of a maximality-based labeled transition system is defined by the following rules:

1. $\dfrac{t \in T_{el} \wedge t \in enabled\,(M_1)\,, M_3 \in Min\,(M_1, t)}{\left(ST_1, (M_1)_R^N\right) \xrightarrow{E\ \lambda(t)_x} \left(ST_2, (M_2)_R^N\right)}$ such that:
   - $E = \delta\,(M_3)\,, M_4 = makefree\,(E, M_1 - M_3)$
   - For any place p of R, with $M_4\,(p) = (\mathcal{FT}_4, \mathcal{BT}_4)$, $M_2\,(p) = (\mathcal{FT}_4, \mathcal{BT}_2)$ where: $\mathcal{BT}_2 = \mathcal{BT}_4 \cup \left\{\left(W^+\,(p, t)\,, t, x\right) / W^+\,(p, t) \neq 0\right\}$
   - $ST_2 = ST_1$
   - $x = get\left(\mathcal{H} - \left(\psi\left(clean\left(E, \left(ST_1, (M_1)_R^N\right)\right)\right)\right)\right)$
2. $\dfrac{t \in T_{ab} \wedge t \in enabled\,(M_1)\,, M_3 \in Min\,(M_1, t)}{\left(ST_1, (M_1)_R^N\right) \xrightarrow{E\ admitted(\lambda(t))_x} \left(ST_2, (M_2)_R^N\right)}$ such that:
   - $E = \delta\,(M_3)\,, M_2 = makefree\,(E, M_1 - M_3)$
   - $ST_2 = ST_1 \cup \left\{\left(\emptyset, (M_0)_{ref(t)}^{\{x\}}\right)\right\}$ such that for any place p of $ref(t)$
     $$M_0\,(p) = \begin{cases} (0, \{(\Omega\,(t)\,(p)\,, admitted\,(t)\,, x)\}) & if\ \Omega\,(t)\,(p) \neq 0 \\ (0, \emptyset) & otherwise \end{cases}$$
   - $x = get\left(\mathcal{H} - \psi\left(clean\left(E, \left(ST_1, (M_1)_R^N\right)\right)\right)\right)$

3. $\dfrac{TH \in ST_1 \textit{ with } TH = \left(ST, (M)^{x}_{ref(t)}\right) \textit{ and } \exists \gamma_i \in \gamma / cutstep\,(TH, \gamma_i)}{\left(ST_1, (M_1)^N_R\right) \xrightarrow{_{\{x\}}finished(\lambda(t))_x} \left(ST_2, (M_2)^N_R\right)}$ such that:
   - For any place p of R, with $M_1\,(p) = (\mathcal{FT}_1, \mathcal{BT}_1)$, $M_2\,(p) = (\mathcal{FT}_1, \mathcal{BT}_2)$ where: $\mathcal{BT}_2 = \mathcal{BT}_1 \cup \left\{ \left(W^+\,(p, (t, i))\,, t, x\right) / W^+\,(p, (t, i)) \neq 0 \right\}$
   - $ST_2 = ST_1 - \{TH\}$
4. $\dfrac{t_1 \in T_{el} \wedge t_1 \in enabled\,(M_1)\,,\ M_3 \in Min\,(M_1, t) / K\,(t_1, t_2) = i \in I_P \wedge TH \in ST_1 \textit{ with } TH = \left(ST, (M)^{N_1}_{ref(t_2)}\right)}{\left(ST_1, (M_1)^N_R\right) \xrightarrow{_E\,\lambda(t_1)_x} \left(ST_2, (M_2)^N_R\right)}$

   such that:
   - $E = \delta\,(M_3)\,,\ M_4 = makefree\,(E, M_1 - M_3)$
   - For any place p of R with $M_4\,(p) = (\mathcal{FT}_4, \mathcal{BT}_4)$, $M_2\,(p) = (\mathcal{FT}_4, \mathcal{BT}_2)$ where: $\mathcal{BT}_2 = \mathcal{BT}_4 \cup \left\{ \left(W^+\,(p, t_1)\,, t_1, x\right) / W^+\,(p, t_1) \neq 0 \right\} \cup \left\{ \left(W^+\,(p, (t_2, i))\,, t_2, N_1\right) / W^+\,(p, (t_2, i)) \neq 0 \right\}$
   - $ST_2 = ST_1 - \{TH\}$
   - $x = get\left(\mathcal{H} - \left(\psi\left(clean\left(E, \left(ST_1, (M_1)^N_R\right)\right)\right) - \psi\,(TH)\right)\right)$
5. $\dfrac{t \in T,\ TH \in ST_1,\ TH \xrightarrow{_E\,\lambda(t)_x} TH'}{\left(ST_1, (M_1)^N_R\right) \xrightarrow{_E\,\lambda(t)_x} \left(ST_2, (M_1)^N_R\right)}$ such that: $ST_2 = ST_1\,[TH'/TH]$

For showing the application of the operational semantics (the five rules), we consider the recursive Petri net of Fig. 7. Its associated maximality-based labeled transition system is depicted in Fig. 10. Before applying the operational semantics, the initial configuration is constructed. This configuration is $TH_0 = \left(ST_0, (M_0)^{x_0}_{R_1}\right)$ with $M_0 = \langle (1, \emptyset)\,, (1, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\rangle$ and $ST_0 = \emptyset$. In fact, this configuration corresponds to the ancestor thread (The root of the tree) represented by the Petri net R_1. By applying the proposed semantics rules from the initial configuration (defining the initial state s_0), we obtain the maximality-based labeled transition system of Fig. 10. We remark that in the initial state s_0 there are no actions in execution, so the set of maximal events is empty ($\psi\left(\left(\emptyset, (M_0)^{x_0}_{R_1}\right)\right) = \delta\,(M_0) = \emptyset$.

From TH_0, transitions t_1 and t_2 can be fired. The transition t_1 is an elementary transition. As we have explained in Sect. 5, this transition preempts the execution of the thread associated with the transition t_2, but here we remark that in TH_0 the son threads set is empty, so there is no thread corresponding to the transition t_2. This means that there is a condition which is not satisfied in Rule 4 ($\nexists TH \in ST_0$ corresponding to $ref\,(t_2)$). In this case, the transition t_1 will be fired as an elementary transition without preemption effect. Hence, Rule 1 is applied (Rule 1 treats the firing of elementary transitions). $Min\,(M_0, t_1) = \{\langle (1, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\rangle\}$. In $Min\,(M_0, t_1)$ there is only one minimal marking, then $TH_0 \xrightarrow{_\emptyset\, a_x} TH_1$. The resulting thread TH_1 corresponds to the state s_1 of the maximality-based labeled transition system. $TH_1 = \left(ST_1, (M_1)^{x_0}_{R_1}\right)$ with:

- $M_1 = \langle (0, \emptyset)\,, (1, \emptyset)\,, (0, \{(1, t_1, x)\})\,, (0, \emptyset)\,, (0, \emptyset)\rangle$.
- $ST_1 = ST_0 = \emptyset$.

The bound token in p_3 means that the transition t_1 has been started, since t_1 preempts the abstract transition t_2, then the latter will never be fired from the state s_1.

In the state $s_1, \theta\,(s_1) = \psi\,(TH_1) = \{x\}$.

As said previously, the abstract transition t_2 may be fired from TH_0. In this case, Rule 2 is applied (Rule 2 treats the firing of abstract transitions). The firing of this abstract transition gives birth to a new thread which corresponds to the Petri net $ref\,(t_2)$ (As explained in Sect. 4, $ref\,(t_2) = R_2$), this new thread is added to the son thread set ST_0.

$Min\,(M_0, t_2) = \{\langle (0, \emptyset)\,, (1, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\rangle\}$. In $Min\,(M_0, t_2)$, there is only one minimal marking, for the latter we have $TH_0 \xrightarrow{_\emptyset\, admitted(b)_x} TH_2$, TH_2 corresponds to the state s_2 in the maximality-based labeled transition system. $TH_2 = \left(ST_2, (M_2)^{x_0}_{R_1}\right)$ with:

- $M_2 = \langle (1, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\,, (0, \emptyset)\rangle$.
- $ST_2 = ST_0 \cup \left\{ \left(\emptyset, \left(M^2_0\right)^{x}_{R_2}\right)\right\}$, we have $ST_0 = \emptyset$ so $ST_2 = \left\{ \left(\emptyset, \left(M^2_0\right)^{x}_{R_2}\right)\right\}$.

- $M_0^2 = \langle (0, \{(1, \mathit{admitted}(t_2), x)\}), (0, \emptyset) \rangle$, the first place of this marking is p_6 and the second place of this marking is p_7. We remark here that according to the starting marking $\Omega(t_2)$, the firing of the transition t_2 puts a bound token in p_6.

We say that the thread TH_0 is the father thread of the thread $\left(\emptyset, \left(M_0^2\right)_{R_2}^{x}\right)$. In other words, $\left(\emptyset, \left(M_0^2\right)_{R_2}^{x}\right)$ is the son thread of TH_0.

In the state s_2, $\theta(s_2) = \psi(TH_2) = \{x\}$.

From the thread TH_2, transitions t_1 and t_3 can be fired. For executing the transition t_1, Rule 4 is applied (Rule 4 treats the case of preemption), all conditions of this rule are satisfied. In this case, the firing of this transition preempts the execution of the thread associated to the abstract transition t_2. Then, this thread is removed from the son thread set of TH_2.

$\mathit{Min}(M_2, t_1) = \{\langle (1, \emptyset), (0, \emptyset), (0, \emptyset), (0, \emptyset), (0, \emptyset) \rangle\}$. In $\mathit{Min}(M_2, t_1)$ there is only one minimal marking which is equal to M_2. For the latter, $TH_2 \xrightarrow{\emptyset\, a_y} TH_3$, TH_3 corresponds to the state s_3 in the maximality-based labeled transition system. $TH_3 = \left(ST_3, (M_3)_{R_1}^{x_0}\right)$ with:

- $M_3 = \langle (0, \emptyset), (0, \emptyset), (0, \{(1, t_1, y)\}), (0, \emptyset), (0, \{(1, t_2, x)\}) \rangle$. We remark that a bound token linked to the transition t_1 is produced in the place p_3. Also, a bound token linked to the transition t_2 is produced in the place p_5 according to $W^+(p_5, (t_2, i))$, where $i = K(t_1, t_2) = 0$.
- $ST_3 = ST_2 - \left\{\left(\emptyset, \left(M_0^2\right)_{R_2}^{x}\right)\right\}$, we have $ST_2 = \left\{\left(\emptyset, \left(M_0^2\right)_{R_2}^{x}\right)\right\}$ so $ST_3 = \emptyset$.

In the state s_3, $\theta(s_3) = \psi(TH_3) = \{x, y\}$.

When the transition t_3 is fired from the thread TH_2, Rule 1 is applied. In fact, the transition t_3 is fired in the son thread of TH_2, so Rule 1 is applied for the son thread.

$\mathit{Min}\left(M_0^2, t_3\right) = \{\langle (0, \{(1, \mathit{admitted}(t_2), x)\}), (0, \emptyset) \rangle\}$. We remark that $\mathit{Min}\left(M_0^2, t_3\right)$ contains only one minimal marking which is equal to M_0^2. For this minimal marking, we have $\left(\emptyset, \left(M_0^2\right)_{R_2}^{x}\right) \xrightarrow{\{x\}\, c_x} \left(\emptyset, \left(M_1^2\right)_{R_2}^{x}\right)$ with $M_1^2 = \langle (0, \emptyset), (0, \{(1, t_3, x)\}) \rangle$.

Now, we have $TH \in ST_2$ such that $TH \xrightarrow{\{x\}\, c_x} TH'$. This means that the condition of Rule 5 is satisfied. By applying Rule 5 for TH_2, we have $TH_2 \xrightarrow{\{x\}\, c_x} TH_4$, TH_4 corresponds to the state s_4 of the maximality-based labeled transition system. $TH_4 = \left(ST_4, (M_4)_{R_1}^{x_0}\right)$ with:

- $M_4 = M_2$.
- $ST_4 = ST_2\left[\left(\emptyset, \left(M_1^2\right)_{R_2}^{x}\right) / \left(\emptyset, \left(M_0^2\right)_{R_2}^{x}\right)\right]$ so $ST_4 = \left\{\left(\emptyset, \left(M_1^2\right)_{R_2}^{x}\right)\right\}$

In the state s_4, $\theta(s_4) = \psi(TH_4) = \{x\}$.

From the thread TH_3 no transition can be fired. However, from the thread TH_4, the transition t_1 can be fired, then Rule 4 will be applied. In this case, the transition t_1 preempts the execution of the thread corresponding to t_2 after the firing of the transition t_3, this thread will be removed from ST_4.

$\mathit{Min}(M_4, t_1) = \{\langle (1, \emptyset), (0, \emptyset), (0, \emptyset), (0, \emptyset), (0, \emptyset) \rangle\}$. $\mathit{Min}(M_4, t_1)$ contains only one minimal marking which is equal to M_4. For this minimal marking we have $TH_4 \xrightarrow{\emptyset\, a_y} TH_5$, $TH_5 = \left(ST_5, (M_5)_{R_1}^{x_0}\right)$ with:

- $M_5 = \langle (0, \emptyset), (0, \emptyset), (0, \{(1, t_1, y)\}), (0, \emptyset), (0, \{(1, t_2, x)\}) \rangle$.
- $ST_5 = ST_4 - \left\{\left(\emptyset, \left(M_1^2\right)_{R_2}^{x}\right)\right\}$, we have $TH_4 = \left\{\left(\emptyset, \left(M_1^2\right)_{R_2}^{x}\right)\right\}$ so $ST_5 = \emptyset$.

We remark that $M_5 = M_3$ and $ST_5 = ST_3$, so we deduce that $TH_5 = TH_3$, then TH_5 corresponds to State s_3.

Also, from the thread TH_4, a cut step of the transition t_2 may be applied. We remark that in the marking M_4, the place p_7 contains a bound token which means that the condition of γ_1 is satisfied, $\mathit{cutstep}\left(\left(\emptyset, \left(M_1^2\right)_{R_2}^{x}\right), \gamma_1\right) = \mathit{true}$, so Rule 3 will be applied (Rule 3 treats the case of a cut step). $TH_4 \xrightarrow{\{x\} \mathit{finished}(b)_x} TH_6$, TH_6 corresponds to the state s_5. $TH_6 = \left(ST_6, (M_6)_{R_1}^{x_0}\right)$ with:

- $M_6 = \langle (1, \emptyset), (0, \emptyset), (0, \{(1, t_2, x)\}), (0, \emptyset) \rangle$. We remark here that the bound token is produced in the place p_4 according to $W^+(p_4, (t_2, i))$ where i is the index of the cut step $i \in I_C$, $i = 1$.
- $ST_6 = ST_4 - \left\{\left(\emptyset, \left(M_1^2\right)_{R_2}^{x}\right)\right\}$, we have $TH_4 = \left\{\left(\emptyset, \left(M_1^2\right)_{R_2}^{x}\right)\right\}$ then $ST_6 = \emptyset$.

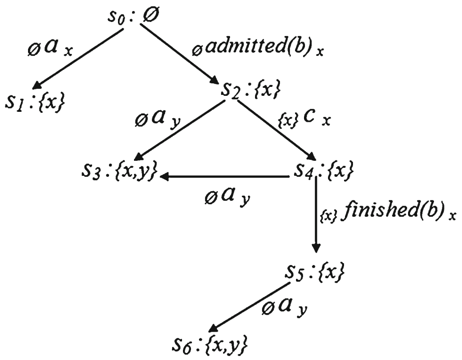


Fig. 10. MLTS obtained by the application of the operational semantics

In the state s_5, $\theta(s_5) = \psi(TH_6) = \{x\}$.

From the thread TH_6, the transition t_1 can be fired. In the thread TH_6 there is no thread corresponding to the transition t_2 because $ST_6 = \emptyset$. In this case, there is no preemption, hence Rule 1 is applied. $Min(M_6, t_1) = \{\langle(1,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset),(0,\emptyset)\rangle\}$. In $Min(M_6, t_1)$ there is only one minimal marking, for the latter $TH_6 \xrightarrow{\emptyset\, a_y} TH_7$, TH_7 corresponds to the state s_6 in the maximality-based labeled transition system. $TH_7 = \left(ST_7, (M_7)^{x_0}_{R_1}\right)$ with:

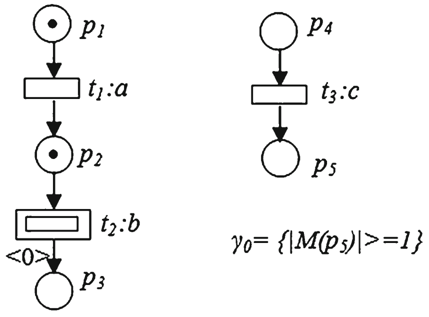
- $M_7 = \langle(0,\emptyset),(0,\emptyset),(0,\{(1,t_1,y)\}),(0,\{(1,t_2,x)\}),(0,\emptyset)\rangle$.
- $ST_7 = ST_6 = \emptyset$.

In the state s_6, $\theta(s_6) = \psi(TH_7) = \{x, y\}$.

6. An on-the-fly reduction method for recursive Petri nets

For explanation, we consider the recursive Petri net of Fig. 11 in which t_2 is an abstract transition. We remark that the thread b corresponding to this transition can be launched twice. Whether it will be launched independently from the action a, or it will be launched after the termination of the action a. All these scenarios are summarized in the maximality-based labeled transition system, this one contains 24 states and 36 transitions. Due to its size we give in Fig. 12 only a part of this graph.

Before applying the operational semantics, the initial configuration is built from the ancestor Petri net. We named the ancestor Petri net R, this configuration is $TH_0 = \left(ST_0, (M_0)^{x_0}_{R}\right)$ with $ST_0 = \emptyset$ and $M_0 = \langle(1,\emptyset),(1,\emptyset),(0,\emptyset)\rangle$. From TH_0, Rules 1 and 2 may be applied leading to the firing of the elementary transition t_1 and the abstract transition t_2, respectively.



Father Petri net **Son Petri net**

Fig. 11. Labeled recursive Petri net

The firing of the transition t_2 leads to the state s_1. We have $Min\left(M_0, t_2\right) = \{\langle(0, \emptyset), (1, \emptyset), (0, \emptyset)\rangle\}$. We remark that $Min\left(M_0, t_2\right)$ contains only one minimal marking, for the latter, we have: $TH_0 \xrightarrow{_{\emptyset}admitted(b)_x} TH_1$, $TH_1 = \left(ST_1, (M_1)_R^{x_0}\right)$ with:

- $M_1 = \langle(1, \emptyset), (0, \emptyset), (0, \emptyset)\rangle$
- $ST_1 = ST_0 \cup \left\{\left(\emptyset, \left(M_0^2\right)_{ref(t_2)}^{x}\right)\right\}$, we have $ST_0 = \emptyset$ so $ST_1 = \left\{\left(\emptyset, \left(M_0^2\right)_{ref(t_2)}^{x}\right)\right\}$

The firing of t_1 leads to the state s_2. We have $Min\left(M_0, t_1\right) = \{\langle(1, \emptyset), (0, \emptyset), (0, \emptyset)\rangle\}$. In $Min\left(M_0, t_1\right)$ there is only one minimal marking, for the latter, we have: $TH_0 \xrightarrow{_{\emptyset}a_x} TH_2$, $TH_2 = \left(ST_2, (M_2)_R^{x_0}\right)$ with:

- $M_2 = \langle(0, \emptyset), (1, \{(1, t_1, x)\}), (0, \emptyset)\rangle$
- $ST_2 = ST_1 = \emptyset$

From TH_2 the abstract transition t_2 can be fired. The firing condition of this transition is the existence of one token in p_2. $Min\left(M_2, t_2\right) = \{\langle(0, \emptyset), (1, \emptyset), (0, \emptyset)\rangle, \langle(0, \emptyset), (0, \{(1, t_1, x)\}), (0, \emptyset)\rangle\}$. We remark that in $Min\left(M_2, t_2\right)$ there are two minimal markings. The first minimal marking corresponds to the case in which we choose the free token of p_2. The second minimal marking corresponds to the case in which we choose the bound token of p_2. Since $Min\left(M_2, t_2\right)$ contains two minimal markings, Rule 2 will be applied twice from TH_2.

For the first minimal marking we have $TH_2 \xrightarrow{_{\emptyset}admitted(b)_y} TH_3$, TH_3 corresponds to the state s_3 of the maximality-based labeled transition system. $TH_3 = \left(ST_3, (M_3)_R^{x_0}\right)$ with:

- $M_3 = \langle(0, \emptyset), (0, \{(1, t_1, x)\}), (0, \emptyset)\rangle$
- $ST_3 = ST_2 \cup \left\{\left(\emptyset, \left(M_0^2\right)_{ref(t_2)}^{y}\right)\right\}$ where $M_0^2 = \langle(0, \{(1, admitted\left(t_2\right), y)\}), (0, \emptyset)\rangle$. The first place of this marking is p_4. The second place of this marking is p_5.

For the second minimal marking we have $TH_2 \xrightarrow{_{\{x\}}admitted(b)_x} TH_4$, TH_4 corresponds to State s_4 of the maximality-based labeled transition system. $TH_4 = \left(ST_4, (M_4)_R^{x_0}\right)$ with:

- $M_4 = \langle(0, \emptyset), (1, \emptyset), (0, \emptyset)\rangle$
- $ST_4 = ST_2 \cup \left\{\left(\emptyset, \left(M_{0'}^2\right)_{ref(t_2)}^{x}\right)\right\}$ where $M_{0'}^2 = \langle(0, \{(1, admitted\left(t_2\right), x)\}), (0, \emptyset)\rangle$. The first place of this marking is p_4. The second place of this marking is p_5.

We remark that from the state s_2 there is redundant derivation concerning the abstract transition t_2. Later, in this section, we show that TH_3 and TH_4 are equivalent according to the maximality bisimulation relation on transition, then we can make aggregation of these redundant derivations. We keep only the firing of the transition t_2 which is independent from the action a and we omit the firing of t_2 caused by the termination of the action a. This means that we omit the derivation $s_2 \xrightarrow{_{\{x\}}admitted(b)_x} s_4$, automatically all the derivations from the state s_4 will be omitted too, which reduces the size of the generated maximality-based labeled transition system.

6.1. Operational maximality semantics for recursive Petri nets with aggregation of transitions

Usually, a marking graph is generated in the same manner as in Sect. 5. Therefore, we keep the same semantics rules, 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 $Min\left(M, t\right)$ if and only if for each place of this marking, bound tokens are only taken in the case when the free part does not satisfy the precondition of this transition. Thus, we can ensure that a transition t will be executed sequentially after a transition t' if it cannot be executed independently from it.

Formally, $Min\left(M, t\right)$ is the set of markings $M' \subseteq M$ such that for any place p where $M\left(p\right) = \left(\mathcal{FT}, \mathcal{BT}\right)$, $M'\left(p\right)$ is defined as follows:

$$M'\left(p\right) = \begin{cases} \left(W^-\left(p, t\right), \emptyset\right) & \text{if } \mathcal{FT} \geq W^-\left(p, t\right) \\ \left(\mathcal{FT}, \mathcal{BT}'\right) & \text{otherwise} \end{cases}$$

With $\mathcal{BT}' \subseteq \mathcal{BT}$ and $\sum_{i=1}^{m} n_i = W^-\left(p, t\right) - \mathcal{FT}$ for $\mathcal{BT}' = \{(n_1, t_1, x_1), \ldots, (n_m, t_m, x_m)\}$.

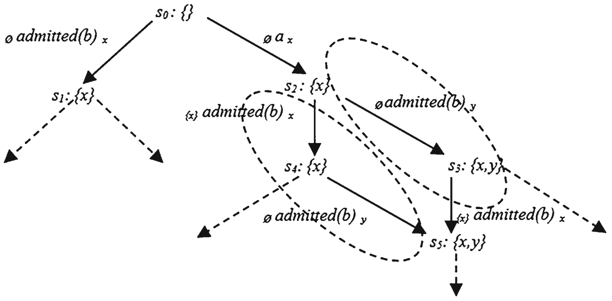


Fig. 12. Part of the corresponding MLTS

6.2. Maximality bisimulation relation on transitions

Definition 6.1 Let T be a set of transitions and $\rightarrow$ be a derivation relation between threads.

A relation $\mathfrak{R} \subseteq \mathcal{THR} \times \mathcal{THR} \times \mathfrak{F}$ is said to be a maximality bisimulation relation according to a set of transitions T iff $\forall \left(TH_i, TH_i', f\right) \in \mathfrak{R}$, $t_i \in T, f : C \longrightarrow D/C \subseteq \psi\left(TH_i\right)$ and $D \subseteq \psi\left(TH_i'\right)$:

1. *if* $TH_i \rightarrow {}_{E_i}t_{i\,x}\, TH_j$ *then* $\exists TH_i' \xrightarrow{E_i'\, t_{i\,y}} TH_j'$ such that:
   - $\forall (u, v) \in f$, if $u \notin E_i$ then $v \notin E_i'$.
   - $\left(TH_j, TH_j', f'\right) \in \mathfrak{R}/f' = \left(f\lceil(C - E_i)\rfloor\left(D - E_i'\right)\right) \cup \{(x, y)\}$.
2. *if* $TH_i' \rightarrow {}_{E_i'}t_{i\,y}\, TH_j'$ *then* $\exists TH_i \xrightarrow{E_i\, t_{i\,x}} TH_j$ such that:
   - $\forall (u, v) \in f$, if $v \notin E_i'$ then $u \notin E_i$.
   - $\left(TH_j, TH_j', f'\right) \in \mathfrak{R}/f' = \left(f\lceil(C - E_i)\rfloor\left(D - E_i'\right)\right) \cup \{(x, y)\}$.

Definition 6.2 Two labeled recursive Petri nets $\Sigma_1 = \left(P_1, T, I_1, W_1^-, W_1^+, \Omega_1, \gamma_1, K_1, \lambda_1\right)$ and $\Sigma_2 = (P_2, T, I_2, W_2^-, W_2^+, \Omega_2, \gamma_2, K_2, \lambda_2)$ are said to be maximally bisimilar according to T, noted $\Sigma_1 \approx_m^T \Sigma_2$, iff there exists a maximality bisimulation relation $\mathfrak{R}$ according to T such that $\left(TH_0^1, TH_0^2, \emptyset\right) \in \mathfrak{R}$. TH_0^1 (respectively TH_0^2) is the ancestor thread of Σ_1 (respectively Σ_2).

Notation 6.1 In the sequel the following notations will be used:

- $TH_1 \approx_m^T /f TH_2$ denotes the fact that $(TH_1, TH_2, f) \in \mathfrak{R}$, for $\mathfrak{R}$ a maximality bisimulation relation according to a set of transitions T.
- $\approx_m^{\mathbb{L}}$ is the maximality bisimulation relation according to a set of actions $\mathbb{L}$. $\approx_m^{\mathbb{L}}$ is similar to $\approx_m^T$ where instead of $t \in T$ we use $\lambda(t) \in \mathbb{L}$.

Corollary 6.1 $\approx_m^T \subseteq \approx_m^{\mathbb{L}}$.

Proof. Obvious, because $\approx_m^T$ is more discriminating than $\approx_m^{\mathbb{L}}$, two different transitions t_i and t_j may have the same label $(\lambda(t_i) = \lambda(t_j))$. If λ is bijective then $TH_i \approx_m^T TH_j \Longrightarrow TH_i \approx_m^{\mathbb{L}} TH_j$. □

Proposition 6.1 Let TH be a thread with $A \subseteq \psi(TH)$ then:

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

Proof.

1. Let $\mathfrak{R}_{Id/TH} = \left\{\left(TH, TH, Id_{\psi(TH)}\right)\right\}$. We prove that $\forall\, TH, \mathfrak{R}_{Id/TH}$ is a maximality bisimulation relation on a set of transitions.
   Let $\mathfrak{R}_{Id/TH} = \cup_{i \geq 0}\mathfrak{R}_i$ such that:
   $\mathfrak{R}_0 = \left\{\left(TH, TH, Id_{\psi(TH)}\right)\right\}$.

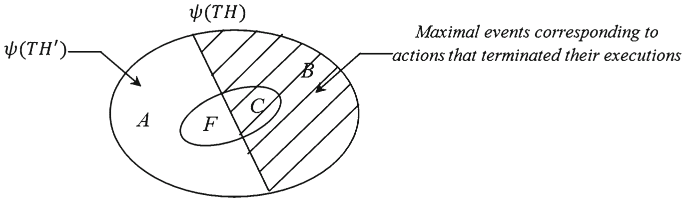


Fig. 13. Set of events names of thread TH (proposition 6.1.3)

$\mathfrak{R}_1 = \left\{\left(TH_1, TH_1, Id_{\psi(TH_1)}\right)\right\}$ such that $\exists TH \xrightarrow{E\, t_x} TH_1$.

$\mathfrak{R}_{n+1} = \left\{\left(TH_{n+1}, TH_{n+1}, Id_{\psi(TH_{n+1})}\right)\right\}$ such that $\exists \left(TH_n, TH_n, Id_{\psi(TH_n)}\right) \in \mathfrak{R}_n$ and $TH_n \xrightarrow{E_{n+1}\, t_{x_{n+1}}} TH_{n+1}$

Then $\forall \left(TH_i, TH_i, Id_{\psi(TH_i)}\right) \in \mathfrak{R}_{Id/TH}$, $\psi\,(TH_i) \subseteq \psi\,(TH_i)$ and if $TH_i \xrightarrow{E_{i+1}\, t_{x_{i+1}}} TH_{i+1}$ then for $E'_{i+1} = E_{i+1}$ and for $x'_{i+1} = x_{i+1}$, $TH_i \xrightarrow{E'_{i+1}\, t_{x'_{i+1}}} TH_{i+1}$. On the other hand $\forall\, u \in \psi\,(TH_i)$, if $u \notin E_{i+1}$ then $u \notin E'_{i+1}$ because $E'_{i+1} = E_{i+1}$ then $(TH_{i+1}, TH_{i+1}, Id_{A_{i+1}}) \in \mathfrak{R}_{Id/TH}$ such that $A_{i+1} = (\psi\,(TH_i) - E_i) \cup \{x_{i+1}\} = \psi\,(TH_{i+1})$, consequently $(TH_{i+1}, TH_{i+1}, Id_{A_{i+1}}) \in \mathfrak{R}_{Id/TH}$. We deduce that $\mathfrak{R}_{Id/TH}$ is a maximality bisimulation relation on the set of transitions.

2. Let $\mathfrak{R}_{Id/A} = \{(TH, TH, Id_A)\}$. We prove that: for any TH and $A \subseteq \psi(TH)$, $\mathfrak{R}_{Id/A}$ is a maximality bisimulation relation on the set of transitions.
   Let $\mathfrak{R}_{Id/A} = \cup_{i\geq 0}\mathfrak{R}_i$ such that:
   $\mathfrak{R}_0 = \{(TH, TH, Id_A)\}$.
   $\mathfrak{R}_1 = \left\{(TH_1, TH_1, Id_{A_1})\right\}$ such that $\exists TH \xrightarrow{E\, t_x} TH_1$ and $A_1 = (A - E) \cup \{x\}$.
   $\mathfrak{R}_{n+1} = \left\{(TH_{n+1}, TH_{n+1}, Id_{A_{n+1}})\right\}$ such that $\exists (TH_n, TH_n, Id_{A_n}) \in \mathfrak{R}_n$ and $\exists\, TH_n \xrightarrow{E_{n+1}\, t_{x_{n+1}}} TH_{n+1}$ with $A_{n+1} = (A_n - E_{i+1}) \cup \{x_{i+1}\}$.
   Then for any $(TH_i, TH_i, Id_{A_i}) \in \mathfrak{R}_{Id/A}$ with $A_i \subseteq \psi\,(TH_i)$
   If $TH_i \xrightarrow{E_{i+1}\, t_{x_{i+1}}} TH_{i+1}$ then for $E'_{i+1} = E_{i+1}$ and $\exists\, x'_{i+1} = x_{i+1}$, $TH_i \xrightarrow{E'_{i+1}\, t_{x'_{i+1}}} TH_{i+1}$. On the other hand $\forall\, u \in A_i$, if $u \notin E_{i+1}$ then $u \notin E'_{i+1}$ because $E'_{i+1} = E_{i+1}$ then $(TH_{i+1}, TH_{i+1}, Id_{A_{i+1}}) \in \mathfrak{R}_{Id/A}$ such that $A_{i+1} = (A_i - E_{i+1}) \cup \{x_{i+1}\}$. We deduce that $\mathfrak{R}_{Id/A}$ is a maximality bisimulation relation on the set of transitions.

3. $B = \psi\,(TH) - A$ with $A \subseteq \psi\,(TH)$; Let $TH' = clean\,(B, TH)$: we build a relation $\mathfrak{R}$ containing (TH', TH, Id_A) and we prove that $\mathfrak{R}$ is a maximality bisimulation relation for the set of transitions T. Let $\mathfrak{R} = \left\{\left(TH', TH, Id_{\psi(TH')}\right)\right\}$ such that $TH' = clean\,(\psi\,(TH) - \psi\,(TH'), TH)$, note that $\psi\,(TH) - \psi\,(TH') = B$ and $\psi\,(TH') = A$.
   i. If $TH' \xrightarrow{F\, t_u} TH'_1$, $\exists\, C \subseteq B$ such that $TH \xrightarrow{C\cup F\, t_v} TH_1$ with:
   $u = get\,(\mathcal{H} - (\psi\,(TH') - F))$ and $v = get\,(\mathcal{H} - (\psi\,(TH) - (C \cup F)))$.
   $\psi\,\left(TH'_1\right) = (\psi\,(TH') - F) \cup \{u\}$.
   $\psi\,\left(TH'_1\right) = ((\psi\,(TH) - B) - F) \cup \{u\}$.
   $\psi\,(TH_1) = (\psi\,(TH) - (C \cup F)) \cup \{v\}$.
   $\psi\,(TH_1) = ((\psi\,(TH) - C) - F) \cup \{v\}$.
   $\psi\,(TH_1) - \{v\} = \left(\psi\,\left(TH'_1\right) - \{u\}\right) \cup (B - C)$.
   Let: $w = get\,(\mathcal{H} - ((\psi\,(TH') - F) \cup (\psi\,(TH) - (C \cup F))))$ since $\psi\,(TH') - F \subseteq \psi\,(TH) - (C \cup F)$ then

   $w = get\,(\mathcal{H} - (\psi\,(TH) - (C \cup F)))$ which implies that $w = v$. $\psi\,(TH_1) = \left(\psi\,\left(TH'_1\,[v/u]\right) \cup (B - C)\right)$ however $(B - C) \cap \psi\,\left(TH'_1\,[v/u]\right) = \emptyset$, then $clean\,\left((B - C), TH'_1\right) = TH'_1\,[v/u]$ which implies that $\left(TH'_1\,[v/u], TH'_1, Id_{\psi\left(TH'_1[v/u]\right)}\right) \in \mathfrak{R}$.
   Let: $x \in A - F$, since $A \cap B = \emptyset$, then $x \in A - (F \cup B)$ and because $C \subseteq B$ then $x \in A - (C \cup F)$.
   ii. If $TH \xrightarrow{C\cup F\, t_v} TH_1$ then $TH' \xrightarrow{F\, t_u} TH'_1$ such that $C \subseteq B$ with: $v = get\,(\mathcal{H} - (\psi\,(TH) - (C \cup F)))$ and $u = get\,(\mathcal{H} - (\psi\,(TH') - F))$.

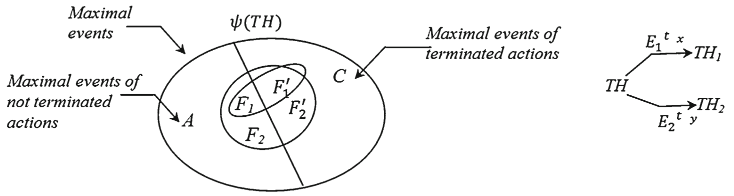


Fig. 14. Division of the set $\psi\,(TH)$ (Proposition 6.2)

We have proved in 3.i that: $\psi\,(TH_1) = \left(\psi\,\left(TH_1'[v/u]\right) \cup (B - C)\right)$ and $clean\,((B - C)\,,TH_1) = TH_1'[v/u]$, which implies that $\left(TH_1'[v/u]\,,TH_1, Id_{\psi\left(TH_1'[v/u]\right)}\right) \in \mathfrak{R}$.

Let $x \in A - (C \cup F)$ and because $A \cap C = \emptyset$ then $x \in (A - F)$, from (i) and (ii) $\mathfrak{R}$ is a maximality bisimulation relation on the set of transitions T. □

In order to propose a method for choosing an adequate derivation among a set of derivations leading to bisimilar states, we have to consider a case of derivations of length one from any states. This is the purpose of the following proposition.

Proposition 6.2 Let $A \subseteq \psi\,(TH)$. If $TH \xrightarrow{E_1\,t_x} TH_1$ and $TH \xrightarrow{E_2\,t_y} TH_2$ such that $A \cap E_1 \subseteq A \cap E_2$ then, for $z = get\,(\mathcal{H} - ((\psi\,(TH) - E_1) \cup (\psi\,(TH) - E_2)))$ and $B = (A - E_2) \cup \{z\}$ we have $TH_1\,[z/x] \approx_m^T /Id_B\; TH_2\,[z/y]$.

Proof. Let $C = \psi\,(TH) - A$, $TH' = clean\,(C, TH)$

Proposition 6.1.(1) $\Longrightarrow TH' \approx_M^T /Id_A\; TH$. Let $F_1 = A \cap E_1$, $F_2 = A \cap E_2$ then $F_1 \subseteq F_2$.

Let the events names x, y, u and v defined as follows:
$x = get\,(\mathcal{H} - (\psi\,(TH) - E_1))$.
$y = get\,(\mathcal{H} - (\psi\,(TH) - E_2))$.
$u = get\,(\mathcal{H} - (\psi\,(TH') - F_1))$.
$v = get\,(\mathcal{H} - (\psi\,(TH') - F_2))$.
The sets E_1 and E_2 can be partitioned as follows:
$F_1' \cup F_1 = E_1$ and $F_1 \cap F_1' = \emptyset$ such that: $\forall\, x \in F_1'$ then $x \in C$ and $F_1 \subseteq A$.
$F_2' \cup F_2 = E_2$ and $F_2' \cap F_2 = \emptyset$ such that: $\forall\, x \in F_2'$, $x \in C$ and $F_2 \subseteq A$.
By hypothesis we have: $\psi\,(TH_1) = (\psi\,(TH) - E_1) \cup \{x\}$ hence $\psi\,(TH_1) = \left((C \cup A) - \left(F_1' \cup F_1\right)\right) \cup \{x\}$.
$\psi\,(TH_1) = \left(\left(C - F_1'\right) \cup (A - F_1)\right) \cup \{x\}$.
$TH_1' = clean\,\left(C - F_1', TH_1\right)[u/x]$.
$\psi\,\left(TH_1'\right) = (A - F_1) \cup \{u\}$.

When $\psi\,(TH_2) = (\psi\,(TH) - E_2) \cup \{y\}$ then $\psi\,(TH_2) = \left((C \cup A) - \left(F_2' \cup F_2\right)\right) \cup \{y\}$ from where $\psi\,(TH_2) = \left(\left(C - F_2'\right) \cup (A - F_2)\right) \cup \{y\}$.
On the other hand $\psi\,\left(TH_2'\right) = (A - F_2) \cup \{v\}$ then $TH_2' = clean\,\left(\left(C - F_2'\right), TH_2\right)[v/y]$.
Proposition 6.1 implies: $TH_1 \approx_m^T /Id_{((A-F_1)\cup\{x\})}\, clean\,\left(\left(C - F_1'\right), TH_1\right)$ from where $TH_1 \approx_m^T /Id_{((A-F_1)\cup\{x\})}\, TH_1'$ and $TH_2 \approx_m^T /Id_{((A-F_2)\cup\{y\})}\, clean\,\left(\left(C - F_2'\right), TH_2\right)$ from where

$TH_2 \approx_m^T /Id_{((A-F_2)\cup\{y\})}\, TH_2'$. We want to prove that $TH_1 \approx_m^T /f TH_2$ such that $f = Id_{(A-F_2)\cup\{(x,y)\}}$.
For this it suffices to show that:
$TH_1' \approx_m^T /f'\, TH_2'$ such that $f' = Id_{(A-F_2)\cup\{(u,v)\}}$. We remark that f' is a bijection.
We have: $u = get\,(\mathcal{H} - (\psi\,(TH') - F_1)) = get\,(\mathcal{H} - (A - F_1))$.
$v = get\,(\mathcal{H} - (\psi\,(TH') - F_2)) = get\,(\mathcal{H} - (A - F_2))$.
Let $w = get\,(\mathcal{H} - (A - (F_1 \cup F_2)))$ with the fact that $F_1 \subseteq F_2$ then :
$w = get\,(\mathcal{H} - (A - F_2)) = v$.
We remark that $v \notin \left(\psi\,\left(TH_1'\right) - \{u\}\right)$.
Let $TH_1'' = TH_1'[v/u]$ and $F = F_2 - F_1$, then $clean\,\left(F, TH_1''\right) = TH_2'$.
Proposition 6.1 implies $TH_1'' \approx_m^T /Id_{((A-F_2)\cup\{v\})}\, TH_2'$ we can deduce that $TH_1 \approx_m^T /f TH_2$. □

Proposition 6.2 states that we can construct a maximality bisimulation relation relating TH_1 and TH_2 by adding elements of $A \cap (E_2 - E_1)$ to maximal events in the resulting state TH_1 and which correspond to actions

that terminated their executions. In fact, though these events remain maximal, the deduction of the termination of their corresponding actions is provided from the fact that the related events by f correspond to actions which terminated their execution in TH_2 (they are not maximal in TH_2). In the example of Fig. 12, we have $TH_2 \xrightarrow{\emptyset\, admitted(b)_y} TH_3$, $TH_2 \xrightarrow{\{x\}\, admitted(b)_x} TH_4$.
$\psi\,(TH_2) = \{x\}\,,\, E_1 = \emptyset,\, E_2 = \{x\}$. If we take $A = \{x\}$, $A \cap E_1 \subseteq A \cap E_2$. $B = (A - E_2) \cup \{z\} = \{z\}$, using the Proposition 6.2 we deduce that $TH_3\,[z/y] \approx_m^T /Id_B\, TH_4\,[z/x]$.

Proposition 6.3 Let Σ be a labeled recursive Petri net with the initial marking M_0, and let σ be a sequence of transitions such that $TH_0|\sigma > TH$ and $TH_0|\sigma > TH'$ with $D \subseteq \psi\,(TH)$, $C \subseteq \psi\,(TH')$ and $f : D \to C$ a bijection, then $TH \approx_m^T /f\; TH' \Rightarrow clean\,(\psi\,(TH) - D,\, TH) \approx_m^T /f\; clean\,(\psi\,(TH') - C,\, TH')$.

Proof. From Proposition 6.1 we deduce that:
$TH \approx_m^T /Id_D\, clean\,(\psi\,(TH) - D,\, TH)$ and $TH' \approx_m^T /Id_C\, clean\,(\psi\,(TH') - C,\, TH')$.
By hypothesis, $TH \approx_m^T /f TH'$, with $f : D \longrightarrow C$ then by transitivity we deduce that:
$clean\,(\psi\,(TH) - D,\, TH) \approx_m^T /f clean\,(\psi\,(TH') - C,\, TH')$, then the result. □

Proposition 6.3 states that in the case when the sequence of transitions defined by σ is executed from the initial thread TH_0 and leads to two different threads TH and TH' which are maximally bisimilar such that only sets $D \subseteq \psi\,(TH)$ and $C \subseteq \psi\,(TH')$ are related by the function f, then TH and TH' remain maximally bisimilar when maximal events in $\psi\,(TH) - D$ and $\psi\,(TH') - C$ are removed from TH and TH' respectively. In fact, these maximal events correspond to actions that have terminated their executions.

As an illustration of such a case, let us consider the example of Fig. 12. From the state s_2, the firing condition of the transition t_2 labeled by b is the existence of one token in the place p_2. At this state, two cases may be considered as we have explained in the calculation of $Min\,(M_2,\, t_2)$: In the first case we can choose the free token, which leads to the state s_3. In the second case, we can choose the token bounded to the transition t_1, which leads to the state s_4.

For $\sigma = t_1 t_2$, we have $TH_0|\sigma > TH_3$ and $TH_0|\sigma > TH_4$. $\psi\,(TH_3) = \{x,\, y\}$ and $\psi\,(TH_4) = \{x\}$. If we take $D = \{y\}$ and $C = \{x\}$ then TH_3 and TH_4 are maximally bisimilar. Using the Proposition 6.3, we can remove the event name x in TH_3 and the resulting threads remain maximally bisimilar: we have $\psi\,(TH_3) - D = \{x\}$ and $\psi\,(TH_4) - C = \emptyset$, so $clean\,(\{x\}\,,\, TH_3) \approx_m^T /f\; clean\,(\emptyset,\, TH_4)$.

Theorem 6.1 Let $\Sigma = \left(S,\, T,\, I,\, W^-,\, W^+,\, \Omega,\, \gamma,\, K,\, \lambda\right)$ be a labeled recursive Petri net and let $mlts_1$ (resp. $mlts_2$) be a maximality-based labeled transition system generated from Σ by using the maximality semantics (resp. the maximality semantics with aggregation of transitions), then $mlts_1$ and $mlts_2$ are maximally bisimilar for the set of transitions T.

Proof. It is a consequence of Definition 6.2 and Proposition 6.2.
It is worth to note that $mlts_2$ is isomorphic to a sub-graph of $mlts_1$. Proposition 6.2 implies that for any state in $mlts_1$ which is not in $mlts_2$ there exists a state in $mlts_1$ and $mlts_2$ which is maximally bisimilar to it. Then the result. □

Let us consider the labeled recursive Petri net of Fig. 11. When we generate the reduced MLTS associated with this labeled recursive Petri net using the semantics rules with the new function Min, we obtain the same derivations, with the same values of threads: $TH_0 \xrightarrow{\emptyset\, admitted(b)_x} TH_1$, $TH_0 \xrightarrow{\emptyset\, a_x} TH_2$. However, now when we calculate the set of minimal marking for the firing of the abstract transition t_2, we find only one minimal marking.

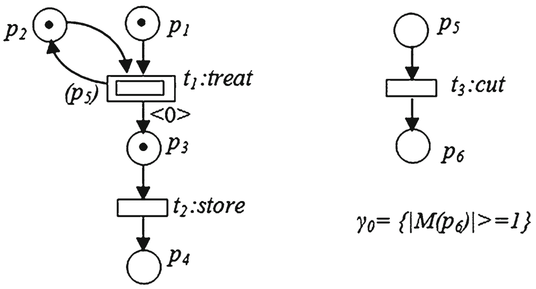


Fig. 15. Modeling of woodshop cutting

Table 1. Number of states and transitions before/after reduction

i=	Command	Piece	S before	T before	S after	T after	Reduction S (%)	Reduction T (%)
1	1	1	14	16	13	14	7.14	12.50
2	2	1	42	55	38	46	9.52	16.36
3	2	2	88	121	66	86	25.00	28.93
4	3	2	212	304	150	200	29.50	34.21
5	3	3	421	610	218	301	48.22	50.66
6	4	3	929	1369	448	617	51, 78	54.93
7	4	4	1830	2701	590	829	67.76	69.31
8	5	4	3874	5764	1165	1619	69.93	71.91
9	5	5	7648	11,374	1443	2030	81.13	82.15
10	6	6	31,377	46,831	3346	4682	89.34	90.00

This minimal marking corresponds to the case in which we choose the free token of the place p_2, $Min\,(M_2, t_2) = \{\langle (0, \emptyset), (1, \emptyset), (0, \emptyset)\rangle\}$. Then Rule 2 is applied only once from the thread TH_2.

For this minimal marking, we have $TH_2 \xrightarrow{{}_{\emptyset}admitted(b)_y} TH_3$. Therefore, in the resulting MLTS we have only one derivation from the state s_2, $s_2 \xrightarrow{{}_{\emptyset}admitted(b)_y} s_3$. The derivation $TH_2 \xrightarrow{{}_{\{x\}}admitted(b)_x} TH_4$ will never happen. As we see, in this MLTS the derivation $s_2 \xrightarrow{{}_{\{x\}}admitted(b)_x} s_4$ is omitted on the fly. Automatically all the derivations from the state s_4 are omitted too.

7. Case study

A workshop consists of a cutting machine and a stock of wood. When an order arrives and the cutting machine is available, the command can be treated (cut operation). Otherwise, the command is delayed until the cutting machine is released. When the treatment terminates, the command will be stored. The modeling of this system is shown in Fig. 15.

- The number of tokens in the place p_1 indicates the number of pending commands.
- A token in the place p_2, indicates that the cutting machine is free.
- The number of tokens in the place p_3 indicates the number of cut pieces.
- The place p_4 stores the number of treated pieces.

In order to show the reduction rate, we use first the generation method of maximality-based labeled transition systems presented in Sect. 5. After that, we apply the generation method with reduction. This is achieved by varying the number of commands and the number of cut pieces. The obtained results are summarized in Table 1.

Figure 16 shows the difference between the number of states of the maximality-based labeled transition systems generated by means of the two methods.

Figure 17 shows the difference between the number of transitions of the maximality-based labeled transition systems generated by means of the two methods.

Figure 18 presents the reduction rate of states and transitions obtained by the proposed method.

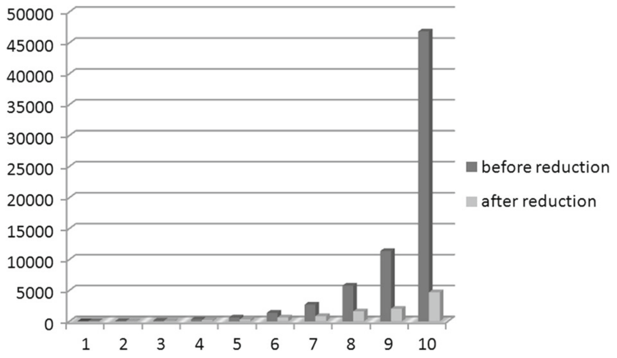


Fig. 16. Number of states

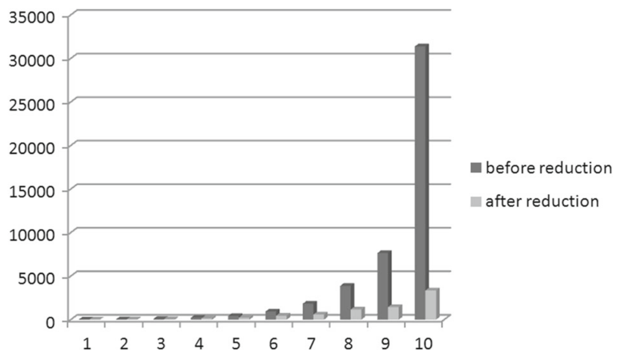


Fig. 17. Number of transitions

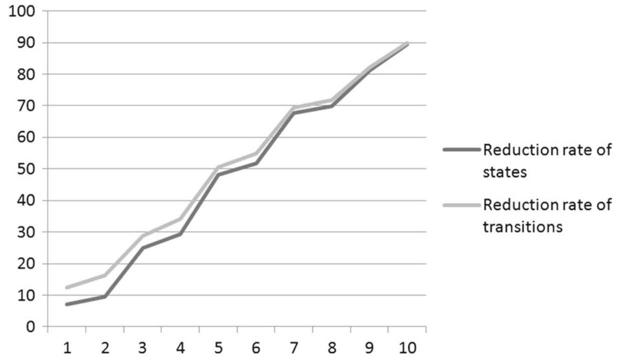


Fig. 18. Reduction rate

8. Conclusion

In this paper, we have proposed an operational method for generating a reduced maximality labeled transition system associated with recursive Petri nets. The method consists of the case of two possible derivations of a transition, to conserve only the derivation allowing more parallelism (less causal dependencies). As a result, we have shown that the obtained maximality based labeled transition system is equivalent to the one obtained without transition aggregation according to the maximality bisimulation relation. Then, the degree of parallelism and system properties are preserved. Also, by means of a case study, we have shown that the proposed method significantly reduces the size of the maximality-based labeled transition system.

In [Bes13], authors have proposed an algorithm for reducing maximality-based labeled transition systems modulo a maximality bisimulation relation $\approx_m^{\mathbb{L}}$. Though this relation is weaker than the relation $\approx_m^{\mathbb{T}}$ considered in our work (see Corollary 6.1), the proposed algorithm assumes the existence of a maximality-based labeled transition system, this later is supposed generated firstly from a high level specification written in any specification model like Petri net, recursive Petri net, Lotos,. . . . Thus, the proposed approach is only useful for specifications for which the corresponding maximality-based transition systems may be contained in the used computer storage. From this point of view, our approach is more interesting because it on the fly generates the reduced maximality-based labeled transition system from the specification. As future work, in order to improve the reduction rate of the maximality-based labeled transition system, it is interesting to generalize our approach to the maximality bisimulation relation $\approx_m^{\mathbb{L}}$.

References

[Ach91] Aceto L, Hennessy M (1991) Adding action refinement to finite process algebra. In: Albert JL, Monien B, Artalejo MR (eds) ICALP91, LNCS, vol 510. Springer, pp 506–519

[Agm93] Andrews D, Groote J, Middelburg C (eds) (1993) Workshop on Semantics of Specification Languages (SoSL93), Workshops in Computing, Springer-Verlag, Utrecht, pp 289-303

[Arn92] Arnold A (1992) Systme de transition finis et smantique des processus communicants. Masson, Paris

[Bes13] Benamira A, Saidouni DE (2013) Maximality-based labeled transition systems normal form, modeling approaches and algorithms for advanced computer applications, studies in computational intelligence. 488:337–346

[Bdk91] Best E, Devillers R, Kiehn A, Pomello L (1991) Concurrent bisimulations in Petri nets. Acta Inf 28:231–264

[Boc88] Boudol G, Castellani I (1988) Concurrency and atomicity. TCS 59:1–60

[Ces86] Clarke EM, Emerson EA, Sistla AP (1986) Automatic verification of finite state concurrent systems using temporal logic specifications. ACM Trans Program Lang Syst 8(2):244–263

[Clh93] Cleveland R, Hennessy M (1993) Testing equivalence as a bisimulation equivalence. Form Asp Comput 5:1–20

[Cos94] Courtiat JP, Sadouni DE (1994) Action refinement in LOTOS. In: Danthine A, Leduc G, Wolpe P (eds) Protocol Specification, Testing and Verification (PSTV93), North-Holland, pp 341–354

[Cos95] Courtiat JP, Sadouni DE (1995) Relating maximality-based semantics to action refinement in process algebras

[Dah09] Dahmani D (2009) Extensions of recursive Petri nets for temporal analysis of systems with dynamic structure. Doctoral thesis. University of Science and Technology Houari Boumediene algeria

[Dad89] Darondeau P, Degano P (1989) Causal trees. In: ICALP'89, LNCS, vol 372. Springer, pp 234-248

[Dad91] Darondeau P, Degano P (1991) About semantic action refinement. Fundam Inf 14:221–234

[Dad93] Darondeau P, Degano P (1993) Refinement of actions in event structures and causal trees. TCS 118:21–48

[Deg91] Degano P, Gorrieri R (1991) Atomic refinement in process description languages. In: Tarlecki A (ed) Mathematical Foundations of Computer Science, LNCS, vol 520. Springer pp 121–130

[Dev92a] Devillers R (1992) Maximality preservation and the ST-idea for action refinement. In: Rozenberg G (ed) Advances in Petri Nets, LNCS, vol 609. Springer pp 108–151

[Dev92b] Devillers R (1992) Maximality preserving bisimulation. TCS 102:165–183

[Dev93] Devillers R (1993) Construction of S-invariants and S-components for refined Petri boxes. In: Marsan MA (ed) ATPN93, LNCS, vol 691. Springer, pp 242–261

[Dij71] Dijkstra EW (1971) Hierarchical ordering of sequential processes. Acta Inf 1(2):115–138

[Hol94] Hogrefe D, Leue S (eds) (1994) IFIP TC6/WG6.1, 7th Int. Conf. on Formal Description Techniques (FORTE94), Chapman, Hall, pp 293–308

[Jpz91] Janssen W, Poel M, Zwiers J (1991) Action systems and action refinement in the development of parallel systems, CONCUR91, LNCS, vol 527. Springer, pp 298–316

[Sai96] Saidouni DE (1996) Maximality semantic: Application to actions refinement in LOTOS, Ph.D. thesis., LAAS-CNRS, 7 av. du Colonel Roche, 31077 Toulouse Cedex France

[Sbb08a] Saidouni DE, Belala N, Bouneb M (2008) Using maximality- based labeled transitions as model for Petri nets. The International Arab Conference on Information Technology(ACIT)

[Sbb08b] Saidouni DE, Belala N, Bouneb M (2008) Aggregation of transitions in marking graph generation based on maximality semantics for Petri nets. Verification and Evalution of Computer and Communication systems (VECOS)

[Sbb09a] Saidouni DE, Belala N, Bouneb M (2009) Using maximality- based labeled transitions as model for Petri nets. Int Arab J Inf Technol (IAJIT)

[Sbb09b] Saidouni DE, Belala N, Bouneb M (2009) Maximality-Based Structural Operational Semantics for Petri Nets. In: 2nd Mediterranean Conference on Intelligent Systems and Automation (CISA)

[Sac03] Saidouni DE, Courtiat JP (2003) Prise en compte de durées d'actions dans les algébres de processus par l'utilisation de la sémantique de maximalité. In ingénierie des protocoles (CFIP'03). Hermes, France

[Sac94] Saidouni DE, Courtiat JP (1994) Syntactic action refinement in presence of multiway synchronization. Semant Specif Lang 289–303

[Sbi13] Saidouni DE, Bouneb M, Ilie JM (2013) Maximality semantic for recursive Petri net. Europeen conference on modelling and simulation(ECMS'13). pp 544–550. ISBN 978-0-9564944-7-4

[Sep99a] Serge H, Denis P (1999) Recursive Petri nets: theory and application to discrete event systems. Acta Inf

[Sep99b] Serge H, Denis P (1999) Theoretical Aspects of Recursive Petri Nets. Application and Theory of Petri Nets. Lecture Notes in Computer Science, volume 1639. Springer pp 228–247

[Van90] Van Glabbeek RJ (1990) The refinement theorem for ST-bisimulation semantics. In: IFIP Working Conference on Programming Concepts and Methods, North-Holland

[vog93] Vogler W (1993) Bisimulation and actions refinement. TCS 114:173–200

Received 2 October 2014
Revised 28 June 2015
Accepted 29 June 2015 by Amilcar Sernadas