DEADLOCK ANALYSIS FOR CONTINUOUS PETRI NETS BY USING OVERLAPPING DECOMPOSITIONS

Hanife Apaydın-Özkan and Aydın Aybar

Department of Electrical and Electronics Engineering
Anadolu University
26555 Eskisehir, Turkey

hapaydin1@anadolu.edu.tr *aaybar@anadolu.edu.tr*

Abstract: The continuous Petri nets model, in which the tokens are denoted by real numbers and the firing of transitions depends on speeds of each transition and time, is considered to analyse deadlock in this work. Deadlock analysis, based on overlapping decompositions, is described for this Petri nets model. *Copyright © 2007 IFAC*

Keywords: Large-scale systems, discrete event systems, continuous Petri nets, overlapping decompositions, deadlock.

1. INTRODUCTION

Petri nets model which is a common modelling method of the discrete event systems has the several types such as generalized, colored, timed (transitions and/or places) and etc. (Cassandras and Lafortune, 1999). Since the firing of transitions in these types is described such that the tokens are transferred from the input places to the output places depending on the weight of arcs, the marking vectors correspond to the integer number of tokens of each place in Petri nets. Thus, the number of tokens indicates the availability and/or the state of a machine (device) (for example, if there exist three pallets, three tokens are in the related places). Since the integer number of tokens are used in classic (generalized / colored / timed) Petri nets, there may exist large number of reachable markings for the reachability set. The continuous Petri nets model was introduced by David and Alla (see, (David and Alla, 1987; David and Alla, 1990)) to prevent the explosion of the number of reachable markings in a classic Petri nets and modelling continuous systems which can not be modelled by classic Petri nets. Although the topology of this model is same as the classic Petri nets model, the enableness of transitions (strong and weak) and the number of tokens are different from the classic model.

In the continuous Petri nets model, the tokens are denoted by real numbers and the firing of transitions depends on speed of each transition and time. Furthermore, the evolution graph which represents the behaviour of the system with finite number of node (David and Alla, 1998) is constructed for this model. Note that, the firing speeds (with time notation) are defined and assigned to the transitions in this model. Several types of timed continuous Petri net have been developed (David and Alla, 1990; J. Le Bail and David, 1993; David and Alla, 2005). In this work, Constant Speed Continuous Petri Nets (CCPN) in which constant firing speeds associated with transitions is considered.

The behavioural properties (deadlock freeness and etc.) of classic Petri nets depend on the reachability set. Since the number of markings of the considered Petri net is, in general, exponentially related to its size (number of places and transitions) (see, (Aybar and İftar, 2003*b*)), the decentralized controllers were presented for enforcement of these properties. In the decentralized approach, the overlapping decompositons and expansions which was introdeced by Ikeda and

Šiljak (Ikeda and Šiljak, 1980) are used for the determination of each Petri subnets (Aybar and İftar, 2002), the controller design approach is used to design a controller for each subnet, and these controllers are then combined for implementation of the original Petri net (for example, (Aybar, *et al*., 2005)). Therefore, the computation complexity decreases for the controller design which enforces the behavioural property or properties. In the present work, we first extend the overlapping decompositions and expansions approach (Aybar and İftar, 2002) to CCPN. In this approach, the overlapping subnets of a CCPN are first identified. Deadlock analysis, based on overlapping decompositions, is described using the subnets of the CCPN. Finally, deadlock occurence in the original CCPN is detected by the deadlock analysis of all subnets.

2. CONSTANT SPEED CONTINUOUS PETRI NET

In this work, we define a constant speed continuous Petri net by a tuple $G = (P, T, N, O, V, m_0, \mathcal{P})$. Here, P is the set of *places*, T is the set of transitions, $N : P \times T \to \{0,\ 1\}$ is the *input matrix* that specifies the weights of arcs directed from places to transitions, $O\ :\ P \times T\ \to\ \{0,\ 1\}$ is the *output matrix* that specifies the weights of arcs directed from transitions to places. V: $T\ \to\ \mathcal{R}^+$ is vector of maximal firing speed, $V(t)$ corresponds to the maximal firing speed of transition $t\ \in\ T$, m_0 denotes initial marking of the CCPN, and $\mathcal{P}\ :\ T\ \to\ \mathcal{R}$ is the *priority vector*, $\mathcal{P}(t)$ corresponds to the priority value of transition $t\ \in\ T$, where $\mathcal{R}$ is the set of real numbers, and $\mathcal{R}^+$ is the set of nonnegative real numbers. For example, the initial marking is $m_0 = [1\ 0]^T$, the maximal firing speed vector is $V\ =\ [2\ 1]^T$, and the priority vector is $\mathcal{P} = [2\ 1]^T$ for the CCPN in Figure 1a.

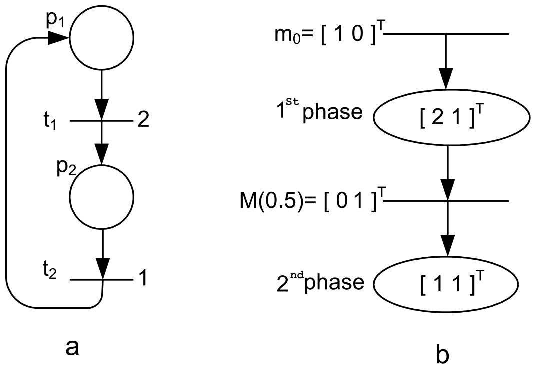

Fig. 1. a) An example CCPN   b) The evolution graph

The initial marking at time $\tau_0\ =\ 0$ is also denoted as $M(\tau_0) = m_0$. $M(\tau) : P \to \mathcal{R}$ is the marking vector at time τ, $M(\tau, p)$ denotes the number of tokens in place $p \in P$ at time τ.

In CCPN there are two kinds of enabling such as strong and weak. Transition $t \in T$ is *strongly enabled* at time τ if $M(\tau, p)\ >\ 0\ \forall p\ \in\ \bullet t$. Transition $t\ \in\ T$ is *weakly enabled* at time τ if for all input places p of transition t, there exists at least one enable transition $t' \in \bullet p$ (David and Alla, 1998). Here, $\bullet t$ denotes the set of input places of t, $\bullet p$ denotes the set of input transitions of p.

During the operation of CCPN each transition has *instantaneous firing speed*. The instantaneous firing speed vector at time τ is given by $v(\tau)$ and $v(\tau, t)$ indicates the instantaneous firing speed of $t \in T$ at time τ, i.e. $0 \leq v(\tau, t) \leq V(t)$. If a transition t is strongly enabled at a time τ, then its instantaneous firing speed is equal to its maximal firing speed, $v(\tau, t)\ =\ V(t)$. If any transition t is weakly enabled at a time τ its instantaneous firing speed is given by the relation:

$$v(\tau, t) = \min \left(V(t), \min_{p \in \mu} (B(\tau, p) + v(\tau, t)) \right) \qquad (1)$$

where $\mu\ :=\ \{p\ |\ p\ \in\ \bullet t$ and $M(\tau, p)\ =\ 0\}$ and $B(\tau)\ :\ P\ \to\ \mathcal{R}$ denotes the *dynamic balance* of a CCPN at time τ and calculated as, $B(\tau)\ =\ Av(\tau)$ (David and Alla, 2005), where $A\ =\ O\ -\ N$ is the incidence matrix.

In this work, we used the algorithms developed by (David and Alla, 1998) to determine enable transitions and instantaneous firing speed vector of CCPN.

Definition 1 : *Deadlock* is said to be occur in a CCPN, if $v(\tau, t) = 0,\ \ \forall t \in T$, and $M(\tau') = M(\tau),\ \tau' > \tau$.
Definition 2 : In a CCPN, there is an *actual conflict* among transitions in a subset of T, if the instantaneous firing speed of at least one of them has to be slowed down due to the other transitions in this set (David and Alla, 2005).

In this work, we use *priority rule* for resolution of actual conflict. The priority rule of the CCPN is given by $\mathcal{P}$ vector in the definition of Petri net. If transition $t_a \in T$ has higher priority than $t_b \in T$, then $\mathcal{P}(t_a) > \mathcal{P}(t_b)$.

2.1 Evolution graph

In a CCPN the instantaneous firing speed calculated at a time τ remains constant as long as the enabling conditions do not change. These conditions change when the marking of a place becomes zero. The time interval in which the instantaneous firing speed vector do not change is called as *phase* and the number of phases is denoted by θ.

In a CCPN, the marking of a place is a continuous function of time and the number of markings is infinite. The set of reachable markings from m_0 is denoted by $U(G, m_0)$, called as *reachability set*. It is then difficult to construct a graph of reachable markings in the general case. However, the behaviour of a CCPN can be represented by an evolution graph in which any element is called as node. Evolution graph represents the evolution of the CCPN with finite number of nodes. For example, the evolution graph of the

Download English Version:

https://daneshyari.com/en/article/720979

Download Persian Version:

https://daneshyari.com/article/720979

Daneshyari.com