Analysis of Petri net models using Transitive Matrix

Yu-Jin Song, Jong-kun Lee

Laboratory of Information System, Dept. of Computer Science,

Changwon National University, Changwon, Kyungnam, South Korea

Tel : +82-551-279-7421, Fax : +82-551-279-7420

E-mail : syj@sarim.changwon.ac.kr

Abstract : In this paper, we are focused on the analysis of Petri nets model using the subnet. Specially, we are proposes a divide-conquer method of Petri nets under the condition of one-boundedness for all the Petri nets. The usefulness of the approach is shown by applying the proposed techniques to a illustration model.

1. Introduction

The well-known strengths of Petri nets include their visual and easily understandable representation, their well-defined semantics, their ability to model concurrent and asynchronous system behavior, the variety of mature analysis techniques they offer (e.q. rechability, deadlock, safety, invariant, etc.), and the availability of software tools to assist modeling and analysis.

Methods of analysis for Petri nets may be classified into the following three groups [1]: 1) the cover ability (reachability) tree method [6], 2) the matrix-equation approach [7], and 3) reduction or decomposition techniques [2]. The first method involves essentially the enumeration of all reachable markings or their coverable markings. It should be able to apply to all classes of nets, but is limited to "small" nets due to the complexity of the state-space explosion. On the other hand, matrix equation and reduction techniques are powerful but in many cases they are applicable only to special subclasses of Petri nets or special situations.

Although many techniques have been proposed for analysis of Petri nets, there is an inherent difficulty of complexity in using them for real-life application.

To cope with this problem, this paper proposes a divide-conquer method of Petri nets under the condition of one-boundedness for all the Petri nets. We introduce the P-invariant transitive matrix of Petri nets and relationship between them. The feature of the P-invariant transitive matrix is that each element stands for the transitive relationship between input place and output place through the firing of the enable transition.

The rest of our paper consists of the following. The definitions and notations needed in this paper are given in Section 2. Section 3 proposes a divide-conquer method of Petri nets under the condition of one-boundedness for all the Petri nets. The Illustrative example is given in Section 4. Conclusion and future work are drawn in the last section.

2. Petri Nets

In this section, we present basic definitions of Petri nets. For a more detailed description, refer to [1,4]. A formal definition of Petri nets follows.

Definition 2.1 : A Petri net, PN, is a five-tuple structure, $PN = (P, T, I, O, M_0)$ where

1) P is a finite set of places.
2) T is a finite set of transitions.
3) $I : P \times T \rightarrow N$ is the input function, a mapping from Cartesian product of the set of places and the set of transitions to nonnegative integers.
4) $O : T \times P \rightarrow N$ is the output function, a mapping from Cartesian product of the set of transitions and the set of places to nonnegative integers.
5) $M_0 : P \rightarrow N$ is the initial marking function, a mapping from the set of places P to the nonnegative integers.

subject to the constraint : $P \cup T = \phi$

3. Petri net Slice

The method of partitioning the model divide the system as basic elements and the information of not included places are added in minimal invariant and existed in basic elements.

Definition 3.1 [5]: S-invariant

When PN is Petri net and there exists place vector i : $S_N \rightarrow Z$ which satisfies N' • i = 0, we say S-invariant. N' is represented by matrices of PN model (transition × place).

Definition 3.2 [5]: minimal invariant

When it is S-invariant and is represented by only positive value, we say minimal invariant (not including other S-invariants).

Definition 3.3 [5]: Petri net Slices

In Petri net $PN = (P, T, I, O, M_0)$ when place sets divided by partition algorithm are Slice Set={P_Slice$_i$ | i=1, N}, Petri net Slices are defined by (Slice$_i$ | i=1, n) and each Petri net Slice$_i$ = $(P_i, T_i, I_i, O_i, M_i)$ satisfies the following conditions.

- P_i = P_Slice$_i$,
- $L_i : T_i \rightarrow \Sigma^+$ is a function defined level of string in transition.
- $T_i = \{t \in T \mid s \in P_i, (s, t) \in F \text{ or } (t, s) \in F\}$,
- $I_i = \{(p, t) \in I, (p, t) \in I \mid p \in P_i, t \in T_i\}$,
- $O_i = \{(t, p) \in O, (p, t) \in O \mid p \in P_i, t \in T_i\}$,
- $\forall p \in M_i,\ M_i(p) = M(p)$.

In Petri net model, behavioral condition of transition t should be existed.

Petri net Slices are divided as subnets and are synchronized by transition level, and a behavioral condition is defined by the following conditions.

- When $L_i(t)$ is not shared : Only behavioral condition of transition should be satisfied.

If L_i (t) is shared: Every transition with L_i (t) should satisfy the behavioral condition.

4. Divide-conquer method

4.1 P-invariant Transitive Matrix

Definition 4.1: Let L_{BP} be the labeled place transitive matrix, defined by [7]

$$L_{BP} = B^{-}diag(t_1, t_2, ..., t_n)(B^{+})^{T} \qquad (1)$$

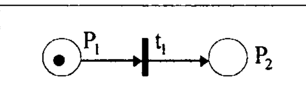
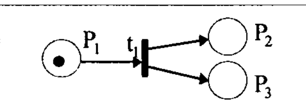
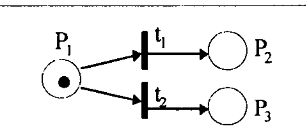
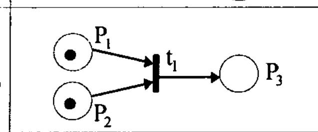
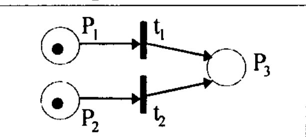
Here $t_i (i = 1,2,...,n)$ represent the labels with

$$|t_i| = \begin{cases} 1 \text{ when } t_i \text{ fires} \\ 0 \text{ when } t_i \text{ doesn't fire} \end{cases} \qquad (2)$$

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

We observe the relationship between some basic components of Petri net and labeled P-invariant transitive matrix.

Table 1. Relationship between PNs and labeled place transititive matrices

(a)		$L_{BP}=\begin{pmatrix} 0 & t_1 \\ 0 & 0 \end{pmatrix}$
(b)		$L_{BP}=\begin{pmatrix} 0 & t_1 & t_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
(c)		$L_{BP}=\begin{pmatrix} 0 & t_1 & t_2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
(d)		$L_{BP}=\begin{pmatrix} 0 & 0 & t_1 \\ 0 & 0 & t_1 \\ 0 & 0 & 0 \end{pmatrix}$
(e)		$L_{BP}=\begin{pmatrix} 0 & 0 & t_1 \\ 0 & 0 & t_2 \\ 0 & 0 & 0 \end{pmatrix}$

<Table 1> shows the examples of Petri net and their corresponding labeled place transitive matrix L_{BP}. The following observation can be obtained from the transition firing rule of Petri net.

Definition 4.2: Let L_{BP}^{*} be the m×m weighted place transitive matrix. 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 in L_{BP}^{*}. Otherwise, $L_{BP}^{*}[i, j]$ has the same entry as L_{BP}.

On the other hand, in Table 1 (d) we can obtain

$$L_{BP}^{*} = \begin{bmatrix} 0 & 0 & t_1/2 \\ 0 & 0 & t_1/2 \\ 0 & 0 & 0 \end{bmatrix}$$

4.2 Subnet

We made a subnet based on the slice concepts which is introduced in previews scetion; in this section we are proposed a subnet-made algorithm.

1) Defined L_{BP}^{*} in the Place transitive matrix.
2) Define correlation table of Place and Transition used L_{BP}^{*}.
3) Marked token place like as $P_i \bullet$.
4) Consider $P_i \bullet$ is an initial marking place in each subnets.
5) Find the all relational places in each column and made an element of own subnet with this initial marking place. Also, linked the same place after find the place in row.
6) Repeat 5) processes until the fire condition of transition satisfy to equation (3) in each row place.

$$\left| \frac{t_{k_i}}{s} \right| \geq 1, \qquad (3)$$

Where, t_k is a transition,

$i = 1,2,3,...,n$

s is times in the same column

7) If the place transitive matrices is equal to table 1(a), then repeat 5) and 6) process until transitive matrices is not equal to table 1(a).

5. Illustrative Examples

For the purpose of illustrating our method [3], an example is given in this section.

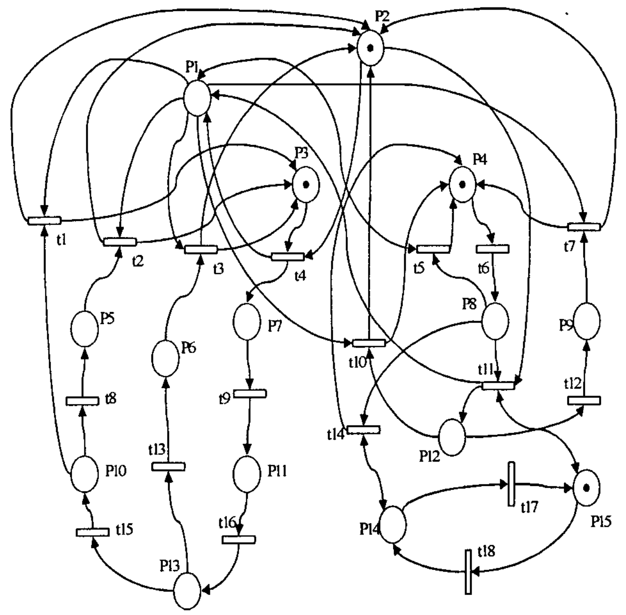


Fig 1. P/T net for basic call processing and call forwarding.

Table 2. Significance table of place and transition

Place	Significance	Transition	Significance
P_1	Busy	t_1	o-abandon
P_2	Not-busy	t_2	o-end-talk
P_3	C-idle	t_3	on-hook
P_4	R-idle	t_4	start-call
P_5	Talking	t_5	t-busy-tone
P_6	Busy-tone	t_6	t-routing
P_7	Hang-up	t_7	t-end-talk
P_8	Arrival	t_8	o-activate
P_9	Talk	t_9	Dialing
P_{10}	Ringing-tone	t_{10}	t-abandon
P_{11}	Number	t_{11}	t-ringing
P_{12}	Ringing	t_{12}	t-activate
P_{13}	Routing	t_{13}	o-busy-tone
P_{14}	CF	t_{14}	cf-routing
P_{15}	not-CF	t_{15}	o-ringing
		t_{16}	o-routing
		t_{17}	unsubscribe-cf
		t_{18}	subscribe-cf

Table 3. Correlation table of Place and Transition

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
P_1	$t_5/2$	$t_1/2$ $t_2/2$ $t_3/2$ $t_7/2$ $t_{10}/2$	$t_1/2$ $t_2/2$ $t_3/2$	$t_5/2$ $t_7/2$ $t_{10}/2$											
P_2•	$t_4/2$ $t_{11}/2$						$t_4/2$					$t_{11}/2$			
P_3•	$t_4/2$						$t_4/2$								
P_4•								t_6							
P_5		$t_2/2$	$t_2/2$												
P_6		$t_3/2$	$t_3/2$												
P_7											t_9				
P_8	$t_5/2$ $t_{11}/2$			$t_5/2$ $t_{14}/2$								$t_{11}/2$		$t_{14}/2$	
P_9		$t_7/2$		$t_7/2$											
P_{10}		$t_1/2$	$t_1/2$	t_8											
P_{11}													t_{16}		
P_{12}		$t_{10}/2$		$t_{10}/2$					t_{12}						
P_{13}						t_{13}				t_{15}					
P_{14}				$t_{14}/2$										$t_{14}/2$	t_{17}
P_{15}•														t_{18}	

<Table 2> show the function of places and transitions in Illustrative model <Fig. 1>.

<Table 3> show a correlation table of place and transitions in <Fig. 1> after used subnet-made algorithm 1)-3).

Let P_2• be a start-marking place in subnet. If we linked all places with P_2• after find places and transitions which are satisfy equation (3), then we can find a list as P_1-P_2-P_7-P_{12}, t_1-t_2-t_3-t_4-t_5-t_7-t_9-t_{10}-t_{11}-t_{12}. By the same method, P_3•, P_4•, P_{15}• have owns list. Specially, t_6 is equal to table 1(a)'s, so P_4• made a subnet after connect with P_8.

Finally, we made 4 subnets and show in <Fig 2>, <Fig 3>. In <Fig 2> (2), since this subnet has a connect point with other subnet, t_5, t_7, t_9, t_{10} are may ignored when analyze the subnet (2). Transition t_5, t_7, t_9, t_{10} are considered at (1) and (3) subnet. By the same method, P_7 is may ignored since it's give an affect to a connect point t_9. So, if we are deleting all connected point in this model, (2) subnet considers only P_1-P_3, t_1-t_2-t_3-t_4, and we can find this subnet is not deadlock, live, and bounded. The other subnets have the same properties. Finally, we can say that this model is not deadlock, live and bounded after analyzed the subnets of them.

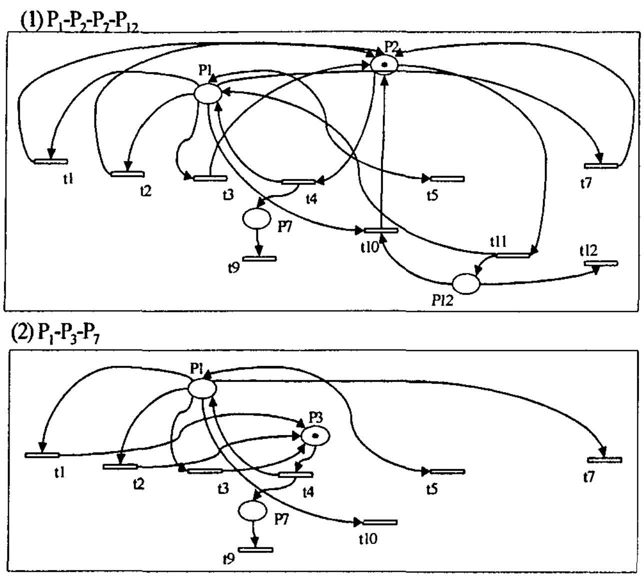


Fig 2. Subnet of Petri net model

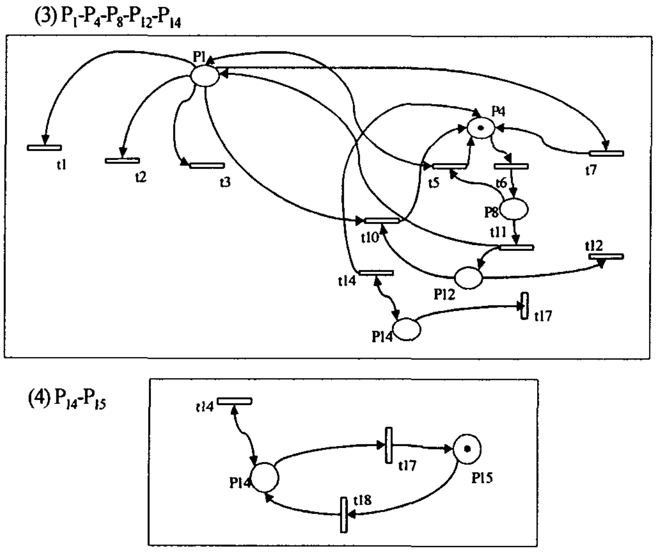


Fig 3. Subnet of Petri net model

6. Conclusion and Future work

It's very difficult to analyze the Petri nets model which have some properties like as concurrent, asynchronous, distributed, parallel, and/or no deterministic. Because these models be done more complex and very large size, so they need many times and more efforts to analyze. To solve this problem, in this paper proposed a transformation method using the subnet. Also, we show a divide-conquer method that is dived some subnet from original model used P-invariant transitive matrix. Our proposed method has a limited condition like as used in one-bounded Petri net. We will study continuously to propose the more general method for n-bounded Petri net's model and develop automatic tool.

References

[1] T. Murata, "Petri nets: Properties, analysis and applications", Proc. IEEE, vol. 77, pp. 541-580, 1989.

[2] A. Valmary, "Compositional State Space Generation", Advances in Petri Nets '93, 1993.

[3] G.J.Holzmann, "Design and Validation of Computer Protocols", Englewood Cliffs, NJ: Prentice Hall, 1991.

[4] J.L.Peterson, "Petri Net Theory and Modeling of Systems", Englewood Cliffs, NJ: Prentice Hall, 1981.

[5] Woo Jin Lee, Sung Deok Cha, Yong Rae Kwon, "Integration and Analysis of Use Cases Using Modular Petri Nets in Requirements Engineering", IEEE Transaction on Software Engineering, Vol. 24, No. 12, 1998.

[6] Masato Notomi and Tadao Murata, "Hierarchical Reachability Graph of Bounded Petri Nets for Concurrent-Software Analysis", IEEE Transaction on Software Engineering, Vol. 20, No. 5, 1994.

[7] Jinghong Liu, Yoshiaki Itoh, Iko Miyazawa, Tadashi Sekiguchi, "A Research on Petri Net Properties using Transitive Matrix", IEEE International Conference on Systems, Man, and Cybernetics, 1999