

Polynomial-time algorithm for the legal firing sequences problem of a type of synchronous composition Petri nets

JIANG Changjun (蒋昌俊)

Department of Computer Science, Tongji University, Shanghai 200092, China (email: cjjiang@online.sh.cn);
Department of Computer Science, Shandong Science & Technology University, Tai'an 271019, China;
Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China

Received November 7, 2000

Abstract As far as we know, the testing problem of legal firing sequence is NP-complete for general Petri net; the related results of this problem on the polynomial-time solvability are limited only to some special net classes, such as persistent Petri nets, conflict-free Petri nets and state machine Petri nets. In this paper, the language properties of synchronous composition net are discussed. Based on these results, the testing algorithm polynomial-time complexity for legal firing sequence is proposed. Therefore, net classification of polynomial-time solvability for testing legal firing sequence is extended.

Keywords: Petri net, synchronous composition, legal firing sequence, testing algorithm, NP-complete problem, polynomial-time complex.

In the dynamic analysis method, the testing problem of legal firing sequence (LFS) of Petri nets is quite fundamental in the sense that it is included, as a sub-problem, in various basic problems of Petri net theory. For example, (i) classical scheduling problems^[1-3] or cyclic scheduling problems that can be formulated as the testing problems of legal firing sequence for timed Petri nets^[4-6]; (ii) the minimum initial resource allocation problem (called the minimum initial marking problem)^[7-11]; (iii) the well-known state reachability problem^[12-14]. As far as we know, the testing problem of legal firing sequence is NP-complete for general Petri net, the related results of this problem on the polynomial-time solvability are limited only to some special net classes, such as persistent Petri nets, conflict-free Petri nets and state machine Petri nets^[7, 8, 15]. For a general Petri net model, the testing of legal firing sequence is also only an approximation algorithm^[15]. In this paper, the language properties of synchronous composition net are discussed. Based on these results, the testing algorithm polynomial-time complexity for legal firing sequence is proposed. Therefore, net classification of polynomial-time solvability for testing legal firing sequence is extended.

The question to be answered is described as follows:

(LFS) Given a Petri net $\Sigma = (P, T; F, M_0)$ and a nonnegative integers vector X , is there a legal firing sequence σ on M_0 , such that $M_0[\sigma >$ and each component $X(t)$ of X equals all to occurrence times $\#(t/\sigma)$ of the transition t in sequence σ ?

1 Synchronous composition operation of Petri net and its properties

Definition 1^[16-20]. Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2$ be two Petri nets. $P_1 \cap P_2 = \emptyset$, $T_1 \cap T_2 \neq \emptyset$. Set $\Sigma = (P, T; F, M_0)$, such that

- (i) $P = P_1 \cup P_2$;
- (ii) $T = T_1 \cup T_2$;
- (iii) $F = F_1 \cup F_2$; and
- (iv) $M_0(p) = M_{0i}(p)$, if $p \in P_i$, $i = 1, 2$.

Then Σ is called a synchronous connection net of Σ_1 and Σ_2 , while $\oplus$ is used to represent the synchronous operations, denoted as $\Sigma = \Sigma_1 \oplus \Sigma_2$. On the contrary, if a Petri net $\Sigma = (P, T; F, M_0)$ is synchronous decomposed into two Petri nets $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2$, such that the above conditions (i)–(iv) are satisfied. Then Σ_1 and Σ_2 are called synchronous decomposition nets of Σ .

Definition 2. Let X be a finite alphabet, $Y \subseteq X$.

(i) Set $\Gamma_{X \rightarrow Y}: X^* \rightarrow Y^*$, such that $\forall \sigma \in X^*$, and $\Gamma_{X \rightarrow Y}(\sigma)$ is the remnant sub-string after deleting each element in $X - Y$ from σ . $\Gamma_{X \rightarrow Y}$ is called a projection mapping from X to Y .

(ii) Set $\Gamma_{Y \rightarrow X}^{-1}: Y^* \rightarrow X^*$, such that $\forall \sigma' \in Y^*$, and $\Gamma_{Y \rightarrow X}^{-1}(\sigma') = \{\sigma \mid (\sigma \in X^*) \wedge (\Gamma_{X \rightarrow Y}(\sigma) = \sigma')\}$. $\Gamma_{Y \rightarrow X}^{-1}$ is called an extension mapping from Y to X , where “ $*$ ” is a closed operation of the language.

Definition 3. Let X be a finite alphabet, $Y \subseteq X$, L_X and L_Y be the languages on X and Y , respectively. Set $\Gamma_{X \rightarrow Y}(L_X) = \{\Gamma_{X \rightarrow Y}(\sigma) \in Y^* \mid \sigma \in L_X\}$ and $\Gamma_{Y \rightarrow X}^{-1}(L_Y) = \bigcup_{\sigma' \in L_Y} \Gamma_{Y \rightarrow X}^{-1}(\sigma')$. Then $\Gamma_{X \rightarrow Y}(L_X)$ and $\Gamma_{Y \rightarrow X}^{-1}(L_Y)$ are called the projection language of L_X from X to Y , and the extension language of L_Y from X to Y , respectively.

Lemma 1. Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2$ be two Petri nets, and $\Sigma = \Sigma_1 \oplus \Sigma_2 \overset{\Delta}{=} (P, T; F, M_0)$. Then $L(\Sigma) = \Gamma_{T_1 \rightarrow T}^{-1}(L(\Sigma_1)) \cap \Gamma_{T_2 \rightarrow T}^{-1}(L(\Sigma_2))$.

Proof. $\forall \sigma \in L(\Sigma)$, denoted $\sigma = \sigma[1]\sigma[2]\cdots\sigma[g] \in (T_1 \cup T_2)^*$. Set $\sigma_i \leftarrow \varepsilon$ (ε is an empty string). Since $M_0[\sigma >$, consider the following three cases:

Case 1. If $\sigma[1] \in T_1 - T_2$, $M_1 = (M_{11}^T M_{12}^T)^T$, such that $M_{01}[t_{i1} > M_{11} \wedge M_{12} = M_{02}$, then set $\sigma_1 \leftarrow \sigma_1 t_{i1}$;

Case 2. If $\sigma[1] \in T_2 - T_1$, $M_1 = (M_{11}^T M_{12}^T)^T$, such that $M_{02}[t_{i1} > M_{12} \wedge M_{11} = M_{01}$, then set $\sigma_2 \leftarrow \sigma_2 t_{i1}$;

Case 3. If $\sigma[1] \in T_1 \cap T_2$, $M_1 = (M_{11}^T M_{12}^T)^T$, such that $M_{01}[t_{i1} > M_{11} \wedge M_{02}[t_{i1} > M_{12}$, then set $\sigma_1 \leftarrow \sigma_1 t_{i1} \wedge \sigma_2 \leftarrow \sigma_2 t_{i1}$.

Similar processes will be also applied to $M_2, M_3, \dots, M_g$. Finally, σ_1 and σ_2 can be obtained such that $(\sigma_i \text{ is a sub-string of } \sigma, \sigma_i \in T_i^*) \wedge (M_{0i}[\sigma_i > M_{gi}) \wedge (i = 1, 2)$. Thus $\sigma_i = \Gamma_{T_i \rightarrow T}(\sigma) \in L(\Sigma_i)$. This gives $\sigma \in \Gamma_{T_1 \rightarrow T}^{-1}(\sigma_i) \subseteq \Gamma_{T_1 \rightarrow T}^{-1}(L(\Sigma_i))$, $i = 1, 2$. Or equivalently, $\sigma \in \Gamma_{T_1 \rightarrow T}^{-1}(L(\Sigma_1)) \cap \Gamma_{T_2 \rightarrow T}^{-1}(L(\Sigma_2))$, which leads to $L(\Sigma) \subseteq \Gamma_{T_1 \rightarrow T}^{-1}(L(\Sigma_1)) \cap \Gamma_{T_2 \rightarrow T}^{-1}(L(\Sigma_2))$.

The following formula can be proved through a reversal of the above procedure:

$$\Gamma_{T_1 \rightarrow T}^{-1}(L(\Sigma_1)) \cap \Gamma_{T_2 \rightarrow T}^{-1}(L(\Sigma_2)) \subseteq L(\Sigma).$$

Based on the above descriptions, we have

$$L(\Sigma) = \Gamma_{T_1 \rightarrow T}^{-1}(L(\Sigma_1)) \cap \Gamma_{T_2 \rightarrow T}^{-1}(L(\Sigma_2)).$$

Hence, Lemma 1 is proved.

Lemma 2. Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2$ be two Petri nets, $\Sigma = \Sigma_1 \oplus \Sigma_2$, and $\Delta = T_1 \cap T_2$. Then $\Gamma_{T \rightarrow \Delta}(L(\Sigma)) = \Gamma_{T_1 \rightarrow \Delta}(L(\Sigma_1)) \cap \Gamma_{T_2 \rightarrow \Delta}(L(\Sigma_2))$.

Proof. From Lemma 1, $L(\Sigma) = \Gamma_{T_1 \rightarrow T}^{-1}(L(\Sigma_1)) \cap \Gamma_{T_2 \rightarrow T}^{-1}(L(\Sigma_2))$. Thus $\Gamma_{T \rightarrow \Delta}(L(\Sigma)) = \Gamma_{T \rightarrow \Delta}(\Gamma_{T_1 \rightarrow T}^{-1}(L(\Sigma_1))) \cap \Gamma_{T \rightarrow \Delta}(\Gamma_{T_2 \rightarrow T}^{-1}(L(\Sigma_2))) = \Gamma_{T_1 \rightarrow \Delta}(L(\Sigma_1)) \cap \Gamma_{T_2 \rightarrow \Delta}(L(\Sigma_2))$. Hence it is completely proven.

Next, we discuss some language properties of Petri nets for synchronous composition (or decomposition). These results will be made into the basis of backward algorithms.

Lemma 3. Let T be a finite alphabet, and $\forall T_1, T_2 \subseteq T: T_1 \cup T_2 = T, T_1 \cap T_2 \neq \emptyset$. $\forall \sigma \in T^*$, set $\Gamma_{T \rightarrow T_i}(\sigma) = \sigma_i$, $i = 1, 2$. Then $\Gamma_{T_1 \rightarrow \Delta}(\sigma_1) = \Gamma_{T_2 \rightarrow \Delta}(\sigma_2)$, where $\Delta = T_1 \cap T_2$.

Proof. $\Gamma_{T \rightarrow T_1 \cap T_2}(\sigma) = \Gamma_{T \rightarrow T_1 \cap T_2}(\sigma)$ if and only if
 $\Gamma_{T_1 \rightarrow T_1 \cap T_2}(\Gamma_{T \rightarrow T_1}(\sigma)) = \Gamma_{T_2 \rightarrow T_1 \cap T_2}(\Gamma_{T \rightarrow T_2}(\sigma))$ if and only if
 $\Gamma_{T_1 \rightarrow \Delta}(\sigma_1) = \Gamma_{T_2 \rightarrow \Delta}(\sigma_2)$.

Definition 4. Let T_i , $i = 1, 2$ be two finite alphabets, $T_1 \cap T_2 \neq \emptyset$, L_{T_i} be the languages on T_i , $i = 1, 2$. For $\sigma_i \in L_{T_i}$, set $\sigma_1 \otimes \sigma_2 = \{\sigma \mid \Gamma_{T_1 \cup T_2 \rightarrow T_i}(\sigma) = \sigma_i, i = 1, 2\}$. Then $\sigma_1 \otimes \sigma_2$ is said as synchronous sequence set of σ_1 and σ_2 . Let X_i be the firing vectors corresponding with firing sequences σ_i , $i = 1, 2$. Set $X = X_1 \Theta X_2$, such that $\forall t \in T_1 \cup T_2$: (i) If $(t \in T_1 \cap T_2) \rightarrow (X_1(t) = X_2(t))$, then $X(t) = X_1(t)$; (ii) if $(t \in T_i - (T_1 \cap T_2)) \wedge (X_{3-i}(t) = 0)$, then $X(t) = X_i(t)$, $i = 1, 2$. Then X is said as synchronous firing vector of X_1 and X_2 .

Lemma 4. Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2$ be two live Petri nets, $\Sigma = \Sigma_1 \oplus \Sigma_2$, and $\Delta = T_1 \cap T_2$. If $\sigma_i \in L(\Sigma_i)$, $i = 1, 2$, then $\sigma_1 \otimes \sigma_2 \subseteq L(\Sigma)$, if and only if $\Gamma_{T_1 \rightarrow \Delta}(\sigma_1) = \Gamma_{T_2 \rightarrow \Delta}(\sigma_2)$.

Proof. $\sigma_1 \otimes \sigma_2 \subseteq L(\Sigma)$ if and only if $\sigma_1 \otimes \sigma_2 \subseteq \Gamma_{T_1 \rightarrow T}^{-1}(L(\Sigma_1)) \cap \Gamma_{T_2 \rightarrow T}^{-1}(L(\Sigma_2))$ (from Lemma 1);

if and only if $\sigma_1 \otimes \sigma_2 \subseteq \Gamma_{T_i \rightarrow T}^{-1}(L(\Sigma_i))$, $i = 1, 2$;

if and only if $\forall \sigma \in \sigma_1 \otimes \sigma_2: \Gamma_{T \rightarrow T_i}(\sigma) \in L(\Sigma_i)$, $i = 1, 2$;

if and only if $\Gamma_{T_1 \rightarrow \Delta}(\sigma_1) = \Gamma_{T_2 \rightarrow \Delta}(\sigma_2)$ (from Lemma 3).

Definition 5. Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2, \dots, k$ be k Petri nets, σ_i be the firing sequences of Σ_i , and X_i be the firing vectors corresponding with firing sequences σ_i , $i = 1, 2, \dots, k$. Then denoted

$$\Sigma_{1 * i} = \Sigma_1 \oplus \Sigma_2 \oplus \dots \oplus \Sigma_i; \quad T_{1 * i} = T_1 \cup T_2 \cup \dots \cup T_i;$$

$$\sigma_{1 * i} = \sigma_1 \otimes \sigma_2 \otimes \dots \otimes \sigma_i; \quad X_{1 * i} = X_1 \Theta X_2 \Theta \dots \Theta X_i;$$

$$i = 1, 2, \dots, k.$$

Lemma 5. Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2$ be two Petri nets, $\Sigma = \Sigma_{1 * 2} = (P, T; F, M_0)$. X be a $|T|$ -nonnegative integers vector of Σ , and $\Gamma_{T \rightarrow T_i}(X) = X_i$, $i = 1, 2$. Then X is a firing vector of Σ if and only if

- (i) X_i is a firing vector Σ_i , $i = 1, 2$;
(ii) $\Gamma_{T_1 \rightarrow T_1 \cap T_2}(\sigma_1) = \Gamma_{T_2 \rightarrow T_1 \cap T_2}(\sigma_2)$, where σ_i is the firing sequence corresponding with firing vector X_i , $i = 1, 2$.

Proof. ($\Rightarrow$) Since X is firing vector of Σ . Then there is a firing sequence corresponding with firing vector X of Σ . Set $\sigma_i = \Gamma_{T \rightarrow T_i}(\sigma)$, $i = 1, 2$. Since $\Gamma_{T \rightarrow T_i}(X) = X_i$, $i = 1, 2$, that is $X = X_1 \Theta X_2$. Then σ_i are the firing sequences corresponding with firing vector X_i , $i = 1, 2$ of Σ_i . Hence X_i are the firing vectors of Σ_i , $i = 1, 2$. Thus the result in (i) is true. Moreover, according to Lemma 3, the result in (ii) is also true.

($\Leftarrow$) Since X_i are the firing vectors of Σ_i , $i = 1, 2$. Then there are the firing sequence σ_i , $i = 1, 2$ corresponding with firing vector X_i of Σ_i . Since $\Gamma_{T_1 \rightarrow T_1 \cap T_2}(\sigma_1) = \Gamma_{T_2 \rightarrow T_1 \cap T_2}(\sigma_2)$, then we have $\sigma_1 \otimes \sigma_2 \subseteq L(\Sigma)$ according to Lemma 3. Thus $\exists \sigma \in \sigma_1 \otimes \sigma_2 \subseteq L(\Sigma)$, such that $\sigma_i = \Gamma_{T \rightarrow T_i}(\sigma)$, $i = 1, 2$. Since σ_i , $i = 1, 2$ are the firing sequences corresponding with firing vector X_i , and $\Gamma_{T \rightarrow T_i}(X) = X_i$, $i = 1, 2$, that is, $X = X_1 \Theta X_2$. Then σ is a firing sequence corresponding with firing vector X of Σ . Therefore, X is a firing vector of Σ .

Theorem 1. Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2, \dots, k$ be k Petri nets, $\Sigma = \Sigma_{1*2} = (P, T; F, M_0)$, X is a $|T|$ -nonnegative integers vector of Σ , and $\Gamma_{T \rightarrow T_i}(X) \stackrel{\Delta}{=} X_i$, $i = 1, 2, \dots, k$. Then X is the firing vector of Σ if and only if

- (i) X_i are the firing vectors of Σ_i , $i = 1, 2, \dots, k$;
(ii) $\Gamma_{T_{1*} \rightarrow T_{1*} \cap T_{i+1}}(\sigma_{1*i}) = \Gamma_{T_{i+1} \rightarrow T_{1*} \cap T_{i+1}}(\sigma_{i+1})$, where σ_{1*i} is the firing sequence corresponding with X_{1*i} of Σ_{1*i} . σ_{i+1} are the firing sequences corresponding with X_{i+1} of Σ_{i+1} , $i = 1, 2, \dots, k-1$.

Proof. Since operations $\oplus$, $\cup$, $\oplus$ and Θ all satisfy the law of association, according to the idea in fig.1 and Lemma 5, the result is true.

Theorem 2. Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2, \dots, k$ be k Petri nets. $\sigma_i \in L(\Sigma_i)$, $i = 1, 2, \dots, k$. Then $\Gamma_{T_{1*} \rightarrow T_{1*} \cap T_{i+1}}(\sigma_{1*i}) = \Gamma_{T_1 \rightarrow T_1 \cap T_{i+1}}(\sigma_1) \otimes \Gamma_{T_2 \rightarrow T_2 \cap T_{i+1}}(\sigma_2) \otimes \dots \otimes \Gamma_{T_i \rightarrow T_i \cap T_{i+1}}(\sigma_i)$, $i = 1, 2, \dots, k-1$.

Proof. $\forall \sigma_{\Delta} \in \Gamma_{T_{1*} \rightarrow \Delta}(\sigma_{1*i})$, where $\Delta = T_{1*} \cap T_{i+1}$. Then $\exists \sigma \in \sigma_1 \otimes \sigma_2$, such that $\Gamma_{T_{1*} \rightarrow \Delta}(\sigma) = \sigma_{\Delta}$. Set $\Delta_i = T_i \cap \Delta$, $i = 1, 2, \dots, k$. Then

$$\begin{aligned} \sigma_{\Delta} &\in \Gamma_{\Delta \rightarrow \Delta_1}(\sigma_{\Delta}) \otimes \Gamma_{\Delta \rightarrow \Delta_2}(\sigma_{\Delta}) \otimes \dots \otimes \Gamma_{\Delta \rightarrow \Delta_i}(\sigma_{\Delta}) \\ &= \Gamma_{\Delta \rightarrow \Delta_1}(\Gamma_{T_{1*} \rightarrow \Delta}(\sigma)) \otimes \Gamma_{\Delta \rightarrow \Delta_2}(\Gamma_{T_{1*} \rightarrow \Delta}(\sigma)) \otimes \dots \otimes \Gamma_{\Delta \rightarrow \Delta_i}(\Gamma_{T_{1*} \rightarrow \Delta}(\sigma)) \\ &= \Gamma_{T_{1*} \rightarrow \Delta_1}(\sigma) \otimes \Gamma_{T_{1*} \rightarrow \Delta_2}(\sigma) \otimes \dots \otimes \Gamma_{T_{1*} \rightarrow \Delta_i}(\sigma) \\ &= \Gamma_{T_1 \rightarrow \Delta_1}(\Gamma_{T_{1*} \rightarrow T_1}(\sigma)) \otimes \Gamma_{T_2 \rightarrow \Delta_2}(\Gamma_{T_{1*} \rightarrow T_2}(\sigma)) \otimes \dots \otimes \Gamma_{T_i \rightarrow \Delta_i}(\Gamma_{T_{1*} \rightarrow T_i}(\sigma)) \\ &= \Gamma_{T_1 \rightarrow \Delta_1}(\sigma_1) \otimes \Gamma_{T_2 \rightarrow \Delta_2}(\sigma_2) \otimes \dots \otimes \Gamma_{T_i \rightarrow \Delta_i}(\sigma_i). \end{aligned}$$

Therefore, $\Gamma_{T_{1*} \rightarrow T_{1*} \cap T_{i+1}}(\sigma_{1*i}) \subseteq \Gamma_{T_1 \rightarrow T_1 \cap T_{i+1}}(\sigma_1) \otimes \Gamma_{T_2 \rightarrow T_2 \cap T_{i+1}}(\sigma_2) \otimes \dots \otimes \Gamma_{T_i \rightarrow T_i \cap T_{i+1}}(\sigma_i)$, $i = 1, 2, \dots, k-1$.

On the other hand, if $\exists \sigma_{\Delta} \in \Gamma_{T_1 \rightarrow \Delta}(\sigma_1) \otimes \Gamma_{T_2 \rightarrow \Delta}(\sigma_2) \otimes \dots \otimes \Gamma_{T_i \rightarrow \Delta}(\sigma_i)$. However, σ_{Δ}

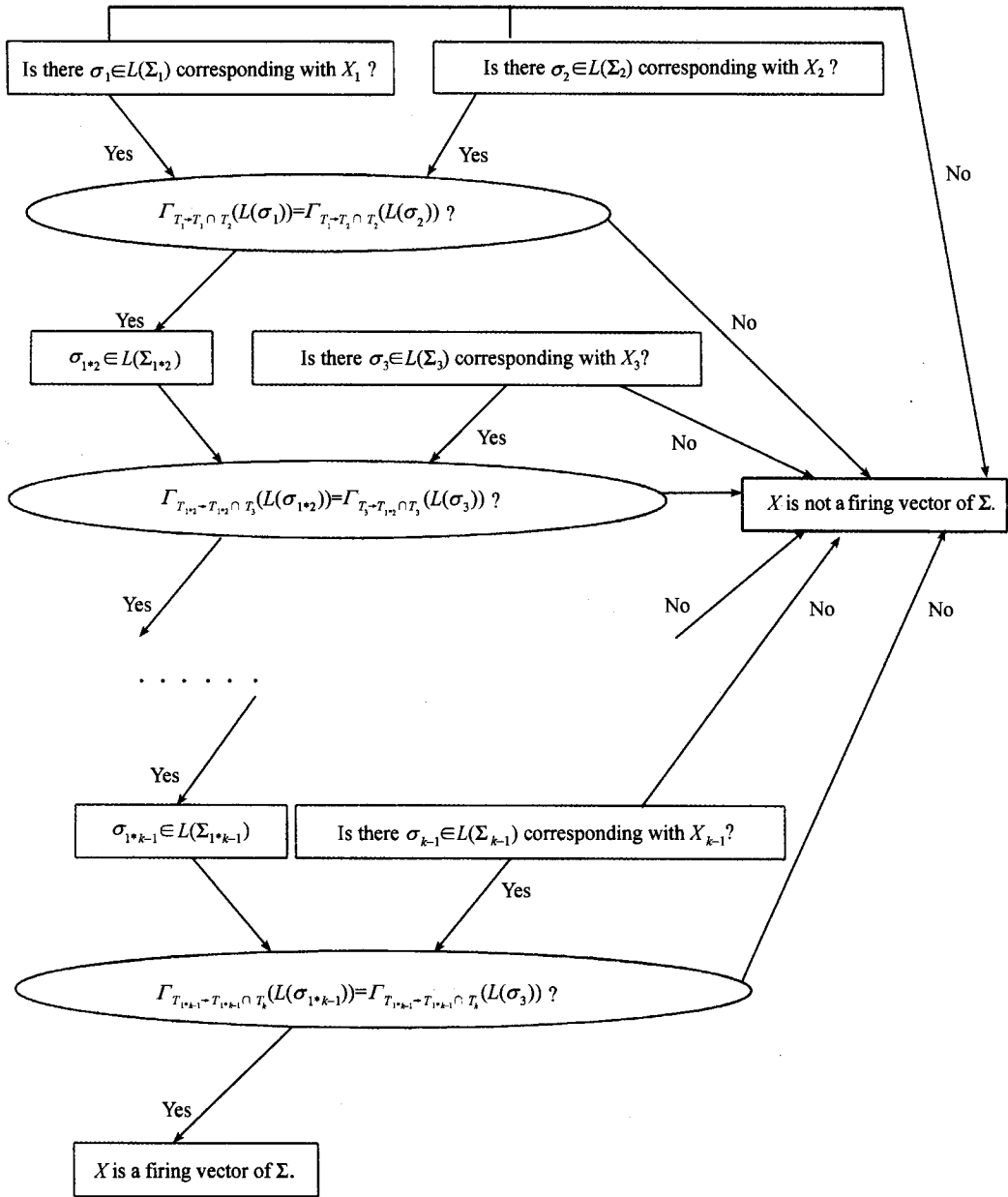


Fig. 1. The proving idea of Theorem 1.

$\notin \Gamma_{T_{1*},i \rightarrow \Delta}(\sigma_{1*i})$. Thus $\forall \sigma \in \sigma_{1*i}$, we have $\Gamma_{T_{1*},i \rightarrow \Delta}(\sigma_i) \neq \sigma_{\Delta}$. Since $\Gamma_{T_{1*},i \rightarrow T_i}(\sigma) = \sigma_i$, set $\Gamma_{T_i \rightarrow \Delta_i}(\sigma_i) = \sigma_{\Delta}$, $i = 1, 2, \dots, k$. Then

$$\sigma_{\Delta} \notin \sigma_{\Delta_1} \otimes \sigma_{\Delta_2} \otimes \dots \otimes \sigma_{\Delta_k} = \Gamma_{T_1 \rightarrow \Delta_1}(\sigma_1) \otimes \Gamma_{T_2 \rightarrow \Delta_2}(\sigma_2) \otimes \dots \otimes \Gamma_{T_k \rightarrow \Delta_k}(\sigma_k).$$

This is contradictory with $\sigma_{\Delta} \in \Gamma_{T_1 \rightarrow \Delta}(\sigma_1) \otimes \Gamma_{T_2 \rightarrow \Delta}(\sigma_2) \otimes \dots \otimes \Gamma_{T_k \rightarrow \Delta}(\sigma_k)$. Hence,

$$\begin{aligned} & \Gamma_{T_1 \rightarrow T_1 \cap T_{i+1}}(\sigma_1) \otimes \Gamma_{T_2 \rightarrow T_2 \cap T_{i+1}}(\sigma_2) \otimes \cdots \otimes \Gamma_{T_i \rightarrow T_i \cap T_{i+1}}(\sigma_i) \\ & \subseteq \Gamma_{T_{1*i} \rightarrow T_{1*i} \cap T_{i+1}}(\sigma_{1*i}), \quad i = 1, 2, \dots, k. \end{aligned}$$

This result is true.

2 Testing algorithm and its computing complexity

Based on Theorems 1 and 2, the testing algorithms of ledge firing sequence are described as follows:

Algorithm Testing.

Input $\Sigma = (P, T; F, M_0)$, $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1, 2, \dots, k, X$

Output b / If $\exists \sigma \in L(\Sigma)$: such that $\forall t \in T, X(t) = \#(t/\sigma)$, then $b = 1$, otherwise $b = 0$. /

- (1) begin
- (2) for $i = 1$ to k do
- (3) $X_i \leftarrow \Gamma_{T \rightarrow T_i}(X)$;
- (4) For X_i and Σ_i , the solutions are found by call algorithm in ref. [15]; / If $\exists \sigma_i \in L(\Sigma_i)$:
such that $\forall t \in T_i, X_i(t) = \#(t/\sigma_i)$, then $b_i \leftarrow 1$, otherwise $b_i \leftarrow 0$. /
- (5) if $b_i = 0$ then $b \leftarrow 0$;
- (6) goto (24);
- (7) endif;
- (8) endfor;
- (9) $b \leftarrow 1$; $T_{1*i} \leftarrow \emptyset$;
- (10) for $i = 1$ to $k - 1$ do
- (11) $T_{1*i} \leftarrow T_{1*i} \cup T_i$;
- (12) $\Delta \leftarrow T_{1*i} \cap T_{i+1}$;
- (13) $\sigma_{i+1, \Delta} \leftarrow \Gamma_{T_{i+1} \rightarrow \Delta}(\sigma_{i+1})$;
- (14) $\Delta_1 \leftarrow T_1 \cap T_{i+1}$;
- (15) $\sigma_{i, \Delta} \leftarrow \Gamma_{T_1 \rightarrow \Delta_1}(\sigma_1)$;
- (16) for $j = 2$ to i do
- (17) $\Delta_j \leftarrow T_j \cap T_{i+1}$;
- (18) $\sigma_{i, \Delta} \leftarrow \sigma_{i, \Delta} \otimes \Gamma_{T_j \rightarrow \Delta_j}(\sigma_j)$;
- (19) endfor;
- (20) if $\sigma_{i, \Delta} \neq \sigma_{i+1, \Delta}$ then $b \leftarrow 0$;
- (21) goto (24);
- (22) endif;
- (23) endfor;
- (24) endbegin.

Now, we analyse the time complexity of testing algorithm. First, the other two results for persistent Petri nets and state machine Petri nets are described in the following.

Lemma 6^[15]. The testing problem of legal firing sequence for a Petri net is NP-com-

plete.

Lemma 7^[7, 8, 15]. Let $\Sigma = (P, T; F, M_0)$ be a persistent Petri nets, and X be a $|T|$ nonnegative integers vector. Then the time complexity of testing algorithm depending whether X is a firing vector of Σ is $O(|P| \|X|)$.

Note. Since conflict-free Petri net is also persistent Petri^[12]. Therefore, the time complexity of testing algorithm for conflict-free Petri net is not larger than $O(|P| \|X|)$.

Lemma 8^[7, 8, 15]. Let $\Sigma = (P, T; F, M_0)$ be a state machine Petri net, and X be a $|T|$ nonnegative integers vector. Then time complexity of testing algorithm whether X is a firing vector of Σ is $O(|X|^2)$.

Theorem 3. Let PCSPN be the set of persistent Petri nets, conflict-free Petri nets and state machine Petri nets, $\Sigma_i = (P_i, T_i; F_i, M_{0i}) \in \text{PCSPN}$, $i = 1, 2, \dots, k$, $\Sigma = \Sigma_1 * \Sigma_2 = (P, T; F, M_0)$, X be a $|T|$ -nonnegative integers vector of Σ . Then time complexity of testing algorithm whether X is a firing vector of Σ is $O(\max(|T|^2, |P| \|X|, |X|^2))$.

Proof. In Algorithm Testing, the time complexity of computing X_i is $|T| \|T_i|$. According to Lemmas 7 and 8, the time complexity of computing σ_i by call algorithm in refs. [7, 8, 15] is $\max\{|P_i| \|X_i| + |X_i|^2\}$, and the time complexity of testing $b_i = 0$ is 1. Thus the time complexity of (2)—(8) is $\sum_{i=1}^k (|T| \|T_i| + \max\{|P| \|X_i|, |X|^2\} + 1)$. The time complexity of computing T_{1*i} is $|T_{1*i}| \|T_i|$. The time complexity of computing Δ is $|T_{1*i}| \|T_i|$. The time complexity of computing $\sigma_{i+1, \Delta}$ is $|T_{1*i}| \|X_{i+1}|$. The time complexity of (14)—(17) is $\sum_{j=1}^i (|T_j| \|T_{i+1}| + |X_j| \|T_j \cap T_{i+1}|)$. The time complexity of computing $\sigma_{i, \Delta}$ is $|X_i| \sum_{j=1}^{i-1} |X_j|$, and the time complexity of tasting $\sigma_{i, \Delta} \neq \sigma_{i+1, \Delta}$ is $|X_{i+1}| \sum_{j=1}^i |X_j|$. Thus, the time complexity of (2)—(8) is

$$\sum_{i=1}^k \left(|T_{1*i}| \|T_i| + |T_{1*i}| \|X_{i+1}| + \sum_{j=1}^i (|T_j| \|T_{i+1}| + |X_j| \|T_j \cap T_{i+1}|) + |X_i| \sum_{j=1}^{i-1} |X_j| + |X_{i+1}| \sum_{j=1}^i |X_j| \right).$$

Therefore, the time complexity of Algorithm Testing is

$$\begin{aligned} & \sum_{i=1}^k \left(|T_{1*i}| \|T_i| + |T_{1*i}| \|X_{i+1}| + \sum_{j=1}^i (|T_j| \|T_{i+1}| + |X_j| \|T_j \cap T_{i+1}|) + |X_i| \sum_{j=1}^{i-1} |X_j| + |X_{i+1}| \sum_{j=1}^i |X_j| \right) \\ & + \sum_{i=1}^k (|T| \|T_i| + \max\{|P| \|X_i|, |X|^2\} + 1) = O(\max(|T|^2, |P| \|X|, |X|^2)). \end{aligned}$$

3 Conclusion

As far as we know, the testing problem of legal firing sequence is NP-complete for general Petri net, the related results of this problem on the polynomial-time solvability are limited only to some special net classes, such as persistent Petri nets, conflict-free Petri nets and state machine

Petri nets^[7, 8, 15]. For general Petri net model, the testing of legal firing sequence is also only an approximation algorithm^[15]. In this paper, a polynomial-time testing algorithm of legal firing sequence for the synchronous composition net of some persistent Petri nets, conflict-free Petri nets and state machine Petri nets is proposed. Therefore, the results in refs. [7, 8, 15] are extended.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 69973029 and 69933020), the National Key Basic Science Foundation of P. R. China (973 Project, Grant No. G1998030604), the Key Project of National Science & Technology (Grant No. 95-B01-04), Excellent Ph. D Paper Author Foundation of China (Grant No. 199934), the Project of Shanghai Key Basic Foundation, the Project of Shanghai Dawn Plan Foundation and Project of Shandong Province Excellent Young Scientist Foundation.

References

1. Berenice, C., Xiaolan, X., Deadlock-free scheduling of manufacturing systems using Petri nets and dynamic programming, in Proc. of the 14th IFAC, Beijing, New York: IEEE Automation Society Press, 1999, 162—168.
2. Damasceno, B. C., Xiaolan X., Deadlock-free scheduling of manufacturing systems with multiple resources, in Proc. INCOM, Nancy-Metz, France, New York: IEEE Automation Society Press, 1998.
3. Ramaswamy, S. E., Joshi, S. B., Deadlock-free schedules for automated manufacturing workstations, IEEE Trans. on Robotics and Automation, 1996, 12(2): 391—400.
4. Watanabe, T. et al., Priority-list scheduling in timed Petri nets, Trans. IEICE of Japan, 1992, 75(10): 1394 ~ 1406
5. Watanabe, T. et al., New Priority-list for Scheduling in Timed Petri Nets, LNCS, Vol. 691, Berlin: Springer-Verlag, 1993. 493—512.
6. Yamauchi, M. et al., An approximation algorithm for the legal firing sequence problem of Petri nets, IPSJ SIG Notes, 1993, 17—24.
7. Watanabe, T. et al., Legal firing sequences and minimum initial markings for Petri nets, Proc. 1989 IEEE Int. Sympo. on Circuits and Systems, New York: IEEE Automation Society Press, 1989, 323—326.
8. Watanabe, T. et al., Legal firing sequences and related problems of Petri nets, Proc. The 3rd Int. Workshop on Petri Nets and Performance Models (PNPM89), New York: IEEE Computer Society Press, 1989, 277—286.
9. Watanabe, T. et al., Minimum initial marking problems of Petri nets, Trans. IEICE of Japan, 1989, 72(12): 1390—1399.
10. Watanabe, T. et al., The minimum initial marking problems for scheduling of timed Petri nets, Trans. IEICE of Japan, 1992, 75(10): 1407—1421.
11. Yamauchi, M. et al., A new approximation algorithm for the minimum initial marking problem of Petri nets, Proc. 6th Karuizawa Workshop on Circuits and Systems, IEICE of Japan, 1993, 279—284.
12. Kosaraju, R., Decidability of reachability in vector addition systems, Proc. 14th Annual ACM Symposium on Theory of Computing, New York: Prentice-Hall, 1982, 267—280.
13. Mayr, E. W., An algorithm for the general Petri net reachability problem, SIAM J. Comput., 1984, 13: 441—460.
14. Murata, T., Circuit theoretic analysis and synthesis of marking graphs, IEEE Trans. Circuits and Systems, 1977, 24(7): 400—405.
15. Watanabe, T. et al., Time complexity of legal firing sequences and related problems of Petri nets, Trans. IEICE of Japan, 1989, 72(12): 1400—1409.
16. Murata, T., Petri nets: Properties, analysis and applications, Proc. IEEE, 1989, 77: 541—580.
17. Wang, H. Q., Jiang, C. J., Liao, S. Y., Behaviour relations in synthesis process of Petri net models, IEEE Trans. on Robotics and Automation, 2000, 16(4): 400—407.
18. Jiang, C. J., Nets Operations (II), J. of Comp. Sci. & Tech., 1995, 10(6): 509.
19. Jiang, C. J., PN Machine Theory of Discrete Event Dynamic Systems, Beijing: Science Press, 2000, 129—171.
20. Shatz, S. M., Tu, S., Murata, T., An application of Petri net reduction for Ada tasking deadlock analysis, IEEE Trans. Parallel and Distributed Systems, 1996, 7(12): 1307.