Model Checking Time Petri Nets using NuSMV

Andrea Bobbio[1], András Horváth[2]

[1]DISTA, Università del Piemonte Orientale, Alessandria, Italy

[2]Dipartimento di Informatica, Università di Torino, Italy

bobbio@di.unito.it, *horvath@di.unito.it*

Abstract

Time Petri Nets (TPN) are Petri Net based models augmented with timing information by associating an interval to each transition. The firing time of a transition is chosen non-deterministically (but not in a stochastic manner) from its associated interval. This paper presents a technique to check if a TPN satisfies temporal properties expressed in real-time Computational Tree Logic (RTCTL). The transition graph of the TPN is built in a compositional manner based on discretization of the firing intervals. The compositional description can be automatically translated into the model description language of NuSMV, a tool for model checking finite state systems against specifications in RTCTL.

1 Introduction

TPNs are devoted to specify and verify properties of systems where timing is a critical parameter that may affect the behavior. In this line of research [5], time is assigned as a constant value or as an interval defined by a min (*earliest firing time - EFT*) and a max (*latest firing time - LFT*) value. The firing semantics is interleaving and with non-determinism [1]. In [4] a modified firing semantics is introduced: time is assigned as intervals, and firing may be forced when the maximum time expires *(strong firing semantics)* or firing may be not mandatory when the maximum time expires *(weak firing semantics)*.

As shown in [1], the incidence matrix of the discretized state space of the TPN can be built in a compositional manner using Kronecker algebra. The global evolution of the process is described by Kronecker expressions containing the descriptors of Discrete Phase Type Timing (DPT) structures that describe the local evolution of the transitions. This technique takes into account the preemption policy of the transitions as well, i.e. the way in which the time that a transition spent enabled before its preemption is taken into account when it gets enabled again. Three policies are considered [2]. In the *preemptive repeat different (prd)* policy (also called *enabling memory*) the age variable is reset each time the transition is disabled or fires; in the *preemptive repeat identical (pri)* policy, when the transition is disabled its age variable is reset, but when the transition is enabled again an identical firing time must be completed. Finally, in the *preemptive resume (prs)* policy the age variable maintains its value when the transition is disabled and then re-enabled, and is reset only when the transition fires.

Discretization implies a state expansion and incurs in the state space explosion problems. The compositional approach via Kronecker algebra may alleviate this problem.

[1]No weight is assigned to the action of atomic firing inside the allowed interval or for resolving conflicts.

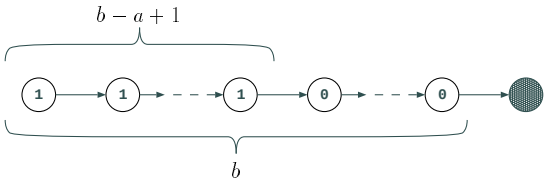

Figure 1: Firing interval $[a, b]$, strong time semantics

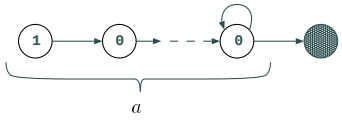

Figure 2: Firing interval $[a, \infty]$, weak or strong time semantics

Moreover, as we will show in this paper, the TPN model may be interfaced with model checking tools that use efficient storage techniques like Binary Decision Diagrams (BDD).

The paper is organized as follows. Section 2 introduces the Discrete Phase Type Timing structures that will be used to describe the local evolution of the transitions of the model. The global process is discussed in brief in Section 3. Section 4 describes the idea of model checking a TPN using NuSMV. Conclusions are drawn in Section 5.

2 Local evolution of transitions

This section gives an example of the DPT structures that describe how the local descriptor of a transition evolves in a step if the transition is enabled. The applied structures depend on the adopted memory policy as well. In this paper we deal only with prs and prd transitions in case of strong time semantics; other cases are discussed in [1].

The structure used to represent the local evolution of an enabled transition with strong time semantics and firing interval[2] $[a, b]$ is depicted in Figure 1. When the initial phase of the structure is chosen the process may enter any of the phases signed with 1. The arrows represent the possible state-jumps; having more than one outgoing arc from a phase indicates a non-deterministic choice. The transition fires if a state-jump to the filled state occurs. This structure ensures that the firing time of the transition will be in the interval $[a, b]$ and the transition fires for certain when it reaches the upper limit of its firing interval. The structure is represented by the row vector $\boldsymbol{t_0}$ that describes the possible initial phases, the square matrix $\boldsymbol{T}$ that describes the possible state-jumps and the column vector $\boldsymbol{t_f}$ that gives the phases from which firing may happen:

$$\boldsymbol{t_0} = [\underbrace{1, \ldots, 1}_{b-a+1}, \underbrace{0, \ldots, 0}_{a-1}], \ \boldsymbol{T} = \begin{bmatrix} 0 & 1 & 0 & \ldots & \\ 0 & 0 & 1 & 0 & \ldots \\ & & \ddots & & \\ 0 & & \ldots & 0 & 1 \\ 0 & & \ldots & & 0 \end{bmatrix}, \ \boldsymbol{t_f} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

In case of firing interval $[a, \infty]$ the structure depicted in Figure 2 is applied. Having been enabled for a time units the transition may either fire or remain in the last phase.

3 Global evolution

Here we give a very brief description on constructing the matrix that governs the evolution of the global model. The procedure, which is based on [6], is described in detail in [1].

The state space is partitioned into as many blocks as many extended markings there are in the model. Extended markings carry information not only on the marking but also on preemption situations and on "immediate candidate" transitions as well. Immediate candidates are the prs transitions that got disabled when reached their firing time and, since

[2] We assume that minimal and maximal firing times are integer values and the minimal firing time is strictly positive.

that, according to the definition of prs preemption policy, they have to fire immediately when get enabled again. Since transitions may fire simultaneously, an arc in the reachability graph of the extended markings may represent the firing of a set of transitions. In the matrix describing the global process, a Kronecker expression either corresponds to an arc of the extended reachability graph or describes the evolution of the global descriptor in the case when the process remains in the same extended marking from one step to another.

4 Description using NuSMV

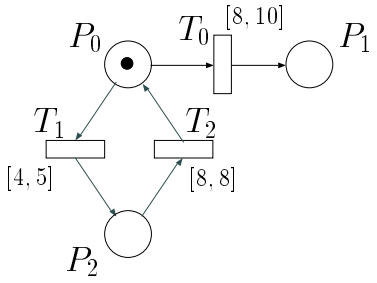

Figure 3: A simple TPN model

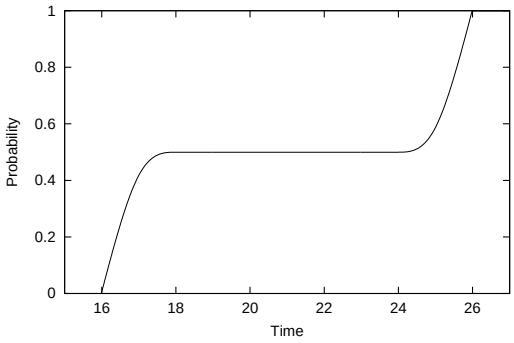

Figure 4: Probability of marking 2 versus time

In order to give an idea on how to describe the global evolution of the model using NuSMV (for information on NuSMV see [7]) we consider the simple TPN depicted in Figure 3. All transitions follow strong firing semantics; T_0 is of prs type, while T_1 and T_2 are prd type transitions. The incidence matrix of the global process is

$$\left[\begin{array}{c|c|c|c} \boldsymbol{T}^{(0)} \otimes \boldsymbol{T}^{(1)} & \boldsymbol{T}^{(0)} \otimes \boldsymbol{t_f}^{(1)} \otimes \boldsymbol{t_0}^{(2)} & \boldsymbol{t_f}^{(0)} \otimes (\boldsymbol{e}^{(1)} - \boldsymbol{t_f}^{(1)}) + \boldsymbol{t_f}^{(0)} \otimes \boldsymbol{t_f}^{(1)} & \boldsymbol{t_f}^{(0)} \otimes \boldsymbol{t_f}^{(1)} \otimes \boldsymbol{t_0}^{(2)} \\ \hline \boldsymbol{I}^{(0)} \otimes \boldsymbol{t_0}^{(1)} \otimes \boldsymbol{t_f}^{(2)} & \boldsymbol{I}^{(0)} \otimes \boldsymbol{T}^{(2)} & & \\ \hline & & 1 & \\ \hline & & \boldsymbol{t_f}^{(2)} & \boldsymbol{T}^{(2)} \end{array}\right] \quad (1)$$

We have 4 extended markings: token in P_0; token in P_2 and transition T_0 has memory; token in P_1; token in P_0 and transition T_0 is immediate candidate (it means that T_0 and T_1 reached their firing time in the same time instant, T_1 fired and, as a result, T_0 fires immediately when gets enabled again). The structures describing transition T_0, T_1 and T_2 have 10, 5, 8 phases, respectively. Consequently, the number of phases in the 1st extended marking is $10 \times 5 = 50$, in the 2nd it is $10 \times 8 = 80$, in the 3rd 1, and 8 in the 4th.

Using NuSMV the transition graph of the system can be described by defining the possible current state/successor state pairs by logic conditions depending on the variables of the system. We associate a variable to each transition. In addition, we have a variable to take the current marking into account (this is necessary if the set of enabled transitions is the same in different markings; otherwise it leads to a slightly redundant description like in the case of this example). A possible description in NuSMV, which can be easily generated automatically from the Kronecker expressions of the incidence matrix, is given in Table 1.

The variables of the model are listed under the second appearance of the keyword `VAR` with their possible values. The variable associated with transition T_0 is `t0`, it encodes the state of the transition the following way: if `t0`=0, then T_0 is not enabled,

```
MODULE main
VAR
  m: m1;
MODULE m1
VAR
  t0: -1..20;
  t1: 0..5;
  t2: 0..8;
  em: {-1,0,1,2};
INIT
  em=-1 & t0=0 & t1=0 & t2=0
TRANS
( (em=-1 & t0=0 & t1=0 & t2=0) &
  (next(em)=0 & (next(t0)=1|next(t0)=2|next(t0)=3) &
  (next(t1)=1|next(t1)=2) & next(t2)=0 ) ) |
( (em=0 & t0>0 & t0<10 & t1>0 & t1<5 & t2=0 ) &
  (next(em)=0 & next(t0)=t0+1 & next(t1)=t1+1 &
  next(t2)=0 ) ) |
( (em=1 & t0>10 & t0<21 & t2>0 & t2<8 & t1=0 ) &
  (next(em)=1 & next(t0)=t0 & next(t2)=t2+1 &
  next(t1)=0 ) ) |
( (em=2 & t0=0  & t1=0  & t2=0 ) &
  (next(em)=2 & next(t0)=0 & next(t1)=0 &
  next(t2)=0 ) ) |
( (em=1 & t2>0 & t2<8 & t0=-1 & t1=0 ) &
  (next(em)=1 & next(t2)=t2+1 & next(t0)=-1 &
  next(t1)=0 ) ) |
( (em=0 & t0>0 & t0<10 & t1=5 & t2=0) &
  (next(em)=1 & next(t0)=t0+11 & next(t1)=0 &
  next(t2)=1 ) ) |
( (em=0 & t0=10 & t1>0 & t1<5 & t2=0 ) &
  (next(em)=2 & next(t0)=0 & next(t1)=0 &
  next(t2)=0 ) ) |
( (em=0 & t0=10 & t1=5 & t2=0) &
  (next(em)=1 & next(t0)=-1 & next(t1)=0
  & next(t2)=1 ) ) |
( (em=0 & t0=10 & t1=5 & t2=0 ) &
  (next(em)=2 & next(t0)=0 & next(t1)=0 &
  next(t2)=0 ) ) |
( (em=1 & t0>10 & t0<21 & t1=0 & t2=8) &
  (next(em)=0 & next(t0)=-10+t0 &
  (next(t1)=1|next(t1)=2) & next(t2)=0) ) |
( (em=1 & t2=8 & t0=-1  & t1=0 ) &
  (next(em)=2 & next(t2)=0 & next(t0)=0 &
  next(t1)=0 ) )
```

Table 1: Description in NuSMV

does not have memory and is not an immediate candidate; if 1 $\leq$`t0`$\leq$ 10, then T_0 is enabled; if 11 $\leq$`t0`$\leq$ 20, then T_0 is disabled and has memory, the state in which it got disabled is `t0`-10; if `t0`=-1, then T_0 is immediate candidate. The variable `em` gives the (non-extended) marking of the system, it may have 4 values: it is -1 initially which corresponds to a "non-existing" marking (as a result of this trick the initial state of the model is deterministic which makes it easier to write specifications); when `em`=0 the token is in P_0; if `em`=1, then the token is in P_2; when `em`=2 the token is in P_1. The initial values of the variables are given under the keyword `INIT`. Under the keyword `TRANS` the possible current state/successor state pairs are described. The first three lines gives the possible initial states of the model: if the model is in the "non-existing" marking (i.e. `(em=-1 & t0=0 & t1=0 & t2=0)`, then the next state is one of the states satisfying the condition `(next(em)=0 & (next(t0)=1|next(t0)=2|next(t0)=3) & (next(t1)=1|next(t1)=2) & next(t2)=0 )`, i.e. the starting phase of the structures representing transition T_0 and T_1 is chosen according to $\boldsymbol{t_0}^{(0)}$ and $\boldsymbol{t_0}^{(1)}$ in a non-deterministic way. Then each three lines describe one of the Kronecker expressions of the incidence matrix. For example the second three lines determines how the variables change according to $\boldsymbol{T}^{(0)} \otimes \boldsymbol{T}^{(1)}$: `t0` and `t1` are incremented by one if they did not reach their last phase.

Model checking of the system can be carried out by complementing the description with specifications written in RTCTL [3]. We can write expressions that describe the temporal behavior at integer multiples of the time unit. For example the formula `EBF 17..17 m.em=2`, which evaluates to true, means that there exists such path that the model is in marking 2 (token in P_1) at the 17th step (see [7] for the syntax of specifications). For what concerns the TPN, since we start from a "non-existing" marking, it means that marking can be reached at time 16. Another example, which evaluates to false, is the specification `ABG 26..26 m.em = 2` means that all the possible paths lead to marking 2 by step 26. The fact that this expression is false means that it is not sure that the TPN gets to marking 2 by time 25. If an expression evaluates to false NuSMV provides a counter example. Specification `ABG 27..27 m.em = 2` evaluates to true which means that the TPN arrives to marking 2 by time 26 for certain.

We have chosen a very simple example to present the idea of model checking TPN;

however, this technique is applicable to real size problems as well thanks to the fact that NuSMV makes use of BDD, a very efficient storage technique.

As a short affix, hereinafter we show that the compositional way used to describe the behavior of the Petri Net can be utilized to carry out stochastic analysis as well. By substituting the local descriptors of the transitions with descriptors of Discrete Phase Type distributions, the global evolution of the process is still described by (1) with one difference. This difference is the result of the fact that while in functional analysis we describe possible successors, in stochastic analysis we have to associate probabilities to the possible successors. Therefore, we have to define what happens if transition T_0 and T_1 are candidates for firing in the same time instant (since the firing time distributions are not continuous it may have non-zero probability). To this end, the terms $\boldsymbol{t_f}^{(0)} \otimes \boldsymbol{t_f}^{(1)}$ are multiplied by $1/2$, which means that both transitions have probability 0.5 to fire first in case of being candidates together. The model was solved with the following parameters: firing time distribution of T_0 and T_1 is discrete uniform in the intervals $[8, 10]$ and $[4, 5]$, respectively, with stepsize 0.01; firing time of T_2 is deterministically 8. The probability that the process is in marking 2 versus time is depicted in Figure 4.

In accordance with the functional analysis, the probability of having a marking in P_1 is 0 before time 16 and is 1 at time 26. This way the *yes-no* results obtained by model checking are quantified in a probabilistic manner.

5 Conclusions

This paper, through a simple example, has presented a technique to check if a TPN satisfies properties expressed in RTCTL. The state space of the TPN is described by Kronecker expressions of local descriptors of the transitions. Using the Kronecker expressions the description of the model in the language of NuSMV can be generated automatically. NuSMV provides the possibility of checking RTCTL specifications of the model.

References

[1] A. Bobbio and A. Horváth. Petri nets with discrete phase type timing: A bridge between stochastic and functional analysis. In *Proc. of MTCS'01*, Aalborg, Denmark, Aug. 2001. To appear in Electronic Notes in Theoretical Computer Science.

[2] A. Bobbio, A. Puliafito, and M. Telek. A modeling framework to implement combined preemption policies in MRSPNs. *IEEE Transactions on Software Engineering*, 26:36–54, 2000.

[3] E.A. Emerson, A.K. Mok, A.P. Sistla, and J. Srinivasan. Quantitative Temporal Reasoning. *Journal of Real Time Systems*, 4:331–352, 1992.

[4] C. Ghezzi, D. Mandrioli, S. Morasca, and M. Pezzè. A unified high level Petri net formalism for time-critical systems. *IEEE Transactions on Software Engineering*, 17:160–171, 1991.

[5] P. Merlin and D. J. Faber. Recoverability of communication protocols. *IEEE Transactions on Communication*, 24(9):1036–1043, 1976.

[6] M. Scarpa and A. Bobbio. Kronecker representation of Stochastic Petri nets with discrete PH distributions. In *International Computer Performance and Dependability Symposium - IPDS98*, pages 52–61. IEEE CS Press, 1998.

[7] *NuSMV*. `http://nusmv.irst.itc.it/index.html`.