Behavioural verification of limited resources systems under true concurrency semantics

1st Bouneb. Messaouda
Department of mathematics and computer science
EL Arbi Ben Mhidi University
Oum El Bouaghi, Algeria
bounebm.univ@gmail.com

2nd Saidouni. Djamel Eddine
Department of computer science
Abed El Hamid Mehri Constantine2
Constantine, Algeria
saidounid@hotmail.com

***Abstract*—In this paper we propose a true concurrency semantics for limited resources systems using K-bounded Petri net as modeling formalism and maximality labeled transition system (MLTS) as semantics model. Indeed the model of MLTS expresses clearly the semantics of true parallelism of concurrent systems. The proposed operational maximality semantics for K-bounded Petri nets makes it possible to interpret any K-bounded Petri net in terms of MLTS. Through an example we show the interest of the proposed semantics in comparison with the interleaving semantics and the ST semantics. The comparison concerns the preservation of true concurrency and the reduction of the size of the semantics model. Furthermore, we will show that expected CTL properties may be verified on the corresponding maximality labeled transition system of a modeled system using our developed tool.**

***Index Terms*—Formal verification, maximality labeled transitions system, concurrent systems, K-bounded Petri net**

I. Introduction

Formal verification of complex systems is now a major issue in many areas. Indeed, the use of methods of specification and formal verification, assisted by powerful computer tools make the analysis of these systems reliable and guarantees a good compromise between cost and performance. The Petri net model is a graphical and mathematical modelling tool used to specify clearly concurrent systems behaviours. The marking graph associated to the Petri net is used to check the properties of the specified system. Indeed this markings graph is seen as a labeled transition system (LTS). However, the model of LTS is an interleaved model that makes abstraction of the parallel execution of transitions. To clarify the ideas, we recall the example of the two Petri nets of Figures 1.(a) and Figure 1.(b) presented in [1] [2]. The Petri net of Figure 1.(a) represents a system able of executing the transitions t_1 and t_2 in parallel, while the Petri net of Figure 1.(b) represents a system which executes either the transition t_1 then t_3 or the transition t_2 then t_4.

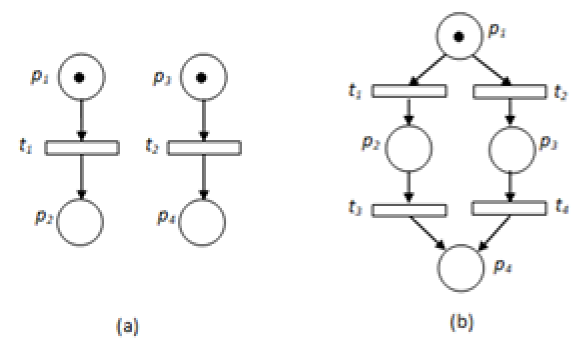

Fig. 1. Ordinary Petri net.

The marking graphs corresponding to the Petri net of Figure 1.(a) and Figure 1.(b) are given respectively by Figure 2.(a) and Figure 2.(b). When the transitions t_1 and t_4 are labeled by the action a and the transitions t_2 and t_3 by the action b, we remark that these marking graphs are isomorphic. Consequently, the parallel execution of the actions a and b is interpreted as the interleaved execution of these two actions in time.

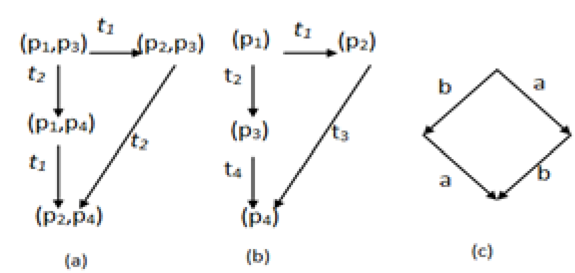

Fig. 2. Interleaving semantics.

This result is acceptable under the assumption that the firing of each transition corresponds to the execution of an indivisible action with zero duration (structural and temporal atomicity of the actions). This hypothesis is far from acceptable in the reality.

In order to accept the results of the verification, it is imperative that the constraints imposed by the real world are taken into account both by the specification and by the underlying semantic model. To support our claim, let us now reconsider that the transition t_1 (resp. t_4) consists of two sequential transitions t_{1-1} followed by t_{1-2} (resp t_{4-1} followed by t_{4-2}), the transitions t_{1-1} and t_{4-1} are labeled by the action

a_1 while the transitions t_{1-2} and t_{4-2} are labeled by the action a_2. The refined Petri nets as well as their labeled transitions systems are represented in Figure 3. It is clear that the behaviours of these two Petri nets are different. Indeed, in the first system, the execution of action b may occur between the execution of actions a_1 and a_2, which is not possible in the second system.

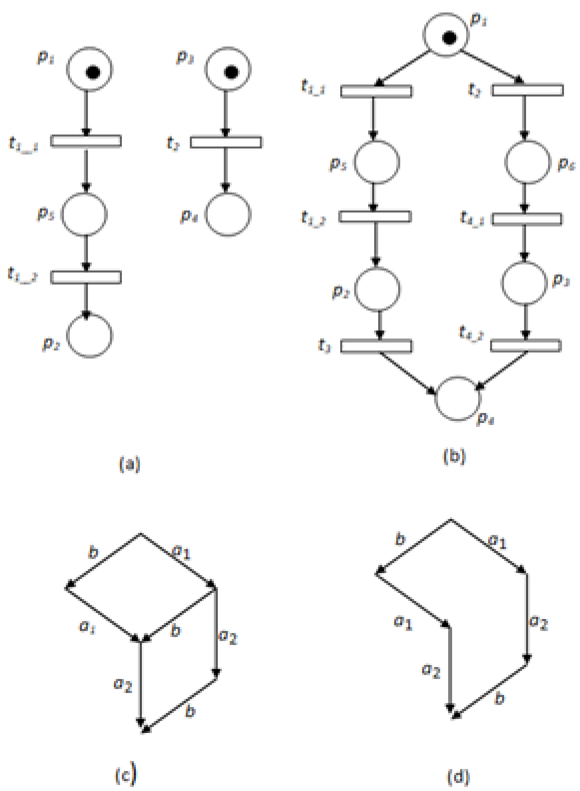

Fig. 3. No structural atomicity of actions.

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 [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16]. Considering such semantics allows a hierarchical design of the systems by refining actions (actions are seen as abstract processes). Another interest of these semantics is the characterization of parallel executions of non-instantaneous actions.

Among these semantics, we can cite the maximality semantics. Which has been defined by Devillers and Vogler [10] [17]. In this context, maximality bissimulation relation has been defined and proved to be coarsest relation preserved by action refinement.

In underlying semantics models of Petri net and event structures, a system with infinite behaviour needs an infinite set of events, which makes the underlying structures interesting just for theoretical point of view [10] [17].

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 and Petri net with the hypothesis that actions are not necessary atomic, i.e. actions are abstractions of finite processes and may elapse on time [18] [1] [19] [2] [20]. The main interest of maximality labeled transition system model is that it can be implemented and used in verification.

To more show the interest of the maximality semantics, we take the same example of the figure 1 while applying the method of generating the MLTS for the Petri nets proposed in [1]. So we get from the start two completely different MLTS that exactly reflect the behaviour described by the Petri nets.

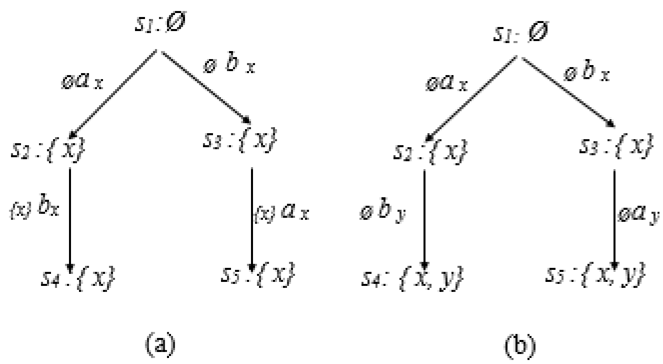

Fig. 4. Maximality semantics.

For the MLTS of figure 4.(a) actions a and b are executed sequentially a then b or b then a. For example for the first branch of this MLTS the start of execution of action a is identified by x, this action is executed independently of any other action, hence the association of the empty set to the cause of the transition $s_1 \xrightarrow{\emptyset a_x} s_2$. However the start of the execution of the action b from state s_2 is caused by the end of the execution of the action a hence the association of the set $\{x\}$ to the set of causes of the transition $s_2 \xrightarrow{\{x\} b_x} s_2$ and the event name x is re-used to identify the start of execution the action b. For the MLTS of figure 4.(b) actions a and b are clearly executed in parallel. The actions a and b are executed independently of any other action. The set $\{x, y\}$ in states s_4, s_5 indicated that there is two actions executed in parallel one is identified by x and the other by y.

In this paper we are interested by limited resources systems, while using the model of K-bounded Petri net as a modelling formalism, indeed the model of K-bounded Petri net is an intuitive model to represent the limitation of resources in a system. To deal with concurrency in the system behaviour we propose an operational maximality semantics that translates any K-bounded Petri net to maximality labeled transition systems.

The proposed semantics is concretized by the development of a software tool named MOS-KBPN for (Maximality Operational Semantics for K-Bounded Petri Net). Consequently we can take advantage of different results developed around the model of maximality labeled transition system.

Through a classic example of a limited buffer producer consumer paradigm, we show the interest of the proposed semantics in comparison to the interleaving semantics and the ST semantics. The comparison concerns the preservation of true concurrency and the reduction of the size of the semantics model. Furthermore, we will show that the properties of good behaviour of system can be verified on the corresponding maximality labeled transition system using our developed tool.

In addition, as we have mentioned to take advantage of the results developed around the model of maximality labeled transition system, we have applied on the fly reduction method to the maximality labeled transition system generated from a K-bounded Petri net which is proposed in [19] [22]. This method is based on the transitions aggregation, which contributes considerably to the reduction of the size of the semantics model.

II. 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 $a, b, ...$ 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 $\mathcal{H}$.

III. Maximality semantics for ordinary Petri net

In this section we recall the maximality approach for ordinary Petri net, proposed in [1]. Consider the example of the

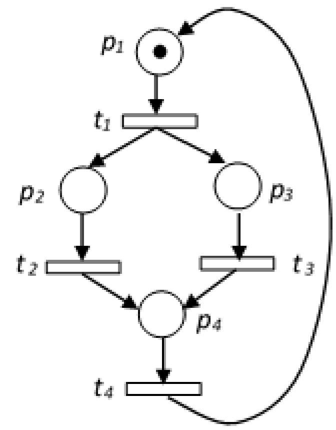

Fig. 5. Marked Petri net.

marked Petri net of Figure 5. After the firing of the transition t_1, it is evident that the execution of the transitions t_2 and t_3 are conditioned by the end of the action linked to the transition t_1. To capture this causal dependence between the firing 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 place p_2 and the token in place p_3. We can remark that, in the initial state, the token in p_1 is not bound to any transition, this token is said to be free in this state. In the case where the transition t_2 is fired, it could be deduced that the action associated with the firing of t_1 has finished. As a result, the token in p_3 will become free. Resulting marking after the firing of the transition t_2 is given in Figure 6.(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 noted by $\mathcal{FT}$, while bound tokens set will be noted by $\mathcal{BT}$. Each bound token identifies an action that is potentially in execution (this token is a maximal event). For example, in the configuration C_2 of Figure 6, the right part $\mathcal{BT}$ of the place p_2 contains a bound token of the firing $\emptyset t_1 x$, which means that $\mathcal{BT}_2 = \{(1, t_1, x)\}$.

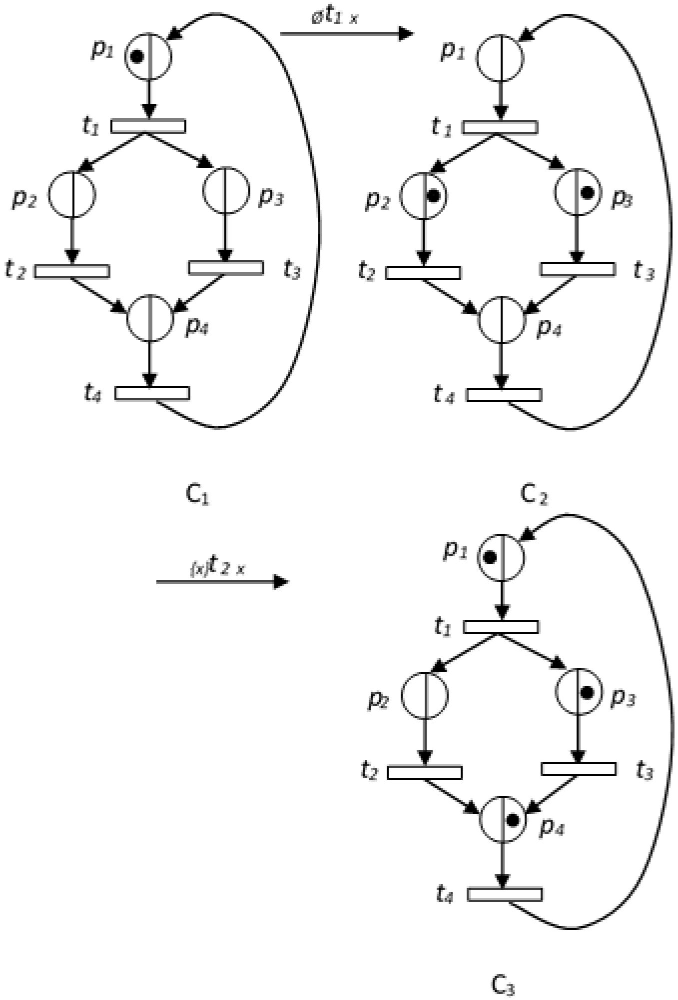

Fig. 6. Free tokens and bound tokens in a marking.

IV. Maximality Semantics for K-bounded Petri net

Through an example we explain the idea behind the proposed maximality semantics for K-bounded Petri nets. Let the Petri net of Figure 7. The tokens in p_1 are not bound to any transition, these tokens are said to be free (see Figure 8.(a)). In the case when the transition t_1 is launched, a bound token is produced in the place p_2.

By firing the transition t_1, we will obtain the marked Petri net of Figure 8.(b). From this marking, it can be seen that transition t_1 can be launched again. The firing of this transition will lead to the configuration of Figure 8.(c). From this configuration, the transition t_1 can not be fired again because the place p_2 is 2-bounded $(k = 2)$.

The maximality labeled transition system of Figure 9 corresponds to the petri net of Figure 7.

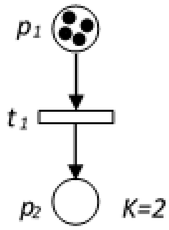

Fig. 7. Modelling of auto-concurrency.

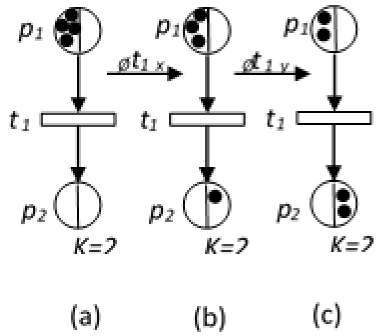

Fig. 8. Evolution of Petri net.

V. Operational maximality semantics for K-bounded Petri net

A. Preliminary definitions:

Definition 5.1: A K-bounded Petri net is a fivefold (P, T, W^-, W^+, K) 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.
- $K : P \longrightarrow \mathbb{N}^+$ is a function defining the limit capacity of places. $K(p) = k$ denotes the fact that the place p can't contain more than k tokens.

Definition 5.2: A labeled system $\Sigma = (P, T, W^-, W^+, K, \lambda)$ is a K-bounded Petri net in which all

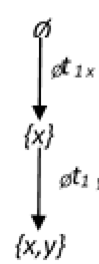

Fig. 9. MLTS in the case of parallelism.

transitions are labeled by actions such that $\lambda : T \longrightarrow L$ is a labeling function.

Definition 5.3: Let (P, T, W^-, W^+, K) be a K-bounded 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 $\mathcal{BT} = \{(n_1, t_1, x_1), ..., (n_m, t_m, x_m)\}$. The set of event names in p is given by a function $\delta^\bullet : P \longrightarrow 2^{\mathcal{H}}$, $\delta^\bullet(p) = \{x_1, x_2, .., x_m\}$.
- 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:
  $\delta : \{M : M\, a\, marking\, of\, the\, Petri\, net\} \longrightarrow 2^{\mathcal{H}}$ 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 :
  - $makefree(\{x_1, x_2, ..., x_m\}, M) = makefree(\{x_2, ..., x_m\}, makefree(\{x_1\}, M))$
  - $makefree(\{x\}, M) = M'$ 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 $\mathcal{BT} = \{(n_1, t_1, x_1), ..., (n_m, t_m, x_m)\}$.
- 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 $p \in P$. And $\mid M(p) \mid - W^-(p, t) + W^+(p, t) \leq k$ if p is K-bouned place $(K(p) = k)$. 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 K-bounded Petri net (P, T, W^-, W^+, K). $M_1 \subseteq M_2$ 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 K-bounded Petri net (P, T, W^-, W^+, K) 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).

B. Semantic rule

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

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

- $E = \delta(M_3), M_4 = makefree(E, M_1 - M_3)$
- 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:
  $\mathcal{BT}_2 = \mathcal{BT}_4 \cup \{(W^+(p, t), t, x) / W^+(p, t) \neq 0\}$
- $x = get(\mathcal{H} - (\delta(makefree(E, M_1))))$

VI. On the fly reduction method for maximality labeled transition system

In this section we recall the approach that generates an on-the-fly reduced MLTS modulo a maximality bisimulation relation proposed for ordinary Petri net in [19].

For explanation we consider the example of Figure 10. In the initial state (state s_1) of the maximality-based labeled transition system of Figure 10.(b), no action is running, from where the association of the empty set with this state. From state s_1, actions a and b can start their execution independently, their starts are respectively identified by event names x and y. a and b can be launched in any order. The set $\{x\}$ (resp. $\{y\}$) in state s_2 (resp. s_3) stipulates that the action a (resp. b) are potentially under execution in this state. $\{x, y\}$ in s_4 shows that actions a and b can be executed simultaneously.

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

Notice that from state s_2, transitions leading respectively to states s_4 and s_5 are due to the firing of the same transition t_2. In the first firing, the token of the initial marking is used whereas in the second firing, the used token is that produced by the firing of t_1. On the other hand, such as we noted above, the derivation by b leading to state s_4 is not conditioned by the end of the action a, while the derivation leading to state s_5 is conditioned by the end of a. In [19] it is clearly proved

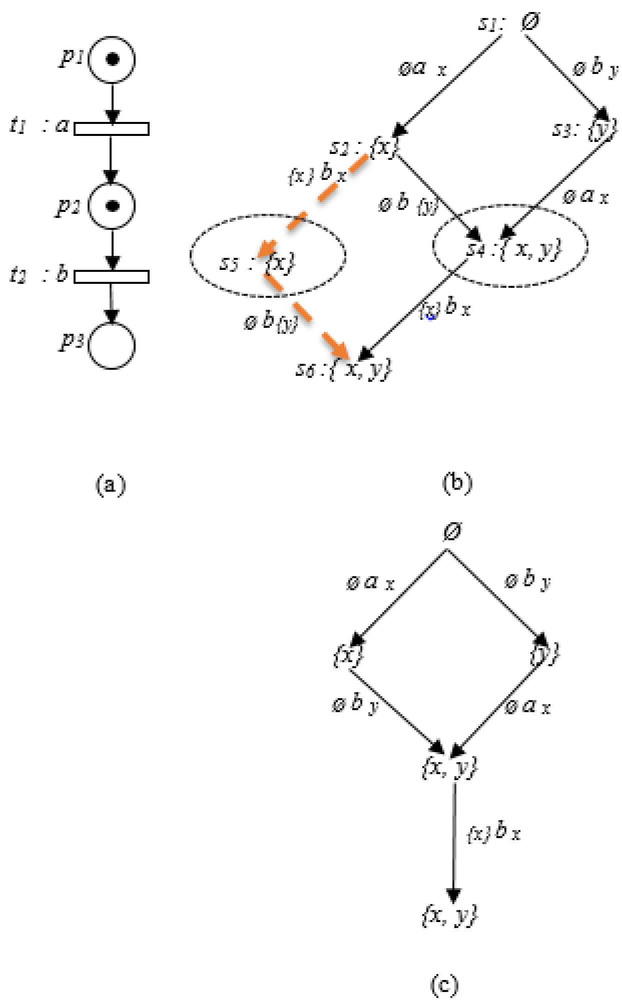

Fig. 10. Example MLTS reduction.

that states s_4 and s_5 are maximally bisimilar which means that is possible to omit the derivations $s_2 \longrightarrow s_5 \longrightarrow s_6$ in the maximality-based labeled transition system.
As we have previously mentioned, to take advantage of the results developed around the model of maximality labeled transition system, maximality bisimulation relations defined on maximality labeled transitions system for ordinary Petri net will be extended in this paper to K-bounded Petri net.

Definition 6.1: Let $Mark$ be a set of markings, T a set of transitions and $\longrightarrow$ a derivation relation between markings which is previously explained.

1) Let $\Re \subseteq 2^{Marq} \times 2^{Marq} \times \mathcal{F}$. the relation $\Re$ is sais maximality bisimulation relation according to a set of transitions of K-bounded Petri net bisimulation relation according to a set of transitions iff: $\forall \left(M_i, M_i^{'}, Id_{A_i}\right) \in \Re,\ A_i \subseteq \delta(M_i)$ et $A_i \subseteq \delta\left(M_i^{'}\right)$.
   a) If $M_i \xrightarrow{E_i t_i x} M_j$ then $\exists\ M_i^{'} \xrightarrow{E_i^{'} t_i y} M_j^{'}\ /\ \forall u \in A_i$ if $u \notin E_i$ then $u \notin E_i^{'}$ and for $z = get\left(\mathcal{M} - \left((\delta(M_i) - E_i) \cup \left(\delta\left(M_i^{'}\right) - E_i^{'}\right)\right)\right)$: $\left(M_j[z/x], M_j^{'}[z/y], Id_{A_{i+1}}\right) \in \Re/\ A_{i+1} = (A_i - E_i) \cup \{z\}$.
   b) If $M_i^{'} \xrightarrow{E_i^{'} t_i y} M_j^{'}$ then $\exists\ M_i \xrightarrow{E_i t_i x} M_j\ /\forall u \in A_i$ if $u \notin E_i^{'}$ then $u \notin E_i$ and for $z = get\left(\mathcal{M} - \left((\delta(M_i) - E_i) \cup \left(\delta\left(M_i^{'}\right) - E_i^{'}\right)\right)\right)$: $(M_j[z/x], M_j^{'}[z/y], Id_{A_{i+1}}) \in \Re/\ A_{i+1} = \left(A_i - E_i^{'}\right) \cup \{z\}$.
2) Let $\Sigma_1 = \left(P_1, T, W_1^-, W_1^+, K1, M_0^1, \lambda_1\right)$, $\Sigma_2 = \left(P_2, T, W_2^-, W_2^+, M_0^2, \lambda_2\right)$, two labeled systems with initial marking. Σ_1, Σ_2 are said to be maximally bisimilar according to T noted $\Sigma_1 \approx_m^T \Sigma_2$ if and only if there exists a maximality bisimulation relation $\Re$ according to Tsuch that $/ \left(M_0^1, M_0^2, \emptyset\right) \in \Re$.
3) $M_1 \approx_m^T /f\ \ M_2$ note that $(M_1, M_2, f) \in \Re$.

In this paper we will apply the same approach to reduce on the fly the maximality labled transition system generated from K-bounded Petri net. For this we keep the same operational semantics by modifying 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 $M' \subseteq M$ such that for any place p where $M(p) = (FT, BT)$, $M'(p)$ is defined as follows:

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

With: $\mathcal{BT}' \subseteq \mathcal{BT}$ and $\mid \mathcal{BT}' \mid = W^-(p,t) - \mathcal{FT}$

VII. Case study

Consider two processes, one called producer and the other consumer. The producer produces data and deposits them in the buffer. The consumer process take a produced data from the buffer and consumes them. When modelling the system, the buffer capacity will be represented by a K-bounded place. The modelling of this example in terms of K-bounded Petri net is given by the Figure 11.

For $k = 1$, the labeled transition system corresponding to the marking graph generated from this Petri net is depicted in Figure 12, it contains 8 states, 12 transitions. At this level we notice that this model is unable to model possible parallel execution of production and consumption operations.

The maximality labeled transition system which corresponds to this Petri net is depicted in Figure 13. Indeed in order to concretize our theoretical study we have

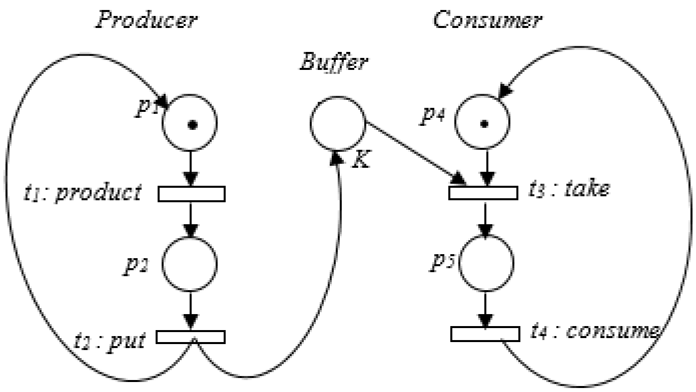

Fig. 11. Producer/consumer.

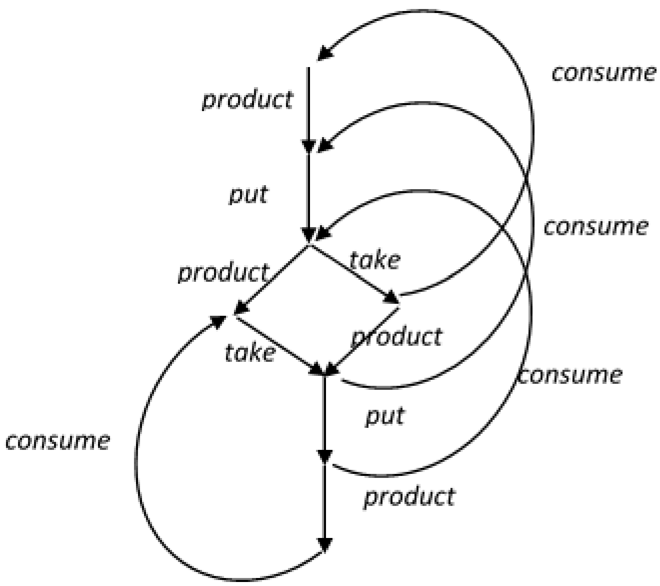

Fig. 12. Labeled transitions system for producer/ consumer.

developed a software tool named MOS-KPN for (Maximality Operational Semantics for K-Bounded Petri Net) this tool interprets any K-bounded Petri net to a MLTS. We have used this tool to generate the MLTS corresponding to the example of producer consumer. This MLTS contains 18 states and 26 transitions, but we can clearly see that the MLTS model represents parallelism and causality with reliability, for example in state s_5 the set $\{0, 1\}$ means that the actions product and take are under execution in this state. The properties of the good functioning of the system are specified in terms of CTL logic and verified using our developed tool.

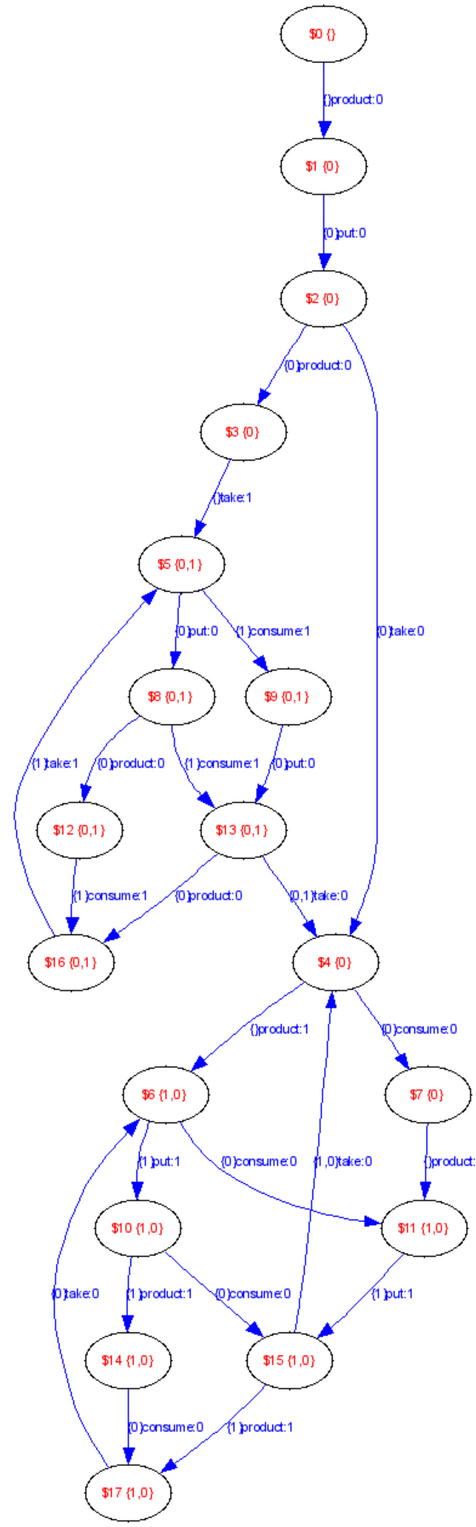

Fig. 13. Maximality labeled transitions system for producer/consumer.

A. Verification

- Safety properties:

- If the buffer is full the producer produced or waiting $AGput \Rightarrow EX(product\, or\, (not(put)))$
- If the buffer is empty then the consumer waiting or consuming. $AG\, take \Rightarrow EX(consume\, or\, (not(take)))$

• Liveness properties :
- If the producer can produce then it produces. $AG\, take \Rightarrow EX\, product$
- If the consumer can consume then it consumes. $AG\, put \Rightarrow EX(take\, or\, consume)$

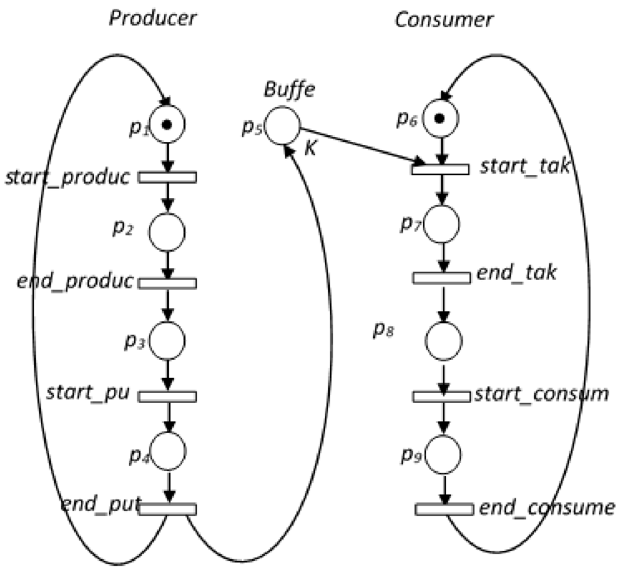

Fig. 14. Petri net with producer/consumer after refinement.

To capture the true concurrency under an interleaving semantics each transition may be splitted into two sequential actions, the start and the end actions like in the ST semantics. So, we consider now the Petri net of Figure 14 and we vary the capacity of buffer k. Then we compare the results obtained with the MLTS. We find that with the labeled transition system (LTS) the number of states and transition is very greater that of MLTS. In this case the reader can understand that the MLTS model represents causality and true parallelism with simplicity and reliability but with a minimum number of states. The obtained results are summarized in Table I.

Now we applied the reduction method proposed in [19] [22] to generate the MLTS for the Petri net of Figure 14 which contributes more to the reduction of the size of the semantics model. All with the change of the number of producers and consumers. The obtained results are summarized in Table II.

VIII. Conclusion

In this paper we have proposed an operational method for the generating of maximality labeled transition system associated to K-bounded Petri net. Noting that the K-bounded Petri net model is the most appropriate for modelling systems with limited resources. Consequently, the properties relating to the good functioning of a system specified by a K-bounded Petri net can be verified on its corresponding maximality labeled transition system. On the other hand, the structure of the maximality labeled transition system integrates information about the parallel execution of actions. This structure allows us to express more easily the properties related to the parallel execution of actions. At the end, we have applied the obtained results to the example of the producer / consumer with limited buffer capacity. Furthermore we have extended the on the fly reduction method for MLTS proposed in [19] [22] to K-bounded Petri net which contribute to the reduction of the number of states and transitions. Through this example we have shown the interest of our approach for the modeling and verifying concurrent systems with limited resources. This result may be extended to the work about recursive Petri net presented in [21] [22] [24].

TABLE I
Number of states and transitions of LTS and MLTS

Buffer	**LTS**		**MLTS**		**reduction rate**	
k	$N°s$	$N°T$	$N°S$	$N°T$	$s\%$	$T\%$
1	125	285	18	26	$85,60\%$	$90,74\%$
2	249	621	30	51	$87,95\%$	$91,78\%$
3	433	1161	42	76	$90,30\%$	$93,45\%$
4	693	1967	54	101	$92,20\%$	$94,86\%$
5	1048	3115	66	126	$93,70\%$	$95,95\%$
6	1548	4690	78	151	$94,86\%$	$96,78\%$
7	2126	6787	90	176	$95,76\%$	$97,40\%$
8	2896	9510	102	201	$96,47\%$	$97,88\%$
9	3855	12973	114	226	$97,04\%$	$98,25\%$
10	5031	17299	126	251	$97,49\%$	$98,54\%$

TABLE II
Number of states and transitions of MLTS before and after reduction for K=10

$N°$	$N°$	MLTS before		MLTS after		reduction rate	
P	C	$N°s$	$N°T$	$N°S$	$N°T$	$s\%$	$T\%$
1	1	126	251	108	206	$14,28\%$	$19,92\%$
2	1	448	1322	297	844	$33,70\%$	$36,15\%$
2	2	1930	7612	1321	4855	$31,55\%$	$36,21\%$
3	1	1342	5102	757	2822	$43,59\%$	$44,68\%$
3	2	7600	36751	4193	19187	$44,82\%$	$47,79\%$
3	3	31986	31986	17836	17836	$44,23\%$	$48,43\%$
4	1	3920	18166	1861	8456	$52,52\%$	$53,45\%$
4	2	28178	159320	12431	66944	$55,88\%$	$57,98\%$
5	1	11416	61898	4511	23750	$60,48\%$	$57,98\%$
6	1	33150	204226	10931	64180	$67,02\%$	$68,57\%$

References

[1] D.E. Saidouni, N. Belala, and M. Bouneb. Using maximality- based labeled transitions as model for petri nets. In The International Arab Conference on Information Technology(ACIT).2008a.

[2] D. E. Saidouni, N Belala, and M. Bouneb. Using maximality- based labeled transitions as model for petri nets. The International Arab Journal of Information Technology (IAJIT),2009a. 6(5):441–446.

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

[4] P. Darondeau and P. Degano. Causal trees. in ICALP'89, LNCS, Springer-Verlag,1989. 372:234–248.

[5] R. J. Van Glabbeek. The refinement theorem for st-bisimulation semantics. in IFIP Working Conference on Programming Concepts and Methods, North-Holland.1990.
[6] L. Aceto and M. Hennessy. Adding action refinement to finite process algebra. in ICALP91, LNCS Springer-Verlag,1991. Pages 506–519.
[7] E Best, R Devillers, A Kiehn, and L. Pomello. Concurrent bisimulations in petri nets. Acta Informatica,1991. 28:231–264.
[8] P. Degano and R. Gorrieri. Atomic refinement in process description languages. in A. Tarlecki, ed., Mathematical Foundations of Computer Science, LNCS, Springer-Verlag,1991. 520:121–130.
[9] W Janssen, M. Poel, and J. Zwiers. Action systems and action refinement in the development of parallel systems. CONCUR91, LNCS, Springer-Verlag,1991. 527:298–316.
[10] R. Devillers. Maximality preservation and the st-idea for action refinement. in G. Rozenberg, ed., Advances in Petri Nets, LNCS Springer-Verlag, 1992. 609:108–151.
[11] D. Andrews, J Groote, and C. Middelburg. Workshop on semantics of specification languages. Workshops in Computing, Springer-Verlag,1993. Pages 289–303.
[12] D. Hogrefe and S. Leue. IFIP TC6/WG6.1, 7th Int. Conf. on Formal Description Techniques (FORTE94), Chapman , Hall,1994. Pages 293–308.
[13] J.P. Courtiat and D.E. Saidouni. Action refinement in lotos. in A. Danthine, G. Leduc and P. Wolpe, eds, Protocol Specification, Testing and Verification (PSTV93), NorthHolland,1994a. Pages 341–354.
[14] D.E. Saidouni and J.P. Courtiat. Syntactic action refinement in presence of multiway synchronization. Semantics of Specification Languages,1994b. Pages 289–303.
[15] Gorrieri R and Laneve C. Split and st bisimulation semantics.1995. Pages 272-288.
[16] Gruska D.P. Bounded concurrency.1997.
[17] W. Vogler. Bisimulation and action refinement. TCS.1993.
[18] D.E. Saidouni. Maximality semantic: Application to actions refinement in LOTOS. PhD thesis, PhD thesis., LAAS-CNRS, 7 av. du Colonel Roche, 31077 Toulouse Cedex France.1996.
[19] D. E Saidouni, N Belala, and M. Bouneb. Aggregation of transitions in marking graph generation based on maximality semantics for petri nets. In Verification and Evalution of Computer and Communication systems (VECOS). 2008b.
[20] D. E Saidouni, N Belala, and M Bouneb. Maximality-based structural operational semantics for petri nets. In 2nd Mediterranean Conference on Intelligent Systems and Automation (CISA).2009b.
[21] D. E. Saidouni, M. Bouneb, J. M. Ilie. Maximality Semantic For Recursive Petri Net. European Council for Modelling and Simulation (ECMS) :2013. Pages 544-550
[22] M Bouneb, D. E Saidouni, and J. M. Ilie. A reduced maximality labeled transition system generation for recursive petri nets. In Journal of formal aspect of computing, Springer,November 2015. Number Volume 27, Issue 56, pp 951-973.
[23] M. Bouneb, D. E. Saidouni. Hierarchical Design Method for Multi-Agent Systems. International Journal of Agent Technologies and Systems (IJATS).2015.IGI Global 7(2): pages 105-134.
[24] M. Bouneb, D. E. Saidouni, J. M. Ilie. Hierarchical System Design Using Refinable Recursive Petri Net. Journal of Computing and Informatics.2018. 37(3):pages 635-655.