EFFICIENT DEADLOCK DETECTION IN FMS BASED ON THE TRANSITIVE MATRIX OF RESOURCE SHARE PLACES

Jongwoog Kim, Jongkun Lee
LIS/Computer Engineering Dept.
Changwon National University
Salim-dong 9, Changwon, Kyungnam, Korea
{Jwkim,Jklee}@changwon.ac.kr

Abstract Since a deadlock is a condition in which the excessive demand for the resources being used by others causes activities to stop, it is very important to detect and prevent deadlocks. This paper proposes a new and more efficient deadlock detection algorithm based on the transitive matrix of resource share places. For presenting the results, the suggested deadlock detection and avoidance algorithms were also adapted to an illustrated model.

Key words: algorithm, conflict, deadlock, FMS, Petri-nets, resource share place, Transitive matrix

1. Introduction

The Flexible Management System (FMS) produces various items by resources such as machines, moving robots and parts. FMS consists of multi-procedures for performing tasks and the distribution of resources for smooth performance. However, since a deadlock is a status of arrested flow of a marking due to a delay in resources, the deadlock problem becomes one of the critical points in the scheduling problem of FMS. Therefore, it is necessary for an effective FMS control policy to ensure that deadlocks never occur [17]. Various deadlock analysis and avoidance methods to achieve confirmation and prevention of a deadlock status in a system have been proposed by researchers [1-3, 5, 6, 13-19]. Several methods among them have been adopted for deadlock resolution, based on a graph model, a finite state machines model and Petri Nets models [17]. In a Petri net analysis model, siphon analysis and reachability graph analysis have been used. Specifically, the siphon analysis method can be implemented by adding a control place with an initial marking and related arcs to the initial model. This requires some effect to find and analyze the characters of the siphon in its initial mode. In addition, the reachability graph method could obtain bigger models if the initial model is big enough. The object of this paper is to propose an easy but effective method to establish a deadlock detection policy. The deadlock problem in PN may occur at the resource share place in PN. This means that a resource share place holds through an alternative transition and at the same time requests a hold transition at the same place [15]. The properties of PN can be classified into behavioral and structural properties. The behavioral properties are investigated in association with the marking of PN, e.g., reachability, boundness and liveness [9]. Both transition and place invariant belong to structural properties in PN. Since transitive matrix could explain all relations between the place and transitions in Petri nets, we have reported an [8,9] to analyze scheduling in FMS using the transitive matrix after slicing resource share environment. In this paper, we propose an efficient deadlock detection policy based on the transitive matrix after showed a relationship between the resource share places.

2. Preliminaries of Petri nets

2.1 Petri Nets

In this section, some definitions of PN which are often used in the latter part of this paper are as follows [4,5,11,12,17-19].

A Petri net is a 5-tuple PN=<P,T,I,O,M,> where $P=\{p_1,p_2,..,p_m\}$ is a finite set of places, $T=\{t_1,t_2,\ldots,t_n\}$ is a finite set of transitions, and $P \cup T = \emptyset, P \cap T = \phi$, I:P x T -> N is an input function, O:T x P -> N is an output function where N is the set of positive integers. M:P-> N, is a marking representing the number of tokens in places with Mo denoting the initial marking.

A transition $t \in T$ is enabled under M, in symbols M[t>, iff $\forall p \in \bullet t$:M(p) > 0 holds. If M[t> holds the transition t may fire, resulting in a new marking M', denoted by M[t> M', with M'(p) = M(p) -1, if $p \in \bullet t \backslash t\bullet$; M'(p)=M(p) +1 if $p\in t\bullet\backslash\bullet t$; and otherwise M(p)=M'(p), for $\forall$ $p\in P$.

The set of reachable marking from a marking Mo is denoted as R(PN,Mo).

Let (PN,Mo) be a Petri net with PN=<P,T,I,O,M>. A transition $t \in T$ is live under Mo iff $\forall M\in R(PN,Mo)$, $\exists M'\in R(PN,M)$,M'[t> holds. PN is dead under Mo iff $\nexists t \in T$, Mo[t> holds. (PN,Mo) is deadlock-free iff $\forall M \in R(PN,Mo),\exists t\in T$,M[t>holds. (PN,Mo) is quasi-live iff $\forall t \in T$, $\exists M \in R(PN,Mo)$, M[t> holds. (PN,Mo) is live iff $\forall t \in T$, t is live under Mo. (PN,Mo) is bounded iff $\exists k \in N, \forall M \in R(PN,Mo),\forall p\in P,M(p) \leq k$ holds.

The matrix of a PN structure, C is $C=<P,T,B^-,B^+>$, where $P=\{p_1,p_2,..,p_m\}$ is a finite set of places, $T=\{t_1,t_2,\ldots,t_n\}$ is a finite set of transitions, and $P \cup T = \emptyset, P \cap T = \phi$, . B^- and B^+ are matrices of m rows by n columns defined by

$B^-=[I,j] = \#(P_i,I(t_j))$, matrix of input function,

$B^+=[I,j] = \#(P_i,O(t_j))$, matrix of output function.

Also, $B = B^+ - B^-$ is called an incidence matrix.

2.2 Transitive matrix

We recall now some basic definitions of transitive matrix which are as used in [7-9].

A column-vector $x^T=(x_1,x_2,\ldots,x_n) \geq 0$ of the homogeneous equation $A x^T=\Delta M =0$ is called a T-invariant, where x^T is x's transpose. An integer solution $y = (y1,y2,\ldots,ym)^T$ of the transposed homogeneous equation $Ay = 0$ is called a S-invariant. The place and the transition transitive matrix are as follows, respectively:

$B = B^-(B^+)^T$: place transitive matrix

Let L_{BP} be the labeled place transitive matrix:

$$L_{BP} = B^- \, diag(t_1,t_2,\ldots,t_n)(B^+)^T ,$$

where $t_i \, (i=1,2,\ldots,n)$ is :

$$|t_i| = \begin{cases} 1 \text{ fire } t_i \\ 0 \text{ not fire } t_i \end{cases}$$

The elements of L_{BP} describe the directly transferring relation that is from one place to another place through one or more transitions.

Let L^*_{BP} be the labeled place transitive matrix (MILN). If a transition t_k appears s times in the same column of L_{BP}, then we replace t_k in L^*_{BP} by t_k/s.

A row (column) vector in transitive matrix indexed by P and row (column) vector means that place p_i received (give) a token from (to) the column (row) places. If column (row) place has o value then this place will not give any token to row (column) place p_i.

A firing possibility of each transition to a row in the transitive matrix is represented and a token exists in the resource common place of a row direction.

Also, a token exists in the resource common place of a row direction. So, if $\sum (\frac{t_{k_i}}{s})_i$ is greater than 1, it is possible to firing, but if smaller than 1, this PN not able to fire.

From the basic definition, we propose a property of a deadlock detection condition based on the column and row vector's relationship in resource share places.

Property: Let $Rc(p_i)$ be a total token number of place p_i in column and $Rr(p_i)$ be a total token number of place p_i in row.

$$Rc(p_i) = \sum_{i=1}^{n}(p_i) = \sum_{i=1}^{n}(f \bullet p_i),$$

$$Rr(p_i) = \sum_{i=1}^{n}(p_i) = \sum_{i=1}^{n}(f \bullet p_i),$$

Where, n is number of place,

f is a calculate function in L^*_{BP}

if $\exists$ t then f = 1 else f = 0.

Then, $Dr(p_i) = Rr(p_i) - Rc(p_i)$ be a deadlock condition of place Pi.

If, $\sum_{j=1}^{n} Dr(p_j) = k$, where j is the number of resource share places,

If $k \geq 0$ and k is an integer value then this PN is deadlock-free, else this PN is deadlock.

Proof: Transitive matrix explains all relations between the places and transitions by OPN (Ordinary Petri net) in definition. This means that all places have only one input arc and one output arc. So, total of the token in each place should be integer value. If total of token value is less than zero then this place able to reach deadlock. Also, if total value of token is not integer value then this place could not able to fire.

3. Deadlock detect algorithm in Petri net using the transitive matrix

3.1 Deadlock

A deadlock problem occurs by the conflict place in the net. In this section, we consider an example of detecting a deadlock status based on the proposed policy.

(Example)

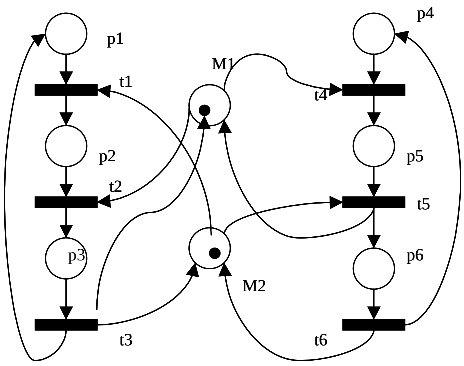

Figure 3.1 Example Petri nets

Table 3.1 Transitive matrix of figure 3.1

$L^*_{BP} =$

	P1	p2	p3	p4	p5	p6	M1	M2
P1	0	t1/2	0	0	0	0	0	0
P2	0	0	t2/2	0	0	0	0	0
P3	t3	0	0	0	0	0	t3	t3
P4	0	0	0	0	t4/2	0	0	0
P5	0	0	0	0	0	t5/2	t5/2	0
P6	0	0	0	t6	0	0	0	t6
M1	0	0	t2/2	0	t4/2	0	0	0
M2	0	t1/2	0	0	0	t5/2	t5/2	0

In this example, resource share places are M1 and M2. A deadlock detection condition of this model is:

1) Case of M1:
Rr(M1) : 1 + 1/2 + 1/2 = 2,
Rc(M1) : 1/2+1/2=1
Dr(M1) = 2-1 = 1

2)Case of M2:
Rr(M2) : 1+1=2,
Rc(M2) : 1/2+1/2 +1/2=1(1/2)
Dr(M2) = 2 – 1(1/2) = 1/2

$$\sum_{i=1}^{2} Dr(p_i) = Dr(M1) + Dr(M2) = 1+1/2 = 1(1/2)$$

This means that places M1 and p2 need tokens for enable transitions t2, i.e. deadlock.

3.2 Deadlock free
Now, in this section, we consider deadlock conditions in two examples of deadlock free (figure 3.2).

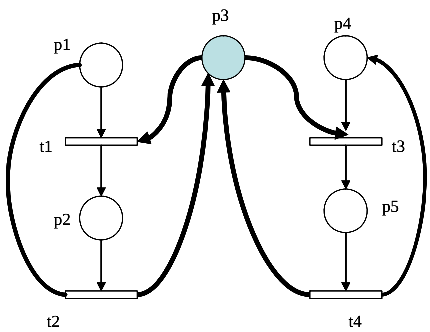

Figure 3.2 Example of deadlock free

Table 3.2 M_{PR} of Figure 3.2

$L^*_{BP} =$

	P1	p2	p3	p4	p5	
	0	$t1/2$	0	0	0	P1
	$t2$	0	$t2$	0	0	P2
	0	$t1/2$	0	0	$t3/2$	P3
	0	0	0	0	$t3/2$	P4
	0	0	$t4$	$t4$	0	P5

In this model, we can summarize the deadlock conditions as follows:
Resource share place p3:
Rr(p3) = 1+1=2,
Rc(p3) = 1+ 1=2
Dr(p3) = 2- 2 = 0
So, deadlock-free.

3.3 Deadlock detect algorithm
We propose a deadlock detect algorithm based on the previous section.
Algorithm: deadlock fine

Input: N = <P,T,F,M>
Output: N is deadlock free or not

(1) Define L^*_{BP} *of a Petri net initial N.*

(2) Find all relation of resource share places in each column . L^*_{BP}

(3) Find all relation of resource share places in each row L^*_{BP} *.*

(4) Calculate Dr(p_i*) : deadlock condition using the* L^*_{BP}

(5) Repeat (2)-(4) for all resource share places.

(6) If $\sum_{i=1}^{n} Dr(p_i) = 0$ *then this net is deadlock free, but if not then find Dr have negative places or not, if have negative place then this net N is deadlock.*

4. Application example

An FMS example introduced in [20] is represented in Fig. 4.1. There are 2 machines (M1 and M2), one robot and two transport devices, also two operations Job 1 and Job 2. We define the incorporate alternative process plans as follows:
Job 1: {M1, M2}
Job 2: {M2, M1}
The PN representation of system is as follows:

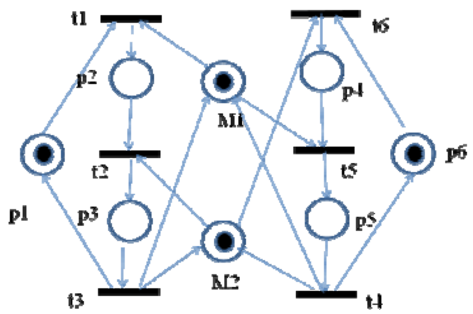

Figure 4.1 Example of model

Table 4.1 L^*_{BP} of figure 4.1

$$L^*_{BP} = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1 & 1 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The resource share places in this model are M1 and M2.

1) Case M1:
Rr(M1) = 1+1=2,
Rc(M1) = 1/2+1/2+1/2=1(1/2)
Dr(M1) = 2 - 1(1/2) = 1/2

2)Case M2:
Rr(M2) = 1+1= 2
Rc(M2) = 1/2+1/2=1
Dr(M2) 2-1= 1

Therefore, $\sum_{i=1}^{2} \mathrm{Dr}(p_i)$ = 1/2+ 1 = 1(1/2)

K is not integer value, and, this PN is deadlock.

5. Conclusion

In this paper, we proposed an analysis of the deadlock problem in Petri nets using the transitive matrix based on the resource share places. The deadlock problem occurs by the relationship between more than two transitions based on the resource share place. We defined a place which has the number of input tokens smaller than the number of output tokens as a dead node, we showed some deadlock conditions based on this relationship between the input and output tokens in the resource share place. In addition, we showed some examples to find a deadlock and a deadlock free status using the transitive matrix directly. The result showed that this method could be simpler than others. In the near future, studies to find a deadlock problem in the General Petri nets using the transitive matrix and also to avoid the deadlock avoidance problem on the cyclic scheduling of FMS will be reported. By way of rider, we will study a benchmark with other deadlock detection algorithms.

References

[1]Corbett JC, Evaluating Deadlock detection methods for concurrent software. International Journal of IEEE tr. Sof. Eng. 22(3), 1996

[2] J.Park, S.A.Reveliotis, Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible manufacturing, IEEE tr. Robotics and Automat., Vol.46, pp.1572-1583,2001

[3]Damasceno BC. And Xie X., Petri nets and deadlock-free scheduling of multiple-resource operations, International Conference of IEEE SMC'99 ,878-883,1999

[4]Desel J. And Esparza J., Free Choice Petri nets. Cambridge univ. Press,1995

[5]Ezpleta J.,Colom JM,Martinez J., A Petri net based deadlock prevention policy for flexible manufacturing systems, International Journal of IEEE tr. Robotics and Automat., 11(2),173-184,1995

[6] A.Giua,et elc., Observer-Based state-feedback control of timed Petri nets with deadlock recovery, In: IEEE Trans. on Automatic control, Vol. 49, No. 1, pp. 17-29,2004

[7]Liu J.,Itoh Y., Miyazawa I., Seikiguchi T., A Research on Petri nets Properties using the Transitive matrix, Proceeding of 1999 International Conference on IEEE SMC99,888-893,1999

[8] Lee J., Korbaa O., Scheduling analysis of FMS:An unfolding time Petri nets approach, mathematics and Computer simulation, 70, pp. 419-432,2006

[9] Lee J., Deadlock find algorithm using the Transitive Matrix", In: Proceeding CIE'04,2004

[10] F.basil,A.Giua,C.Seatzu,, Observer-based state-feedback control of time Petri nets with deadlock recovery: theory and implementation, In Proceed. Of Symp. On Discrete events in Industrial and manufacturing systems, CESA2003,2003

[11]Murata T., Petri Nets: Properties, Analysis an Applications, Proceedings of the IEEE, 77(4), 541-580,1989

[12]Peterson JL, Petri Net Theory and the Modeling of Systems, Englewood Cliffs, NJ : Prentice-Hall, Inc.,1987

[13] Li ZW,Zhao M, On Controllability of Dependent Siphon for Deadlock prevention in Generalized Petri nets, IEEE Tr. SMC-A, vol.38,no.2,pp.369-384,2008

[14]Han Z.,Lee G.,Application of Petri nets for deadlock analysis and avoidance in flexible manufacturing systems, Int J. Adv. Manu. Tech., vol .25, pp.735-742,2005

[15]Xiong HH,Zhou MC, Deadlock free scheduling of an automated manufacturing system based on Petri nets", Proceeding of the 1997 International Conference on IEEE ICRA'97, 945-950,1997

[16]Yoon H. and Lee D., Deadlock-Free Scheduling for Automated Manufacturing cells, Proceeding of International Conference on Control, Automation Robotics and Vision 2000,2000

[17] Uzam M. and Zhou MC, An Iterative Synthesis Approach to Petri net-based deadlock prevention Policy for Flexible manufacturing System, IEEE tran. SMC-A, vol. 37, no.3, pp.362-371,2007

[18[Li ZW, Hu HS,Wang AR, Design of Liveness-Enforcing Supervisors for Flexible manufacturing Systems Using Petri nets, IEEE tras. SMC-C,vol.37,no.4,pp.517-525,2007

[19] Z.W.Li,M.C.Zhou, Elementary siphons of Petri Nets and their application to deadlock prevention in flexible manufacturing system, IEEE tr. on system,Man and Cybern,part A,vol. 34,pp.38-51,2004

[20]Jeng ND,and DiCesare F.,Synthesis using resource Control Nets for Modeling Shated-resource Systems, IEEE Tr. On Robotics and Automation, Vol. 11, no. 3, pp.317-327,1995