Expert Systems with Applications 40 (2013) 4562–4576

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A novel reduction approach for Petri net systems based on matching theory

Victor R.L. Shen [a,*], Yu Fang Chung [b], Shyi-Ming Chen [c], Jhong-Yi Guo [d]

[a] Department of Computer Science and Information Engineering, College of Electrical Engineering and Computer Science, National Taipei University, 151, University Rd., Sanhsia, New Taipei City 237, Taiwan
[b] Department of Electrical Engineering, Tunghai University, Taiwan
[c] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
[d] Graduate Institute of Communication Engineering, College of Electrical Engineering and Computer Science, National Taipei University, 151, University Rd., Sanhsia, New Taipei City 237, Taiwan

ARTICLE INFO

Keywords:
State explosion
Petri net
Net reduction
Reachability

ABSTRACT

In this paper, an efficient method is presented to solve the state explosion problem in Petri nets by using matching theory. It is difficult to analyze a Petri net when there are too many existing states. In order to solve such a problem, it is addressed to label a weight value on a transition according to the relationship between a place and a transition. Then, the transition with the largest weight value is selected. The selected transition is the most important and connective in the entire Petri net. After selecting each transition for several times, the last one denotes the least connective in the whole Petri net and the redundant place is obtained. Furthermore, the Petri net model can be reduced by fusing the transition with the largest weight value and the redundant place. In this novel approach, an incidence matrix, a weight vector, a matching matrix, a compressed incidence matrix, and a reduced and compressed incidence matrix are sequentially built based on the original Petri net model so as to obtain a reduced and compressed Petri net model. Finally, the experimental results regarding the CAVE automatic virtual reality environment demonstrate the high viability of the proposed approach.

1. Introduction

Petri nets are a graphical and mathematical modeling tool which can describe concurrent, asynchronous, distributed, parallel, nondeterministic or stochastic systems (Murata, 1989). However, the state explosion is a major obstacle to their analyses and practical applications. When the reachability graph of a Petri net model, which generates all possible states of the system, becomes so large that people cannot easily tract its behavior, the Petri net model turns out to be inefficient. In order to solve such a state explosion problem, many techniques have been proposed for the analysis of Petri nets.

Many scholars dedicated to developing reduction methods to solve the state explosion problem. They also proposed a lot of reduction methodologies in other types of Petri net model, such as Timed Petri Net (TPN) or Delay Time Petri Nets (DTPN) (Juan, Tsai, Murata, & Zhou, 2001), and Fuzzy Petri Nets (Shen, 2003; Shen, Chang, & Tong-Ying Juang, 2010; Shen & Tong-Ying Juang, 2008; Zhen-Huan, 2006). For example, Murata and Notomi proposed a method to construct hierarchically organized state space (HOSS) of a bounded Petri net (Murata & Notomi, 1992). However, they all reduced tiny parts of the net without considering the whole architecture. When the number of places and transitions becomes larger, reducing tiny parts of the net seems to be insufficient. Instead of focusing on tiny fragments, a new method to compress the Petri net is proposed in view of the whole architecture. At the same time, this method can also search for redundant places in the Petri net model.

The remainder of this paper is organized as follows. Section 2 discusses some preliminary definitions, including Petri nets, matching theory, and connection types. In order to illustrate the modeling capabilities and concepts of Petri nets, the weight vector algorithm, the matching matrix algorithm, the compressed incidence matrix algorithm, and the reduced incidence matrix algorithm are all given in Section 3 and Appendix. Section 4 describes the implementation and experiments. Finally, concluding remarks are given in Section 5.

2. Preliminaries

In this section, some basic definitions regarding Petri nets, matching theory, and connection types are presented.

* Corresponding author.
E-mail addresses: rlshen@mail.ntpu.edu.tw, victor.rlshen@msa.hinet.net (V.R.L. Shen), yfchung@thu.edu.tw (Y.F. Chung), smchen@mail.ntust.edu.tw (S.-M. Chen), mynameisjoyg@gmail.com (J.-Y. Guo).

0957-4174/$ - see front matter
http://dx.doi.org/10.1016/j.eswa.2013.01.057

2.1. Petri nets

A Petri net is composed of four parts, namely, a set of places P, a set of transitions T, an input function I, and an output function O. A Petri net is a 5-tuple, $PN = (P, T, F, W, M_0)$, where P is a finite set of places, T is a finite set of transitions, F is a set of arcs, W is a weight function, and M_0 is the initial marking (Peterson, 1981). Formally, we set

$$P = \{p_1, p_2, \ldots, p_m\}$$

$$T = \{t_1, t_2, \ldots, t_n\}$$

$$F \subseteq (P \times T) \cup (T \times P)$$

$$W : F \to 1, 2, 3, \ldots$$

$$M_0 : P \to 0, 1, 2, 3, \ldots$$

A Petri net structure $PN = (P, T, F, W)$ without any specific initial marking is denoted by PN, while a Petri net structure with the given initial marking is denoted by (PN, M_0). The behavior of a Petri net includes two activities: *Enabled* and *Firing*. Transition t is said to be enabled if each input place p of t is marked with at least $w(p,t)$ tokens, where $w(p,t)$ is the weight of the arc from place p to transition t.

2.2. Matching theory

The basic concepts of Petri nets (Murata, 1989), graph theory (Agnarsson & Greenlaw, 2006; Horowits, Sahni, & Mehta, 2003) and matching theory (Biedl, Beset, Demainet, & Lubiwt, 1999) are all applied to this study. Matching theory is part of graph theory. Given a graph $G = (V, E)$, V is a set of vertices in graph G and E is a set of edges in graph G. A matching M in G is a set of non-adjacent edges. In short, no two edges share a common vertex. A maximum matching is a matching that contains the largest number of edges. A perfect matching is a matching which covers all vertices of the graph. In the proposed matching method, the matching theory is redefined to suit the Petri net. The new definition of matching is based on the original matching theory. However, the vertices in the graph are regarded as places in the Petri net and the edges in the graph are regarded as arcs in the Petri net. Moreover, a maximum matching presents the largest number of arcs in the Petri net. When the Petri net has a maximum matching, the remained unmatched places are called redundant places. The details of the proposed reduction method is discussed below.

2.3. Connection types

There are four types of connection between places and transitions. In Fig. 1, Types 1, 2, and 3 are all special cases of Type 4. A different weight value is set on each transition according to the connection type. The weight is defined as follows:

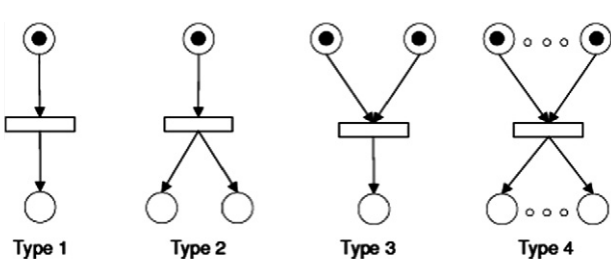

Fig. 1. Connection types.

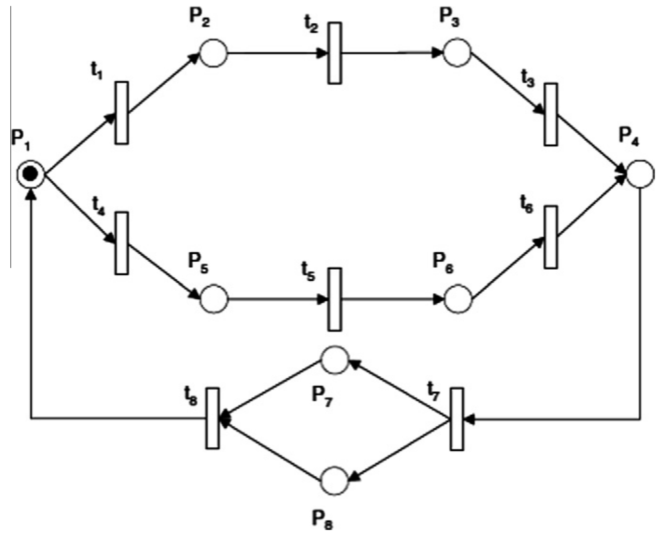

Fig. 2. Parallel processing system of a Petri net model.

$$A = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array}$$

Fig. 3. Incidence matrix.

$$W_1 = \begin{array}{c} \begin{array}{cccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \end{array} \\ [\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 \end{array}] \end{array}$$

$$W_2 = \begin{array}{c} \begin{array}{cccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \end{array} \\ [\begin{array}{cccccccc} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}] \end{array}$$

$$W_3 = \begin{array}{c} \begin{array}{cccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \end{array} \\ [\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}] \end{array}$$

Fig. 4. Weight vectors.

$$\begin{aligned} &I : |T| \\ &O : |T| \\ &W_i : |T| * |T| \end{aligned} \tag{1}$$

where I denotes an input function of transition T, i.e., the number of input places which connect to transition T; O denotes an output function of transition T, i.e., the number of output places to which transition T sends tokens; and W denotes the weight of transition T.

3. The proposed approach

In this section, the algorithms for building a weight vector, a matching matrix, a compressed incidence matrix, and a reduced and compressed incidence matrix are presented.