http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (4): 689-693, 2008
ISSN 1812-5638

Petri Net Methods of Constructing Kleene-Closure Operations of Regular Languages

WenJuan Lian, Hao Fan, YuYue Du and YongQuan Liang
College of Information Science and Engineering, Shandong University of Science and Technology,
Qingdao, Shandong, 266510, China

Abstract: It has been proved that regular language is a subclass of Petri net languages. Standard properly end Petri net, a subclass of Petri nets is defined in the related references, and the equivalency between a standard properly end Petri net and a regular language has been investigated. Thus, the Petri net constructing methods of Kleene closure operations with and without an ε-empty label in regular expressions are presented respectively in this study. By the methods, the net model of producing regular language L* can be constructed from that of producing regular language L. It is proved that a standard properly end Petri net language of Kleene closure operations is close.

Key words: Standard properly end Petri nets, Petri nets, regular language, regular expression, Kleene-closure operation

INTRODUCTION

As an important part of Petri nets (PN), PN language plays an important role in the description and analysis of system behavior (Du *et al*., 2007; Zeng and Duan, 2007; Scarpelli *et al*., 1996; Zeng, 2007). Some good research achievements of PN languages have been made. The relation between PN languages and Chomsky formal languages was analyzed (Peterson, 1981; Murata, 1989). The conceptions of a vector grammar and PN machines were presented, and the relations between vector grammar and PN language and between PN machine and PN languages were investigated (Jiang, 1996). A necessary and sufficient condition of whether a PN is a regular language or a context-free language was obtained based on the Pumping lemma in PNs. The Pumping lemma describes the commonness of PN language and by it we can prove that some languages could not be produced by PNs (Wu, 1994; Jiang and Lu, 2006; Wang *et al*., 2000).

How to create a PN with a known language expression or a language set? This problem is discussed few at present. The method of transferring a concurrence expression to a safe labeled PN was represented (Du *et al*., 2008). A subclass of PNs, the standard properly end PNs, was proposed and the equivalency between standard properly end PNs and regular languages has been proved (Lian *et al*., 2008). This research investigates mainly the constructing methods of Kleene closure operations in regular languages based on PNs. For a regular language L(N) produced by net N, we can construct a net model which produces language L(N)*. However, it's difficult to construct a language closure operation based on PNs, since some part of a PN model is repeatedly used and some positions may be not empty before they are used again during the construction process. According to the properties of standard properly end PNs, the PN methods of constructing closure operations in regular language are well resolved. Thus, the PN methods of constructing a kleene closure operation with and without an ε-empty label are presented in this paper. We prove that the closure operation in standard properly end PNs is close. That is, if L(N) is a formal language produced by a standard properly end PN, the PN model of producing language L(N)* is also a standard properly end PN.

BASIC CONCEPTS AND TERMS

Some basic concepts and properties of PNs are simply reviewed in this section and other concepts of PN languages can be referred by Peterson (1981) and Murata (1989).

Definition 1: (Peterson, 1981; Murata, 1989): Let $N = (P,T;G,M_0,\Sigma,h,F)$ be a labeled PN, where $(P,T;G,M_0)$ is a PN, $G \subseteq (P\times T)\cup(T\times P)$ is flow relation of PN, Σ is a finite alphabet, $h:T\rightarrow\Sigma$ is a labeled function, F is an end state set, then

$L(N) = \{\alpha \in \Sigma^* | \exists \sigma \in T^* \wedge M_0[\sigma>M \wedge M \in F \wedge h(\sigma) = \alpha\}$ is a language produced by the PN.

L(N) is called an L-type language produced by N if F is any subset of $R(M_0)$. According to different situations

Corresponding Author: WenJuan Lian, College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong, China

of labeled function h:T-Σ, there are three types of PN languages, i.e., PN languages without labeled languages, with and without an ε-empty label language. We mainly investigate the standard properly end Petri net methods of constructing closure operations with and without an ε-empty label in regular languages in this study.

Definition 2: Peterson (1981): Let $N = (P,T;G,M_0,\Sigma,h,F)$ be a labeled Petri net, it produces language L(N). It is called a standard Petri net if it satisfies the following conditions.

- $\exists\ p_s, p_f \in P$, such that $P_s = \phi$ and $p_f = \phi$. p_s is called a start position, p_f is called an end position.
- The initial marking M_0 satisfies
$$M_0(p) = \begin{cases} 1, & p = p_s \\ 0, & p \neq p_s \end{cases}$$
- Any transition will not be enabled once there exists a token in end position p_f.
- There is a marking M_f such that
$$M_f(p) = \begin{cases} 1, & p = p_f \\ 0, & p \neq p_f \end{cases}$$

If $\lambda \in L(N)$, then $F = \{M_0, M_f\}$, else F $\{M_f\}$.

Theorem 1: Peterson (1981): Let L be any Petri net language; there exists a standard Petri net which produces languageL.

Definition 3: Lian *et al.* (2008): Let $N = (P,T;G,M_0,\Sigma,h,F)$ be a standard Petri net, if $\forall M \in R(M_0)$: $M(P_f) = 1 \rightarrow \forall p \in P-\{p_f\}$: $M(p) = 0$, then N is a standard properly end Petri net.

Definition 4: Lian *et al.* (2008): If N is a standard properly end Petri net, then $L(N) = \{\alpha \in \Sigma^* | \exists \sigma \in T^* \wedge M_0[\sigma > M_f \wedge h(\sigma) = \alpha\}$ is called a language produced by N.

Theorem 2: Lian *et al.* (2008): Standard properly end Petri nets must be bounded Petri nets.

Theorem 3: Lian *et al.* (2008): A standard properly end Petri net language is equivalent to a regular language.

Definition 5: John *et al.* (2004): Let L_1, be a language in alphabet Σ. Kleene closure operation * is defined as follows:

$$L_1{}^* = \bigcup_{i=0}^{\infty} L_1 = L_1{}^0 \cup L_1{}^1 \cup L_1{}^2 \cup \cdots$$

where, $L_1{}^0 = \{\lambda\}$ and λ is an empty string.

PN METHOD OF CONSTRUCTING KLEENE CLOSURE OPERATIONS

Differences between standard properly end PNs and standard PNs: Standard properly end Petri nets are more special than common standard Petri nets from definitions 2 and 3. This can be illustrated with Fig. 1. In Fig. 1a, N_1 is a standard Petri net, but not a standard properly end Petri net. Since there exists $\sigma = \varepsilon aa\varepsilon b\varepsilon$ and $M_0[\sigma > M'$, where

$$M' = \begin{cases} 1 & p \in \{p_2, p_f\} \\ 0 & p \in P - \{p_2, p_f\} \end{cases},$$

it doesn't satisfy $\forall M \in R(M_0)$: $M(P_f) = 1 \rightarrow \forall p \in P-\{p_f\}$: $M(p) = 0$.

In Fig. 1b, N_2 is a standard properly end Petri net, and it can be seen when there exists a token in p_f. There isn't any transition which can be fired and there is not any place which includes redundant tokens. When N_2 stops running (no transition can be fired), therefore, it is in an acceptable state M_f. Thus N_2 is called a properly end net.

In Fig. 1a, the language produced by net N_1 is $L(N1) = a^n b^n$ and the net is in an acceptable state M_f, where

$$M_f(p) = \begin{cases} 1, & p = p_f \\ 0, & p \neq p_f \end{cases}$$

Although a transition sequence $\sigma = \varepsilon aa\varepsilon b\varepsilon$ can be fired by net N_1 and the net stops running at state M', where

$$M' = \begin{cases} 1 & p \in \{p_2, p_f\} \\ 0 & p \in P - \{p_2, p_f\} \end{cases}$$

M' is not the acceptable state of net N_1, there is redundant token in p_2, $M'(p_2) = 1$, thus, $h(\sigma) = h(\varepsilon aa\varepsilon b\varepsilon) = aab \notin L(N_1)$ Similarly, closure operation * in Petri net languages is not close. When the structure of net N_1 is

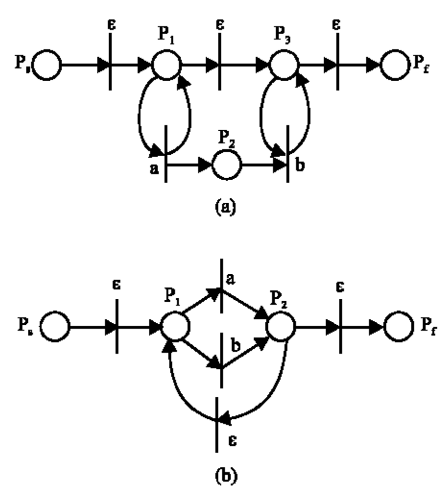

Fig. 1: Standard PNs and standard properly End PNs, (a) Standard Petri Net N_1 $L(N_1) = \{a^n b^n | n \geq 0\}$ and (b): Standard properly end Petri Net N_2 $L(N_2) = (a+b)^+$

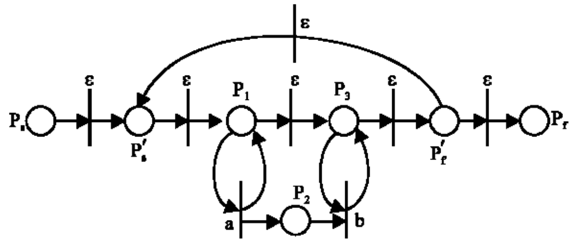

Fig. 2: Net N_3

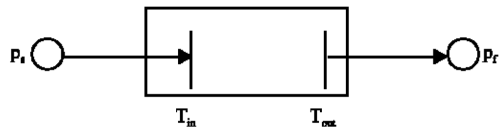

Fig. 3: Model of an abstract standard properly end petri net

used repeatedly to produce language $L(N_1)^* = (a^n b^n)^*$, it can't ensure some positions or position sets be empty before they're used again, such as p_2 in Fig. 2. For net N_2, there exists $M_0[\sigma>M_f$, where $\sigma = a^{n_{i_1}} b^{n_{j_1}} a^{n_{i_2}} b^{n_{j_2}}$ $(n_{i_1} \geq n_{j_1},\ n_{i_1} + n_{i_2} = n_{j_1} + n_{j_2}),\ h(\sigma) \in L(N_3)$. However, $L(N3) \neq (a^n b^n)^*$. For a standard properly end Petri net, an acceptable state M_f is reached when it stops running (no transition can be fired). That means that there is no redundant token in the net. Therefore, standard properly end Petri nets can be used to construct closure operation of regular languages.

Petri net (with ε-empty label) method of constructing closure operation: For convenience, an abstract standard properly end Petri net can be represented in Fig. 3. p_s, p_f are a start position and an end position, respectively. $T_{in} = \{t | t \in p_s^\bullet\}$ is a start transition set and $T_{out} = \{t | t \in {}^\bullet p_f\}$ is an end transition set.

Theorem 4: Let $N_1 = (P_1,T_1;G_1,M_{01},\Sigma,h_1,F_1)$ be a standard properly end Petri net, it produces a language $L(N_1)$. Then there exists a standard properly end Petri net $N = (p,T;,G,M_0,\Sigma,h,F)$ with an ε-empty label, such that the net produces a language $L(N) = (L(N_1))^*$.

Proof: Construct a net N from N_1 (Fig. 4), Let $L(N) = (L(N_1))^*$ be a language produced by N. Add two new places p_s, p_f as a start position and an end position of N, respectively. In order to produce Kleene closure *, we introduce four transitions $t_{k1},t_{k2},t_{k3},t_{k4}$ with an empty label. Therefore, we have $N = (P,T;G,M_0,\Sigma,h,F)$, where

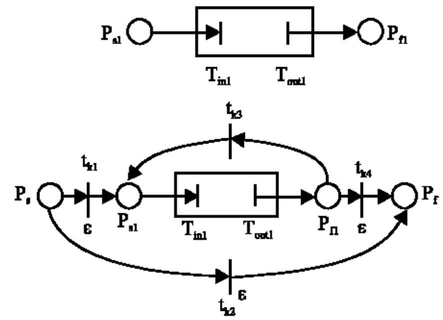

Fig. 4: Net N: L (N) = (L (N_1))*

$$P = P_1 \cup \{p_s,p_f\},\ T = T_1 \cup \{t_{k1},t_{k2},t_{k3},t_{k4}\}$$
$$G = G_1 \cup (\{p_s\}\times\{t_{k1},t_{k2}\}) \cup (\{t_{k1},t_{k3}\}\times\{p_{s1}\}) \cup (\{p_{f1}\}\times\{t_{k3},t_{k4}\}) \cup \{t_{k2},t_{k4}\}\times\{p_f\}$$
$$M_0(p) = \begin{cases} 1, & p = p_s \\ 0, & p \neq p_s \end{cases}$$
$$h(t) = \begin{cases} h_1(t), & t \in T_1 \\ \varepsilon, & t \in \{t_{k1},t_{k2},t_{k3},t_{k4}\} \end{cases} \qquad F = \{M_f\}$$
$$M_f(p) = \begin{cases} 1, & p = p_f \\ 0, & p \neq p_f \end{cases}$$

$L(N) = (L(N_1))^*$ is proved as follows.

If a transition sequence set $N_1(\sigma)$ is defined by $N_1(\sigma) = \{\sigma \mid \sigma \in T_1^* \wedge M_{01}[\sigma > M_{f1}\}$. Then $h_1(N_1(\sigma)) = L(N_1)$. We define two markings M_{s1} and M_{f1} in N, where
$$M_{s1}(p) = \begin{cases} 1, & p = p_{s1} \\ 0, & p \neq p_{s1} \end{cases} \text{ and } M_{f1}(p) = \begin{cases} 1, & p = p_{f1} \\ 0, & p \neq p_{f1} \end{cases}.$$
According to Fig. 4, there exist only three transition firing ways which can transfer a token from p_s to p_f.

(a) $h(t_{k2}) \in L(N)$ and $\varepsilon \in L(N)$ since $M_0[t_{k2}>M_f$.
(b) $M_0[t_{k1}>M_{s1}[\sigma>M_{f1}[t_{k4}>M_f$, where $\sigma \in N_1(\sigma)$. Let $\alpha = h_1(\sigma)$, then $\alpha \in L(N_1)$. According to this transition sequence σ, we have $h(t_{k1}\circ\sigma\circ t_{k4}) \in L(N)$ and $h(t_{k1}\circ\sigma\circ t_{k4}) = h(t_{k1})\circ h_1(\sigma)\circ h(t_{k4}) = \varepsilon\circ\alpha\circ\varepsilon = \alpha \in L(N)$. By means of the randomness of σ, $L(N_1) \subseteq L(N)$.
(c) $M_0[t_{k1}>M_{s1}[\sigma_{i1}>M_{f1}[t_{k3}>M_{s1}[\sigma_{i2}>M_{f1}[t_{k3}> \cdots M_{s1}[\sigma_{ij}>M_{f1}[t_{k4}> \cdots M_f$, where $\sigma_{i1}, \sigma_{i2}, \cdots\sigma_{ij} \cdots$ are random elements in $N_1(\sigma)$. Let $\alpha_{ij} = h_1(\sigma_{ij})$ $(j = 1,2\cdots)$, then $\alpha_{ij} \in L(N_1)$. According to transition sequence $t_{k1}\circ\sigma_{i1}\circ t_{k3}\circ\sigma_{i2}\circ t_{k3}\cdots\circ\sigma_{ij}\cdots\circ t_{k4}$, we have $h(t_{k1}\circ\sigma_{i1}\circ t_{k3}\circ\sigma_{i2}\circ t_{k3}\cdots\circ\sigma_{ij}\cdots\circ t_{k4}) \in L(N)$ and $h(t_{k1}\circ\sigma_{i1}\circ t_{k3}\circ\sigma_{i2}\circ t_{k3}\cdots\circ\sigma_{ij}\cdots\circ t_{k4})$
$= h(t_{k1})\circ h_1(\sigma_{i1})\circ h(t_{k3})\circ h_1(\sigma_{i2})\circ h(t_{k2})\cdots\circ h_1(\sigma_{ij})\cdots\circ h(t_{k4})$
$= \varepsilon\circ\alpha_{i1}\circ\varepsilon\circ\alpha_{i1}\ \varepsilon\circ\alpha_{i2}\circ\varepsilon\cdots\circ\alpha_{ij}\cdots\circ\varepsilon$
$= \alpha_{i1}\circ\alpha_{i2}\circ\cdots\alpha_{ij}\circ\cdots\in L(N)$

According to the randomness of σ_{i1}, σ_{i2}, $\cdots\sigma_{ij}\cdots$, it's known that $L(N_1)^k \subseteq L(N)$, where k is positive integer and $k \geq 2$. From rules a, b and c, we get

$L(N_1)^0 \cup L(N_1)^1 \cup L(N_1)^2 \cup \cdots = (L(N_1))^* \subseteq L(N)$ and $L(N) = (L(N_1))^*$ since there exits only three firing ways in net N . Therefore, N is a standard properly end Petri net and $L(N) = (L(N_1))^*$ based on the structure of N and the previous proof.

Petri net (without ε-empty label) method of constructing closure operation: It's more difficult to construct a Petri net model without ε-empty label, since a net structure must be reused. In the following, we discuss primarily how to construct a Petri net model without an ε-empty label of the closure operation of regular languages.

Theorem 5: Let $N_1 = (P_1,T_1;G_1,M_{01},\Sigma,h_1,F_1)$ be a standard properly end Petri net without an ε-empty label, it produces a language $L(N_1)$. Then there exists a standard properly end Petri net $N = (P,T;G,M_0,\Sigma,h,F)$, its labeled function h is without an ε-empty label and it produces a language $L(N) = (L(N_1))^*$.

Proof: Construct two new places p_s, p_f in N and they are used as a start position and an end position respectively. The structure of N_1 will be repeatedly used for 4 times to produce the kleene closure (*) of $L(N_1)$. The structure of $N = (P,T;G,M_0,\Sigma,h,F)$, is shown in Fig. 5 in detail and some unnecessary processes are omitted. From $\lambda \in (L(N_1))^*$, $F = \{M_0,M_f\}$ is the end marking set of N.

The proof of $L(N) = (L(N_1))^*$ is given as follows.

For convenience, some markings of net N are defined as

$$M_0(p) = \begin{cases} 1, & p = p_s \\ 0, & p \neq p_s \end{cases} \quad M_f(p) = \begin{cases} 1, & p = p_f \\ 0, & p \neq p_f \end{cases}$$

$$\text{and } M_{rep}(p) = \begin{cases} 1, & p = p_{rep} \\ 0, & p \neq p_{rep} \end{cases}.$$

According to the construction of net N, there exist only three ways of transferring a token from p_s to p_f. Therefore, three types of transition firing sequences are analyzed as follows.

(a) $M_0[\sigma > M_f$, where σ is any transition sequence in $N_1(\sigma)$. Then $L(N_1) \subseteq L(N)$.

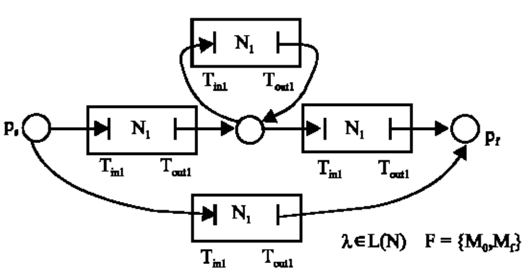

Fig. 5: Net N: $L(N) = (L(N_1))^*$

(b) $M_0[\sigma_1 > M_{rep}[\sigma_2 > M_f$, where $\sigma_1, \sigma_2 \in N_1(\sigma)$. Then $L(N_1) \circ L(N_1) \subseteq L(N)$.

(c) $M_0[\sigma_1 > M_{rep}[\sigma_{i1} > M_{rep}[\sigma_{i2} > M_{rep} \cdots M[\sigma_{ij} > \cdots M_{rep}[\sigma_2 > M_f$, where σ_1, σ_2 σ_{i1}, $\sigma_{i2}\cdots$ are random transition sequences in $N_1(\sigma)$. Then $L(N_1)^k \subseteq L(N)$. where k is a positive integer and $k \geq 2$.

According to cases a), b) and c), we have $L(N_1)^1 \cup L(N_1)^2 \cup \cdots = (L(N))^* \subseteq L(N)$. From $M_0 \in F$, then $L(N_1)^0 = \{\lambda\} \subseteq L(N)$ and $L(N) = (L(N_1))^*$. Therefore, N is a standard properly end Petri net by means of the structure of N and the previous proof.

EXAMPLES

With the respect to PNs with an ε-empty label and without an ε-empty label, Theorems 4 and 5 show respectively the PN methods of constructing kleene closure operation * in regular language. The use of the constructing methods is illustrated with the following examples.

Example 1: Let $\Sigma = \{a,b\}$ and net $N_1 = (P_1,T_1;G_1,M_{01},\Sigma,h_1,F_1)$ be a standard properly end Petri net. The structure of N_1 is shown in Fig. 6a and it produces a language $L(N_1) = ab$. Construct a standard properly end Petri net N with an ε-empty label and $L(N) = (L(N_1))^* = (ab)^*$.

According to Theorem 4, a standard end Petri net N with an ε-empty label can be constructed in Fig. 6b. We can easily verify that it satisfies $L(N) = (L(N_1))^* = (ab)^*$.

Example 2: Let $\Sigma = \{a,b\}$ and net $N_1 = (P_1,T_1;G_1,M_{01},\Sigma,h_1,F_1)$ be a standard properly end Petri net. Net N_1 is shown in Fig. 6a and it produces a language $L(N_1) = ab$. Construct a standard properly end Petri net N without an ε-empty label and $L(N) = (L(N_1))^* = (ab)^*$.

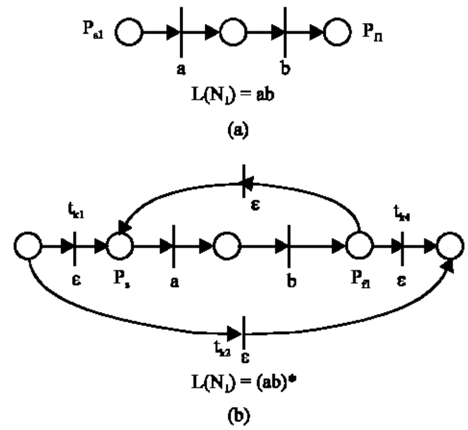

Fig. 6: (a) Net N_1 and (b): Net N

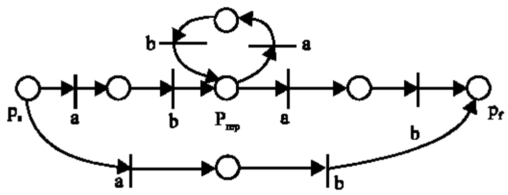

Fig. 7: Net N

According to Theorem 5, a standard end Petri net N without an ε-empty label can be constructed in Fig. 7. We can easily check it satisfies $L(N) = (L(N_1))^* = (ab)^*$.

CONCLUSIONS

Petri net languages play an important role in system behavior description and analysis. It is a fact that many operations of PN languages are close, such as connection, union, reverse, permutation operations. However, a closure operation is not close. It means that there is a net N which produces a language L(N), but the language $(L(N))^*$ is not a Petri net language. Standard properly end PNs are seen as a subclass of PNs in this paper, and its language is a regular language. We prove that closure operation of standard properly end Petri nets is close. Given a standard properly end Petri net N_1 producing a language $L(N_1)$, therefore, a standard properly end Petri net N can be constructed and it produces a language $(L(N_1))^*$. The Petri net with and without an ε-empty label methods of constructing closure operations in regular language are proposed explicitly, and the methods are illustrated with corresponding instances. This work has an important value to investigate further the Petri net language theory and applications.

ACKNOWLEDGMENTS

This study was supported in part by the National Natural Science Foundation of China under Grants 60773034, 60673053, 60603090, 60534060 and 60473094; the Taishan Scholar Construction Project of Shandong Province, China; the National Basic Research Program of China (973 Program) under Grants 2003CB316902, 2007CB316502 and 2004CB318001-03 and the Open Project of the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences under Grant SYSKF0604.

REFERENCES

Du, Y.Y., C.J. Jiang and M.C. Zhou, 2007. Modeling and analysis of real-time cooperative systems using petri nets. IEEE Trans. Syst., Man Cybernetics-Part A: Syst. Hum., 37: 643-654.

Du, Y.Y., C.J. Jiang and M.C. Zhou, 2008. A petri nets based correctness analysis of internet stock trading systems. IEEE Trans. Syst., Man Cybernetics-Part C: Applied Rev., 38: 93-99.

Jiang, C.J., 1996. Vector Grammar and PN Machine. Sci. Chin. (Ser. A), 24: 1315-1322.

Jiang, C.J. and G.J. Liu, 2006. The pumping lemma of petri net language. Chin. J. Comput., 29: 274-278.

John, E., R. Motwani, J.D. Ullman, 2004. Introduction to Automata Theory, Languages and Computation. 2nd Edn. Medium Textbook Hard cover Addison Wesley Longman Publishing Co. Inc. Bostan, MA, USA, pp: 521.

Murata, T., 1989. Petri nets: Properties, analysis and applications. Proceedings of the IEEE, 77: 541-580.

Peterson, J., 1981. Petri net theory and the Modeling of Systems. 1st Edn. Prentice-Hall, Englewood Cliffs, New Jersey .

Scarpelli, H., F. Gomide and R. Yager, 1996. A reasoning algorithm for high level fuzzy Petri nets. IEEE Trans. Fuzzy Syst., 4: 282-294.

Wang, H.Q., C.J. Jiang and S.Y. Liao, 2000. Behavior relations in synthesis process of petri net models. IEEE Trans. Robotics Automation, 16: 834-843.

Wu, Z.H., 1994. Petri net description of pumping lemma-a set of conditions for determining the type of a petri net language. Chin. J. Comput., 17: 852-858.

Zeng, Q.T., 2007. Two Symmetrical Decomposition Methods for Structure-Complex Petri Net and their Applications. 8th ACIS International Conference on Software Engineering, SNPD 2007. Artificial Intelligence, Networking and Parallel/Distributed Computing, pp: 1101-1106.

Zeng, Q.T. and H. Duan, 2007. Behavior description for complex flexible manufacturing system based on decomposition of Petri net. Int. J. Comput. Syst. Sci. Eng., 22: 359-363.