Theory and Engineering of Complex Systems (Advances in Intelligent Systems and Computing 365)
10-th Int. Conf. on Dependability of Complex Systems, June 29 – July 3, 2015, Brunow, Poland, pp.593-602.

DOI 10.1007/978-3-319-19216-1_57.

Symbolic analysis of timed Petri nets

Włodek M. Zuberek

Department of Computer Science, Memorial University,
St.John's, NL, Canada A1B 3X5
email: wlodek@mun.ca

Abstract. In timed Petri nets temporal properties are associated with transitions as transition firing times (or occurrence times). Specific properties of timed nets, such as boundedness or absence of deadlocks, can depend upon temporal properties and sometimes even a small change of these properties has a significant effect on the net's behavior (e.g., a bounded net becomes unbounded or vice versa). The objective of symbolic analysis of timed nets is to provide information about the net's behavior which is independent of specific temporal properties, i.e., which describes properties of the whole class of timed nets with the same structure.
Keywords: timed Petri nets, symbolic analysis, boundedness, absence of deadlocks, producer–consumer model.

1 Introduction

Petri nets are formal models of systems which exhibit concurrent activities [11], [10], [7], [4]. Communication networks, multiprocessor systems, manufacturing systems and distributed databases are simple examples of such systems. As formal models, Petri nets are bipartite directed graphs, in which the two types of vertices represent, in a very general sense, conditions and events. An event can occur only when all conditions associated with it (represented by arcs directed to the event) are satisfied. An occurrence of an event usually satisfies some other conditions, indicated by arcs directed from the event. So, an occurrence of one event causes some other event to occur, and so on.

In order to study performance aspects of systems modeled by Petri nets, the durations of modeled activities must also be taken into account. This can be done in different ways, resulting in different types of temporal nets [2], [3], [14], [8]. In timed Petri nets [17], firing times or occurrence times are associated with events, and the events occur in real–time (as opposed to instantaneous occurrences in other models [1]). For timed nets, the state graphs of nets are Markov chains (or embedded Markov chains), so the stationary probabilities of states can be determined by standard methods [13], [5]. Stationary probabilities are used for the derivation of many performance characteristics of the model [12].

In timed nets, all firings of enabled transitions are initiated in the same instants of time in which the transitions become enabled. If, during the firing period of a transition, the transition becomes enabled again, a new, independent firing can be initiated, which will overlap with the other firing(s). There is no limit on the number of simultaneous firings of the same transition (sometimes this is called "infinite firing semantics").

The firing times of transitions can be either deterministic or stochastic (i.e., described by a probability distribution function); in the first case, the corresponding timed nets are referred to as D–nets [15], in the second, for the (negative) exponential distribution of firing times, the nets are referred to as M–nets (Markovian nets) [16]. In both cases, the concepts of states and state transitions have been formally defined and used in the derivation of different performance characteristics of the models [15], [16], [17].

In Petri nets with deterministic firing times, the properties such as boundedness or absence of deadlocks can depend upon specific values of firing times and sometimes even a small changes of firing times can have a significant effect on the behavior of a net (e.g., a bounded net becomes unbounded or a deadlock is created).

This paper proposes symbolic analysis of timed Petri nets which analyzes the behavior for the whole spectrum of temporal properties, so the results do not depend upon specific temporal properties.

Section 2 recalls a few basic concepts of Petri nets and timed Petri nets. Section 3 introduces symbolic analysis while an illustrative example is presented in Section 4. Several concluding remarks are in Section 5.

2 Petri nets and timed Petri nets

Place/transition Petri nets are bipartite directed graphs in which the two types of vertices are called places and transitions. Place/transition nets are also known as condition/event systems.

A Petri net (sometimes also called net structure) $\mathcal{N}$ is a triple $\mathcal{N} = (P, T, A)$ where:

- P is a finite set of places (which represent conditions);
- T is a finite set of transitions (which represent events), $P \cap T = \emptyset$;
- A is a set of directed arcs which connect places with transitions and transitions with places, $A \subseteq P \times T \cup T \times P$, also called the flow relation or causality relation (and sometimes represented in two parts, a subset of $P \times T$ and a subset of $T \times P$).

For each transition $t \in T$, and each place $p \in P$, the input and output sets are defined as follows:

$$\begin{aligned}
Inp(t) &= \{p \in P \mid (p,t) \in A\},\\
Inp(p) &= \{t \in T \mid (t,p) \in A\},\\
Out(t) &= \{p \in P \mid (t,p) \in A\},\\
Out(p) &= \{t \in T \mid (p,t) \in A\}.
\end{aligned}$$

The dynamic behavior of nets is represented by markings, which assign nonnegative numbers of tokens to the places of a net. Under certain conditions these tokens can "move" in the net, changing one marking into another.

A marked Petri net $\mathcal{M}$ is a pair $\mathcal{M} = (\mathcal{N}, m_0)$, where:

- $\mathcal{N}$ is a net structure, $\mathcal{N} = (P, T, A)$;
- m_0 is the initial marking function, $m_0 : P \to \{0, 1, ...\}$ which assigns a nonnegative number of tokens to each place of the net.

Marked nets are also equivalently defined as $\mathcal{M} = (P, T, A, m_0)$.

In a marked net $\mathcal{M}$, a transition t is enabled by a marking m iff:

$$\forall p \in \text{Inp}(t) : m(p) > 0.$$

An enabled transition t can fire (or occur) transforming a marking m into a directly reachable marking m':

$$\forall p \in P : m'(p) = \begin{cases} m(p) - 1, & \text{if } p \in \text{Inp}(t) - \text{Out}(t), \\ m(p) + 1, & \text{if } p \in \text{Out}(t) - \text{Inp}(t), \\ m(p), & \text{otherwise.} \end{cases}$$

A timed Petri net $\mathcal{T}$ is a pair, $\mathcal{T} = (\mathcal{M}, f)$ where:

- $\mathcal{M}$ is a marked net, $\mathcal{M} = (\mathcal{N}, m_0)$;
- f is the firing–time function, $f : T \to \mathbf{R}^+$, which assigns the (average) firing times (or occurrence times) to transitions of the net.

For performance analysis of timed nets, an additional component is needed to describe random decisions in (nondeterministic) nets. Usually it is a conflict–resolution function, $c : T \to [0, 1]$, which assigns the probabilities of firings to transitions in free–choice classes of transitions, and relative frequencies of firings to transitions in conflict classes [16], [17]. This function c is not needed for symbolic analysis.

3 Symbolic analysis

For symbolic analysis, only relations between the firing times of transitions are needed, so the state descriptions can be simpler than for the detailed behavioral analysis [16]. The states can be represented by pairs of functions, current firing function $n : T \to \{0, 1, ...\}$ and current (residual) marking function $m : P \to \{0, 1, ...\}$.

For each state $s = (n, m)$, the next states correspond to all possible relations between the durations of currently firing transitions. This is described by all (nonempty) subsets of transitions which finish their firings (and initiate new firings if any transitions become enabled):

$$\text{next}(s) = \bigcup_{T_i \subseteq T_f(s)} \text{Next}(s, T_i)$$

where $T_f(s)$ (or equivalently $T_f(n,m)$ as $s = (n,m)$) is the set of transitions which are firing in state s, i.e., transitions with nonzero entries in n:

$$T_f(n,m) = \{t \in T \mid n(t) > 0\},$$

and $\text{Next}(s, T_i)$ is the set of states which can be reached from s by ending the firings of all transitions in T_i (and then initiating all possible firings).

Finding the set $\text{Next}(s, T_i)$ is done in two steps:

1. Terminating the firings of all transitions in T_i, which creates an intermediate state $s' = (n', m'$ where:

$\forall t \in T : n'(t) = \begin{cases} n(t) - 1, & \text{if } t \in T_i; \\ n(t), & \text{otherwise}; \end{cases}$

$\forall p \in P : m'(p) = m(p) + \sum_{t \in Inp(p) \cap T_i} n(t).$

2. Initiating new firings of transitions which are enabled by $m'i$. These new firings can be described by a set of functions $b_j : T \to \{0, 1, ...\}$ such that:

- $\forall p \in P : m'(p) - \sum_{t \in \text{Out}(p)} b_j(t) \geq 0$, and
- $\forall t \in T\ \exists p \in \text{Inp}(t) : m'(p) - \sum_{t \in \text{Out}(p)} b_j(t) = 0.$

These two conditions guarantee that all transitions which can fire, initiate their firings (free–choice nets and nets with conflicts have more than one function b_j).

The set of states reachable from $s = (n,m)$ by a set of transitions T_i is described by procedure $Next(n, m, T_i)$ which first finds the intermediate state (n', m') and then uses a recursive function *Find* to find all possible states by firing transitions enabled by m' (and adjusting n' accordingly):

```
proc Next(n[1:k], m[1:ℓ], T₀);
begin
      n' := n;
      m' := m;
      for each tᵢ ∈ T₀ do
            n'[i] := n' − 1;
            for each pⱼ ∈ Inp(tᵢ) do m'[j] := m'[j] + 1 od
      od;
      New := ∅;
      Find(n'ᵢ, m'ᵢ);
      States := States ∪ New
      od
end
```

States is a global variable which stores the set of reachable states.

Recursive function *Find(n,m)* finds all states which are derived from $s = (n,m)$ by initiating the firings of enabled transitions. It is assumed that n and

m are the firing and marking functions, respectively, represented by k-element and ℓ-element vectors (k is the number of transitions and ℓ is the number of places); moreover, *Find* uses a nonlocal set variable *New*:

proc $Find(n[1:k], m[1:\ell])$;
begin
  $E := \emptyset$;
  for each $t_i \in T$ **do**
    $check :=$ true;
    for each $p_j \in \text{Inp}(t_i)$ **do**
      if $m[j] = 0$ **then** $check :=$ false **fi**
    od;
    if $check$ **then** $E := E \cup \{t_i\}$ **fi**
  od;
  if $E = \emptyset$ **then** $New := New \cup \{(n, m)\}$
  else
    for each $t_i \in E$ **do**
      $n' := n$;
      $n'[i] := n[i] + 1$;
      $m' := m$;
      for each $p_j \in \text{Inp}(t_i)$ **do** $m'[j] := m[p_j] - 1$ **od**;
      $Find(n', m')$
    od
  fi
end

The initial state (or states) of a marked net is (or are) determined by an invocation Find(n_0, m_0), where n_0 is zero for all $t \in T$, while m_0 is the initial marking function.

The procedures are shown as a simple illustration of the approach; they can be improved in many ways.

4 Example

A timed Petri net model of a producer–consumer system with an unbouded buffer is shown in Fig.1.

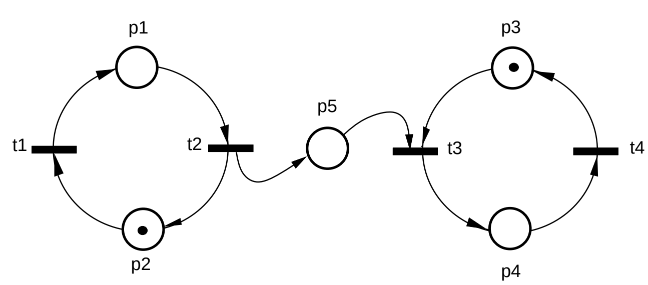

Fig.1. Model of a producer–consumer system with an unbounded buffer.

The two cyclic subnets, (t_1, p_1, t_2, p_2) and (p_3, t_2, p_4, t_4), represent the producer and the consumer, respectively, while place p_5 is the buffer. It is known that for some values of firing times the behavior of this system is finite while for others the model becomes unbounded. For example, for $f(t_1) = 3.0$, $f(t_2) = f(t_3) = 0.5$, $f(t_4) = 2.0$, the state transition graph is shown in Fig.2.

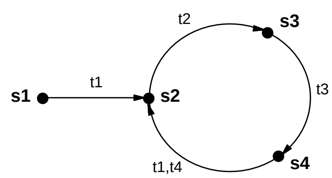

Fig.2. State transition graph for the net shown in Fig.1 with $f(t_1) = 3.0$, $f(t_2) = f(t_3) = 0.5$, $f(t_4) = 2.0$.

Symbolic analysis of the model shown in Fig.1 is presented in Tab.1 where s_i is the current state, n_i and m_i are the two components of s_i, T_i is the set of transitions which terminate their firings in state s_i, b_j is the function describing new firings and s_j is the next state.

It can be traced in Tab.1 that s_{11} is reached from s_{10} by t_2 and that s_{10} is reached from s_8 by t_1. The states s_8 and s_{11} are identical except of marking of p_5. Consequently, if t_1 and t_2 can fire several times before t_3 and t_4 fire, the marking of p_5 can increase arbitrarily, so the model is unbounded.

Similarly, s_{12} is reached from s_9 by t_2, and s_9 is reached from s_7 by t_1.

A timed net is unbounded is there are two states, $s_i = (n_i, m_i)$ and $s_j(n_j, m_j)$ such that s_j is reachable from s_i and s_i is reachable from an initial state, and s_j is componentwise greater or equal to s_i, i.e., for all values of k and ℓ, $n_j[k] \geq n_i[k]$ and $m_j[\ell] \geq m_i[\ell]$.

Moreover, a timed net contains a deadlock if there is a state s_i reachable from an initial state, for which the set of next states is empty.

The conditions for unboundedness and deadlock can be easily recognized during symbolic analysis.

The part of the state transition diagram described in Tab.1 is shown in Fig.3. The regular structure of state transitions can be systematically extended which indicates the unboundedness of the model. Moreover, the behavior shown in Fig.2 can be easily traced in Fig.3 following the same transitions involved in the changes of states:

$$t_1 \to t_2 \to t_3 \to t_1, t_4 \to t_2 \to t_3 \to t_1, t_4 \to \cdots$$

so the corresponding state transitions (in Fig.3) are:

$$s_1 \to s_2 \to s_3 \to s_5 \to s_2 \to s_3 \to \cdots$$

Table 1. Symbolic analysis of the producer–consumer model.

s_i	n_i 1 2 3 4	m_i 1 2 3 4 5	T_i	b_j 1 2 3 4	s_j
s_1	1 0 0 0	0 0 1 0 0	t_1	0 1 0 0	s_2
s_2	0 1 0 0	0 0 1 0 0	t_2	1 0 1 0	s_3
s_3	1 0 1 0	0 0 0 0 0	t_1	0 1 0 0	s_4
			t_3	0 0 0 1	s_5
			t_1, t_3	0 1 0 1	s_6
s_4	0 1 1 0	0 0 0 0 0	t_2	1 0 0 0	s_7
			t_3	0 0 0 1	s_6
			t_2, t_3	1 0 0 1	s_8
s_5	1 0 0 1	0 0 0 0 0	t_1	0 1 0 0	s_6
			t_4	0 0 0 0	s_1
			t_1, t_4	0 1 0 0	s_2
s_6	0 1 0 1	0 0 0 0 0	t_2	1 0 0 0	s_9
			t_4	0 0 0 0	s_2
			t_2, t_4	1 0 1 0	s_3
s_7	1 0 1 0	0 0 0 0 1	t_1	0 1 0 0	s_9
			t_3	0 0 0 1	s_8
			t_1, t_3	0 1 0 1	s_{10}
s_8	1 0 0 1	0 0 0 0 1	t_1	0 1 0 0	s_{10}
			t_4	0 0 1 0	s_3
			t_1, t_4	0 1 1 0	s_4
s_9	0 1 1 0	0 0 0 0 1	t_2	1 0 0 0	s_{12}
			t_3	0 0 0 1	s_{10}
			t_2, t_3	1 0 0 1	s_{11}
s_{10}	0 1 0 1	0 0 0 0 1	t_2	1 0 0 0	s_{11}
			t_4	0 0 1 0	s_4
			t_2, t_4	1 0 1 0	s_7
s_{11}	1 0 0 1	0 0 0 0 2	...		
s_{12}	1 0 1 0	0 0 0 0 2	...		

Structural analysis [18] of the net shown in Fig.1 provides a simple condition for unboundedness for this particular net:

$$f(t_1) + f(t_2) \leq f(t_3) + f(t_4).$$

This condition is clearly not satisfied when $f(t_1) = 1.5$, $f(t_2) = f(t_3) = 0.5$ and $f(t_4) = 3.5$), so net's unbounded behavior is expected in this case. Indeed, the sequence of transitions involved in consecutive state changes is:

$$t_1 \to t_2 \to t_3 \to t_1 \to t_2 \to t_1 \to t_2, t_4 \to t_3 \to t_1 \to t_2 \to t_1 \to t_2, t_4 \to t_3 \cdots$$

with the pattern:

$$t_1 \to t_2 \to t_1 \to t_2, t_4 \to t_3$$

repeated. The sequence of (symbolic) state changes, shown in Fig.4, is more convoluted than in the bounded case:

$$s_1 \to s_2 \to s_3 \to s_5 \to s_6 \to s_8 \to s_{10} \to s_7 \to s_8 \to s_{10} \to s_{11} \to \cdots$$

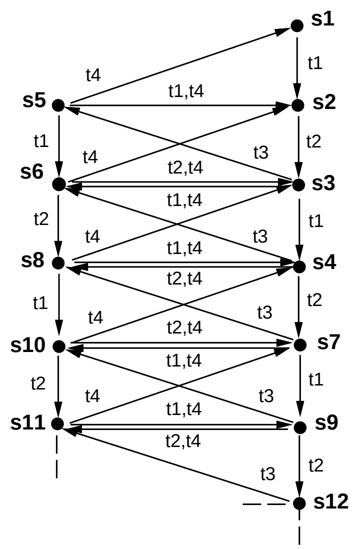

Fig.3. State transition graph for symbolic analysis of net in Fig.1.

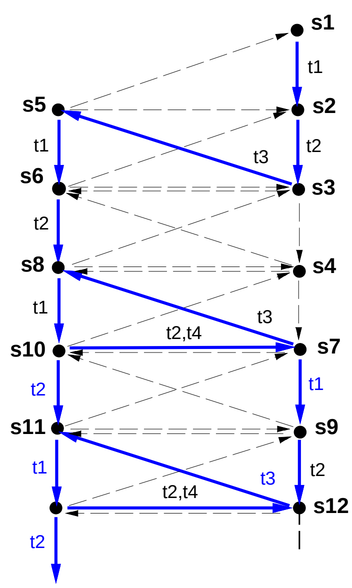

Fig.4. State transitions for net in Fig.1 with
with $f(t_1) = 1.5$, $f(t_2) = f(t_3) = 0.5$, $f(t_4) = 3.5$.

5 Concluding remarks

The behavior of a timed Petri net depends upon the specific values of temporal parameters associated with transitions of a net, and can change in a significant way for even a small changes of these temporal parameters. Symbolic analysis provides general information about the behavior of all nets with the same structure. For example, if symbolic analysis creates a finite space of (symbolic) states, no temporal parameters can result in unbounded behavior. Similarly, if symbolic analysis indicates deadlock freeness, no temporal parameters can create a deadlock in the net.

For large models, symbolic analysis can be quite complex. Therefore analysis of real–life applications is not feasible without efficient software tools. It is expected that such tools will be added to existing software packages for analysis of timed Petri net models.

Symbolic analysis presented in this paper is similar to reachability analysis of marked nets [8], [17]. The obvious difference is that reachability analysis does not consider simultaneous multiple firings. The effects of this difference need to be carefully explored.

Fig.5. shows the initial part of the marking graph for the net in Fig.1.

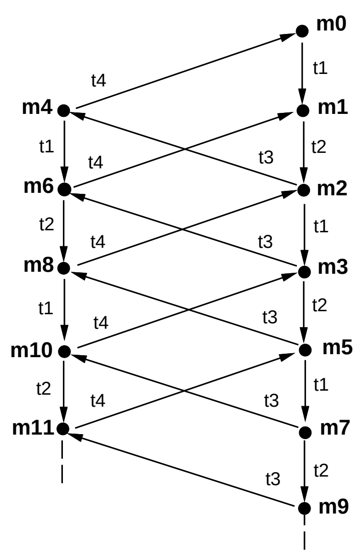

Fig.5. Marking graph for the net in Fig.1.

The similarity of Fig.3 and Fig.5 can be misleading because there is no straightforward correspondence between the markings (Fig.5) and the states (Fig.3). In particular, there are no changes due to multiple transitions in Fig.5 (like t_1, t_4 or t_2, t_4 in Fig.3).

It is believed that symbolic analysis presented in this paper can be extended to other classes of Petri nets, such as inhibitor Petri nets or high–level Petri nets [6].

Acknowledgement

The Natural Sciences and Engineering Research Council of Canada partially supported this research through grant RGPIN-8222.

References

1. Ajmone Marsan, M., Conte, G., Balbo, G., “A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems”. ACM Trans. on Computer Systems, vol.2, no.2, pp.93–122, 1984.
2. Ajmone Marsan, M., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G.: **Modeling with generalized stochastic Petri nets**. Wiley and Sons 1995.
3. Bause, F., Kritzinger, P.S., **Stochastic Petri nets – and introduction to the theory** (2 ed.). Vieweg Verlag 2002.
4. Girault, C., Valk, R., **Petri nets for systems engineering**. Springer-Verlag 2002.
5. Haggstrom, O.: **Finite Markov chains and algorithmic applications**. Cambridge Univ. Press 2003.
6. He, X., Murata, T.: “High-level Petri nets - extensions, analysis and applications”; in: **The Electrical Engineering Handbook**. pp.459-475, Academic Press 2007.
7. Murata, T.: Petri nets: properties, analysis, and applications. Proceedings of the IEEE, vol.77, no.4, pp.541-580 1989,
8. Popova-Zeugmann, L.: **Time and Pertri nets**. Springer-Verlag 2013.
9. Proth, J.M., Xie, X., **Petri nets**. Wiley & Sons 1996.
10. Reisig, W.: **Petri nets – an introduction** (EATCS Monographs on Theoretical Computer Science 4). Springer-Verlag 1995,
11. Reisig, W.: **Understanding Petri nets – modeling techniques, analysis methods, case studies**. Springer-Verlag, 2013.
12. Robertazzi, T.A.: **Computer networks and systems: queueing theory and performance evaluation** (3 ed.). Springer-Verlag 2000.
13. Stewart, W.J., **Introduction to the numerical solution of Markov chains**. Princeton University Press 1994.
14. Wang, J., **Timed Petri nets**. Kluwer Academic Publ. 1998.
15. Zuberek, W.M., “M–timed Petri nets, priorities, preemptions, and performance evaluation of systems”. In: “Advances in Petri Nets 1985” (Lecture Notes in Computer Science 222), pp.478-498, Springer–Verlag 1986.
16. Zuberek, W.M., “D–timed Petri nets and modelling of timeouts and protocols”, Transactions of the Society for Computer Simulation, vol.4, no.4, pp.331-357, 1987.
17. Zuberek, W.M., “Timed Petri nets – definitions, properties and applications”. Microelectronics and Reliability (Special Issue on Petri Nets and Related Graph Models), vol.31, no.4, pp.627–644, 1991.
18. Zuberek, W.M., “Structural methods in performance analysis of discrete-event systems”; Proc. 9-th IEEE Int. Conf. on Methods and Models in Automation and Robotics, pp.878-882, 2003.