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 (3): 420-429, 2008
ISSN 1812-5638

A Construction Method for the Process Expression of Petri Net Based on Decomposition

Qingtian Zeng
Department of Computer Science and Technology, Shandong University of Science and Technology, Qingdao 266510, Peoples Republic of China

Abstract: Process expression is one of the most useful tools to describe the process semantics of a Petri net. However, it is usually not easy to obtain the process expression of a Petri net directly. In this research, a construction method is proposed to obtain the process expressions of all kinds of Petri nets based on decomposition. With this decomposition method, a Petri net is decomposed into a set of S-Nets. The process properties of the decomposition nets are easy to analyze since they are well-formed and their process expressions are easier to obtain. With the process expressions of the decomposition nets, an algorithm to obtain the process expression of the original Petri net is presented, which is expressed by the synchronization shuffle operation between processes.

Key words: Petri net, S-Net, process, process expression, decomposition, synchronization shuffle

INTRODUCTION

To use Petri net for analyzing the properties of the physical systems, several methods have been presented in the net theory, among which process is a powerful tool for describing dynamical behaviors of net systems. Petri net processes are very convenient for analyzing concurrent phenomena and system properties relating to concurrency. This is an advantage of process method compared to other analysis methods about Petri net (Yuan, 2005; Garg and Ragunath, 1992; Murata, 1989). However, a process of a Petri net only gives one possible running case for the net system. There are usually many (sometimes maybe infinite) cases during a Petri net system running. It is difficult to obtain all the running cases, which brings difficulties for the analysis of Petri nets using their processes. Lu (1992a, b) has developed a new kind of nets, P/R nets, as a substitute for occurrence nets to describe executions of elementary net systems. A P/R net consists of a family of occurrences nets (called pages by Lu) and a contact situation of elementary net system is resolved by using several pages. Zeng and Wu (2002c) introduce the concept of process net system of a Petri net. The process net system is a reconstruction Petri net based on the set of the basic process sections, which describes the process behaviors of the original system very well. Wu (1996) and Wu *et al*. (2000) present the concept of process expression for bounded Petri net and unbounded fair Petri net. Roughly, any process of a bounded Petri net or unbounded fair Petri net is really a composition of several basic process sections (called subprocess by Wu). Zeng and Wu (2003) have proved the one-to-one corresponding relation between the transition firing sequences of the process net system and the processes of the original Petri net and presented a method to obtain the process expression of any unbounded Petri net based on its process net system.

This study proposes a new method to construct the process expression of a Petri net based on a kind of decomposition methods. This decomposition method is very useful for property analysis of structure-complex Petri nets since the structure of the decomposition net is well-formed (Zeng and Wu, 2002a, 2004a, b). We analyze the language relations during the decomposition and present a method to obtain the language of a Petri net (Zeng and Wu, 2004a). In Zeng and Wu (2004b), we discuss the conditions to keep states and behaviors invariant during the decomposition and present the judgment algorithm. In this study, we analyze the process properties of the decomposition nets and present the methods to obtain their process expressions. By discussing the process relations during the decomposition, an algorithm is proposed to present the process expression of the original Petri net based on the process expressions of the decomposition nets. This method suits obtaining the process expression for bounded or unbounded, fair or unfair Petri nets and it is not necessary to construct the process net system.

RELATED TERMINOLOGY AND NOTATIONS ABOUT PETRI NET PROCESS

To save space, it is assumed that the readers are familiar with the basic definitions of Petri nets (Yuan, 2005; Murata, 1989; Zeng and Wu, 2002c; Wu, 1996). Some of the essential terminology and notations related to this research are defined as follows. Convenient to define, we only consider the process of P/T system and suppose that $K = \omega$ and $W = 1$. We also assume that the Petri net discussed in this work is finite and connected and is not an S-Net (Yuan, 2005).

PROCESS AND BASIC PROCESS SECTION PETRI NET

Definition 1: A net $N = (B, E; G)$ is an occurrence net if

(1) $\forall b \in B: |{}^\bullet b| \le 1 \wedge |b^\bullet| \le 1$ and
(2) $\forall x, y \in B \cup E: (x, y) \in G^+ \rightarrow (y, x) \notin G^+$,

where, G^+ is the transitive closure of the flow relation G.

Definition 2: Let $N_1 = (S, T; F)$ be a net and $N_2 = (B, E; G)$ be an occurrence net. A mapping $\varphi: B \cup E \rightarrow S \cup T$ is a mapping from N_2 to N_1, denoted by $\varphi: N_2 \rightarrow N_1$, if it satisfies:

(1) $\varphi(B) \subseteq S$; $\varphi(E) \subseteq T$,
(2) $\forall x, y \in B \cup E: (x, y) \in G \rightarrow (\varphi(x), \varphi(y)) \in F$,
(3) $\forall e \in E: \varphi({}^\bullet e) = {}^\bullet\varphi(e)$, $\varphi(e^\bullet) = \varphi(e)^\bullet$

Definition 3: Let $\Sigma = (N_1, M_0) = (S, T; F, M_0)$ be a Petri net and $N = (B, E; G)$ be a occurrence net. A process of Σ is a couple (N, φ), where φ is a mapping from N to N_1 and satisfies the following conditions:

(1) $\forall b_1, b_2 \in B: (b_1 \ne b_2): \varphi(b_1) = \varphi(b_2) \rightarrow {}^\bullet b_1 \ne {}^\bullet b_2 \wedge b_1^\bullet \ne b_2^\bullet$ and
(2) $\forall s \in S: |\{b \mid \varphi(b) = s \wedge {}^\bullet b = \phi\}| \le M_0(s)$

Definition 4: Let φ be a mapping from a occurrence net $N = (B, E; G)$ to a Petri net $\Sigma = (S, T; F, M)$. If:

(1) $\forall b_1, b_2 \in B, (b_1 \ne b_2) \wedge \varphi(b_1) = \varphi(b_2) \rightarrow$
$({}^\bullet b_1 \ne {}^\bullet b_2 \vee {}^\bullet b_1 = {}^\bullet b_2 = \phi) \wedge (b_1^\bullet \ne b_2^\bullet \vee b_1^\bullet = b_2^\bullet = \phi)$
(2) $|\{b \mid \varphi(b) = s \wedge {}^\bullet b = \phi\}| = M_0(s)$

then $P = (N, \varphi)$ is said to be a surjective process of Σ.

Definition 5: Let $P = (N, \varphi)$ be a surjective process of Σ, where $N = (B, E; G)$. Let u_1 and u_2 be two S-cuts of N such that $u_1 \le u_2$. We define $N_1 = (B_1, E_1; G_1)$ such that:

(1) $B_1 = \{x \in B \mid \exists b_1 \in u_1, b_2 \in u_2: (b_1, x) \in G^* \wedge (x, b_2) \in G^*\}$;
(2) $E_1 = \{y \in E \mid \exists b_1 \in u_1, b_2 \in u_2: (b_1, y) \in G^* \wedge (y, b_2) \in G^*\}$;
(3) $G_1 = G \cap (\{B_1 \times E_1\} \cup \{E_1 \times B_1\})$

If $\varphi_1: N_1 \rightarrow \Sigma$ satisfies $\forall x \in B_1 \cup E_1: \varphi_1(x) = \varphi(x)$, (N, φ_1) is said to be a section (between u_1 and u_2) of process P, sometimes it is also called a process section of Σ, denoted by $(N[u_1, u_2], \varphi)$ (Zeng and Wu, 2002c).

Definition 6: Let $P = (N, \varphi)$ be a surjective process of Σ and $P_1 = (N[u_1, u_2], \varphi)$ be a process section of Σ. If any two s-cuts u_i and u_j $(i, j \ne 1,2)$ in $N[u_1, u_2]$ satisfies:

$$u_i \ne u_j \rightarrow (\varphi(u_i) \ne \varphi(u_j)) \wedge (\varphi(u_i) \not< \varphi(u_j)) \wedge (\varphi(u_i) \not> \varphi(u_j))$$

then $P_1 = (N[u_1, u_2], \varphi)$ is said to be a basic process section of Σ.

The set of all the basic process sections of a Petri net Σ is denoted by $BP(\Sigma)$ (Zeng and Wu, 2003).

PROCESS NET SYSTEM OF PETRI NET

The concept of process net system of a Petri net was first introduced by Zeng and Wu (2002c). The process net system Σ_p of a Petri net Σ describes the process behaviors of Σ very well. Every transition firing sequence of Σ_p is corresponding to a surjective process of Σ. If the last process section of a given surjective process is complete, there is a transition firing sequence corresponding to it.

Definition 7: Let $P = ([u_1, u_2], \varphi)$ be a basic process section of a Petri net $\Sigma = (S, T; F, M_0)$ and φ be the mapping from the occurrence net $N = (B, E; G)$ to Σ.

(1) The sets ${}^0P = \{s \mid s \in S \wedge \varphi^{-1}(s) \in u_1\}$ and $P^0 = \{s \mid s \in S \wedge \varphi^{-1}(s) \in u_2\}$ are respectively named as the input and the output place set of P.
(2) The bag $B({}^0P) = \{s \mid s \in S \wedge \varphi^{-1}(s) \in u_1\}$ is the input place bag of P, where

$$\forall s \in B({}^0P), \#(s, B({}^0P)) = |\{b \mid b \in u_1 \wedge \varphi(b) = s\}|$$

Similarly, the bag $B(P^0) = \{s \mid s \in S \wedge \varphi^{-1}(s) \in u_2\}$ is the output bag of P, where

$$\forall s \in B(P^0), \#(s, B(P^0)) = |\{b \mid b \in u_2 \wedge \varphi(b) = s\}|$$

Definition 8: Let $BP(\Sigma)$ be the set of the basic process sections of Σ and $S(P) \subseteq BP(\Sigma)$. We define

(1) ${}^0S(P) = \bigcup_{\forall P \in S(P)} {}^0P$, $S(P)^0 = \bigcup_{\forall P \in S(P)} P^0$ and

(2) $B({}^0S(P)) = \sum_{\forall P \in S(P)} B({}^0P)$, $B(S(P)^0) = \sum_{\forall P \in S(P)} B(P^0)$

Definition 9: Let $\Sigma = (S, T; F, M_0)$ be a Petri net. $\forall M \in R(M_0)$, $S_1 \subseteq S$, the projection of M on S_1 (denoted by $\Gamma_{S \to S_1}(M)$) is defined by $\Gamma_{S \to S_1}(M)(s) = M(s)$ for all $s \in S_1$.

Definition 10: Let φ be a mapping from the occurrence net $N = (B, E; G)$ to the Petri net $\Sigma = (S, T; F; M_0)$ and BP (Σ) be the set of the basic process sections. Petri net $\Sigma_p = (S_p, T_p; F_p, M_{0p})$ is defined as the process net system of Σ iff

(1) $S_P = {}^0BP(\Sigma) \cup BP(\Sigma)^0$;

(2) There is a one-to-one mapping ξ: $T_P \to BP(\Sigma)$ such that $\forall t \in T_P$, $\xi(t) \in BP(\Sigma)$;

(3) $F_P = \bigcup_{\forall t \in T_P} (\{t\} \times B(\xi(t)^0) \cup B({}^0\xi(t)) \times \{t\})$;

(4) $M_{op} = \Gamma_{S \to Sp}(M_0)$

PROCESS EXPRESSION OF PETRI NET

The process expressions of a bounded Petri net and unbounded fair Petri net are defined in Wu (1996) and Wu *et al.* (2000), respectively. The following definition for process expression of a Petri net is an extension of the cases in Wu (1996) and Wu *et al.* (2000).

Definition 11: Let $BP(\Sigma)$ be the set of the basic process sections of a Petri net $\Sigma = (S, T; F, M_0)$, Exp $(P(\Sigma))$ be an expression whose alphabet is isomorphic to $BP(\Sigma)$ and CRE $(P(\Sigma))$ be the set expressed by Exp $(P(\Sigma))$. Exp $(P(\Sigma))$ is defined as the process expression of Σ, if each surjective process of Σ is an element of the set

$$\mathrm{Pref}\{\mathrm{Exp}(P(\Sigma))\} = \bigcup_{P \in \mathrm{CRE}(P(\Sigma))} \mathrm{Pref}(P)$$

DECOMPOSITION BASED ON THE INDEX OF PLACE

Here, we introduce the decomposition method for a structure-complex Petri net based on the indexes of places (Zeng and Wu, 2002a, 2004a, b).

Definition 12: Let $\Sigma = (S, T; F, M_0)$ be a Petri net, a function f: $S \to \{1, 2, ...k\}$ is said to be an index function defined on the place set if $\forall s_1, s_2 \in S$,

$$(s_1^\bullet \cap s_2^\bullet \neq \phi) \vee ({}^\bullet s_1 \cap {}^\bullet s_2 \neq \phi) \to f(s_1) \neq f(s_2)$$

f (s) is named as the index of place s (Zeng and Wu, 2002a).

Definition 13: Let $\Sigma = (S, T; F, M_0)$ be a Petri net, f: $S \to \{1, 2, ..., k\}$ be the index function on the places of Σ. Petri net $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ $(i \in \{1, 2, ...,k\})$ is said to be the decomposition net of Σ based on the index function f if Σ_i satisfies the following conditions (Zeng and Wu, 2002a).

(1) $S_i = \{s \in S \mid f(s) = i\}$;

(2) $T_i = \{t \in T \mid \exists s \in S_i, t \in {}^\bullet s \cup s^\bullet\}$;

(3) $F_i = \{(S_i \times T_i) \cup (T_i \times S_i)\} \cap F$;

(4) $M_{0i} = \Gamma_{S \to S_i} M_0$

Simply, Σ_i is said as the index decomposition net of Σ.

More discussions about this decomposition method can be found in Zeng and Wu (2002a, 2004a) and an algorithm of decomposition for a structure-complex Petri net is also given in Zeng and Wu (2004a). The decomposition results with the method in definition 13 are usually not unique. With the results discussed in Zeng and Wu (2002a, 2004a), it is usually to take the case that k is with the minimal value as the best result.

Definition 14: A Petri net $\Sigma = (S, T; F, M_0)$ is an S-Net if $\forall t \in T$: $|{}^\bullet t| \leq 1$ and $|t^\bullet| \leq 1$ (Yuan, 2005).

Theorem 1: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ $(i \in \{1, 2, ...,k\})$ be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$, then Σ_i is an S-net (Zeng and Wu, 2002a).

Accroding to Theorem 1, each of the index decomposition nets is well-formed and it is easier to analyze the properties of the index decomposition nets than the original Petri net. This is one of the main reasons to decompose a structure-complex Petri net using this decomposition method.

Example 1: A Petri net Σ is shown in Fig. 1. We use the decomposition method in Definition 12 to decompose Σ.

A function f is first defined on the place set such that $f(s_1) = f(s_6) = 1$, $f(s_2) = f(s_3) = 2$, $f(s_4) = f(s_5) = f(s_7) = 3$. It

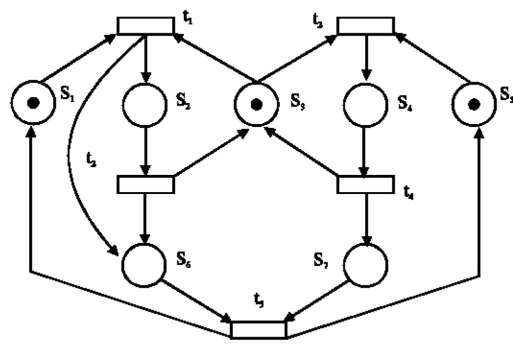


Fig. 1: A Petri net Σ

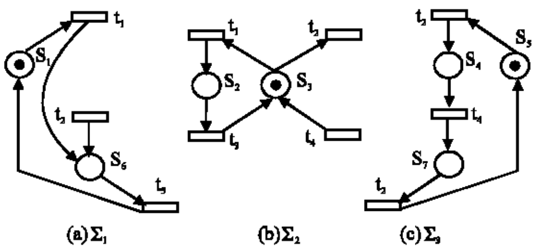


Fig. 2: Three S-Nets of Petri net Σ

can prove that f satisfies all the conditions in Definition 11. Based on the method of Definition 13, three index decomposition net systems Σ_1, Σ_2 and Σ_3 are obtained and shown in Fig. 2. Obviously, Σ_1, Σ_2 and Σ_3 are S-Nets.

PROCESS EXPRESSIONS OF THE DECOMPOSITION NET SYSTEMS

According to Theorem 1, each index decomposition net system is an S-Net. We have analyzed the language and liveness characteristics of an S-Net, respectively in Zeng and Wu (2002b) and Duan and Zeng (2004). Here, we analyze the process characteristics of an S-Net (an index decomposition net system) and present the method to obtain its process expression so as to construct the process expression of the original Petri net.

The following two propositions can be obtained easily following the related definitions.

Proposition 1: Let P = (N, φ) be a surjective process of an S-Net Σ, where N = (B, E; G). For any $e \in E$, $|{}^\bullet e|\le 1$ and $|e^\bullet|\le 1$.

Proposition 2: Let $\Sigma_P = (S_P, T_P; F_P, M_{0P})$ be the process net system of an S-Net $\Sigma = (S, T; F, M_0)$. Σ_P is also an S-Net.

Definition 15: Let $\Sigma = (S, T; F, M_0)$ be an S-Net. $\forall t\in T$, t is a primitive transition of Σ if $|{}^\bullet t| = 0$ and t is a goal transition of Σ if $|t^\bullet| = 0$.

In the index decomposition net, it is possible that there are primitive transitions or goal transitions. It is easy to transform an S-Net with goal transitions into one S-Net without goal transitions. The method is to add an output place for each goal transition in the original S-Net. If we delete all the added places from all the processes of the new S-Net without goal transitions, it can keep all states and behaviors of the original S-Net invariant. Thus, the process semantics can be regarded as equal between the original S-Net and the new S-Net without goal transitions. Therefore, it is assumed that for each transition t in the index decomposition net discussed in this research satisfies $|{}^\bullet t|\le 1$ and $|t^\bullet| = 1$.

All the S-Nets without goal transitions can be classified into four classes based on the standards of primitive transitions and initial markings. These four kinds of S-Nets, respectively are:

- S-Nets without primitive transitions and with empty markings. This kind of S-Nets has no any process, so it is not necessary to discuss.
- S-Nets without primitive transitions but with initial markings. It is obvious that this kind of S-Nets is S-graph (state machine). Without interpretations, this kind of S-Nets is named as S-graph directly in the following discussions.
- S-Nets with primitive transitions but with empty initial markings. Without interpretations, this kind of S-Nets is named as S-Net with Empty Markings directly in the following discussions.
- S-Nets with primitive transitions and with initial markings. Without interpretations, this kind of S-Nets is named as S-Net with Initial Markings directly in the following discussions.

Next, we will discuss the process characteristics of each kind of the S-Nets based on the classification and present the method to obtain their process expression.

PROCESS EXPRESSION OF S-GRAPH

Lemma 1: Each S-Graph is bounded (Yuan, 2005).

Lemma 2: The process expression of each S-Graph is a regular expression.

Proof: With the conclusions in Wu (1996), the process expression of a bounded Petri net is a regular expression.

Theorem 2: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ ($i\in\{1, 2, ..., k\}$) be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$. Exp $(P(\Sigma_i))$ is a regular expression if Σ_i is an S-Graph.

Example 2: Σ_3 shown in Fig. 2c is an S-Net. Based on its reachability graph, we can obtain the set of the basic process sections of Σ_3. The method to obtain the basic process sections can be seen in Wu (1996). $BP(\Sigma_3) = \{P_{31}\}$ and P_{31} is shown in Fig. 3. In the process section, we directly use the names of place and transition to label the corresponding elements and the same to the following discussions. The process expression of Σ_3 is Exp $(P(\Sigma_3)) = P_{31}^*$. It can be shown that each surjective process of Σ_3 is the prefix of the language of Exp $(P(\Sigma_3)) = P_{31}^*$.

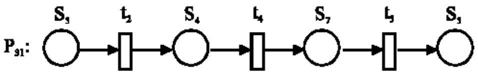


Fig. 3: The basic process section of Σ_3

PROCESS EXPRESSION OF S-NET WITH EMPTY MARKINGS

Lemma 3: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ ($i \in \{1, 2, ..., k\}$) be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$. If there is one and only one primitive transition $t' \in T_i$ and $\forall s \in S$: $M_0(s) = 0$ in Σ_i, Σ_i has one and only one basic source process section.

Proof: Obviously.

Lemma 4: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ ($i \in \{1, 2, ..., k\}$) be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$. If there is one and only one primitive transition $t' \in T_i$ and $\forall s \in S$: $M_0(s) = 0$ in Σ_i, there is a regular expression RE such that Exp $(P(\Sigma_3)) = (RE)^\alpha$, where $(RE)^\alpha$ is the α-closure of RE.

Proof: Suppose that $BP(\Sigma_i)$ is the set of the basic process sections of Σ_i. For any basic process in $BP(\Sigma_i)$, it is denoted by $P_j = (N[u_{1j}, u_{2j}], \varphi)$ where $1 \le j \le |BP(\Sigma_i)|$ and u_{1j} and u_{2j} are the first and last S-Cut of P_j. Based on $BP(\Sigma_i)$, the process net system $\Sigma_i = (S_{iP}, T_{iP}; F_{iP}, M_{0iP})$ of Σ_i can be constructed, where

(1) $$S_{iP} = \{u_{1j} \mid (N[u_{1j}, u_{2j}], \varphi) \in BP(\Sigma_i)\} \cup \{u_{2j} \mid (N[u_{1j}, u_{2j}], \varphi) \in BP(\Sigma_i)\},$$

(2) $T_{iP} = BP(\Sigma_i)$ and

(3) $F_{iP} = \{(u_{1j}, P_j) | (P_j \in BP(\Sigma_i)\} \cup \{(P_j, u_{2j}) | P_j \in BP(\Sigma_i)\}$, where $P_1 = (N[u_{11}, u_{21}], \varphi)$ is the source process section of Σ and $u_{11} = \phi$. So, $(u_{11}, P_1) \notin F_P$. and

(4) $\forall s \in S_P$: $M_{0P}(s) = 0$.

According to Lemma 1 and 2, Σ_{iP} is an S-Net with one and only one primitive transition and with empty markings, so L (Σ_{iP}) can be expressed by the α-closure of a regular expression. The proof can be seen in Zeng and Wu (2002b). With the definition of process net system, there is a one-to-one corresponding relation between Exp $(P(\Sigma_i))$ and L (Σ_{iP}). So, there is a regular expression RE such that Exp $(P(\Sigma_i)) = (RE)^\alpha$, where $(RE)^\alpha$ is the α- closure of RE.

Example 3: An S-Net Σ_{11} with one and only one primitive transition and has empty initial markings is shown in Fig. 4a and its two basic process sections are shown in Fig. 4b. The process expression of Σ_{11} is Exp $(P(\Sigma_{11})) = (P_{111}\, P_{112})^\alpha$, which can be obtained based on the process net system of Σ_{11}. It can be shown that each surjective process of Σ_{11} is the prefix of the language of Exp $(P(\Sigma_{11})) = (P_{111} \circ P_{112})^\alpha$, where $\circ$ is the connection operation between the basic process sections. The definitions and more discussions about operations between the basic process sections can be seen in Wu (1996).

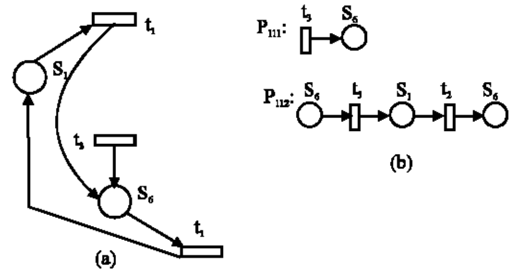


Fig. 4: An S-Net Σ_{11} and its two basic precess sections

Theorem 3: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ ($i \in \{1, 2,...,k\}$) be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$. If there are primitive transitions $t_1, ..., t_n$ ($n \ge 2$) and $\forall s \in S_i$ $M_0(s) = 0$ in Σ_i, there are regular expressions $RE_1, ..., RE_n$ such that Exp $(P(\Sigma_i)) = (RE_1)^\alpha || ... || (RE_n)^\alpha$.

Proof: Let $T_0 = \{t_1, ..., t_n\}$. To construct S-Nets $\Sigma_{ij} = (S_i, T_{ij}; F_{ij}, M_{0i})$ ($j \in \{1, 2, ..., n\}$) to prove the conclusion, where the place set S_i and initial marking M_{0i} keep invariant from $\Sigma_i = (S_i, T_i; F_j, M_{0i})$. $\Sigma_{ij} = (S_i, T_{ij}; F_{ij}, M_{0i})$ ($j \in \{1, 2, ..., n\}$) satisfies the following conditions,

(1) $T_{ij} = (T_i - T_o) \cup \{t_{ij}\}$ and

(2) $F_{ij} = F_i - (T_0 - \{t_{ij}\}) \times S_i$

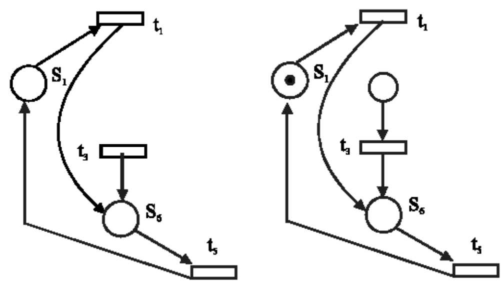
It is easy to prove Exp $(P(\Sigma_i))$ = Exp $(P(\Sigma_{i1})) || ... ||$ Exp (Σ_{in}).

For each $\Sigma_{ij} = (S_i, T_{ij}; F_{ij}, M_{0i})$ ($j \in \{1, 2, ..., n\}$), there is one and only one primitive transition and with empty initial markings. With Lemma 4, there is a regular expression RE_j such that Exp $(P(\Sigma_i)) = (RE_j)^\alpha$.

So, there are regular expressions $RE_1, ..., RE_n$ such that Exp $(P(\Sigma_i)) = (RE_1)^\alpha || ... || (RE_n)^\alpha$.

PROCESS EXPRESSION OF S-NET WITH INITIAL MARKINGS

Theorem 4: Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$ ($i \in \{1, 2, ..., k\}$), be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$. If there are primitive transitions $t_1, ..., t_n$ ($n \ge 1$ and $\{t_1, ..., t_n\} \subseteq T_i$) and at least one place $s \in S$ such that $M_{0i}(s) \ne 0$ in Σ_i, there are regular expressions $RE_1, ..., RE_n$ and RE_{n+1} such that Exp $(P(\Sigma_i)) = (RE_1)^\alpha || ... || (RE_n)^\alpha || RE_{n+1}$.



(a)Σ_{11} (b)Σ_{12}

Fig. 5: The basic process section of Σ_{12}

Proof: Let $T_0 = \{t_1, ..., t_n\}$. To construct two S-Nets $\Sigma_{ij} = (S_{ij}, T_{ij}; F_{ij}, M_{0ij})$ ($j \in \{1, 2\}$ to prove the conclusion, where,

(1) $S_{i1} = S_i$, $T_{i1} = T_i$, $F_{i1} = F_i$, $\forall s \in S_{i1}$: $M_{0i1}(s) = 0$ and
(2) $S_{i2} = S_i \cup \{s_0\}$, $T_{i2} = T_i$, $F_{i2} = F_i \cup (\{s_0\} \times T_0)$,

$$\forall s \in S_{i2} : M_{0i2}(s) = \begin{cases} 0 & \text{if } s = s_0 \\ M_{0i}(s) & \text{otherwise} \end{cases}$$

It is easy to prove Exp $(P(\Sigma_i))$ = Exp $(P(\Sigma_{i1}))$|| Exp $(P(\Sigma_{i2}))$ and Σ_{i1} satisfies all the conditions in Theorem 3 and Σ_{i2} is an S-graph. With Theorem 3, there are regular expressions $RE_1, ..., RE_n$ such that Exp $(P(\Sigma_{i1})) = (RE_1)^\alpha \| ... \| (RE_n)^\alpha$. With Theorem 1, Exp (Σ_{i2}) is a regular expression. So, there are regular expressions $RE_1, ..., RE_n$ and RE_{n+1} such that Exp $(P(\Sigma_i) = (RE_1)^\alpha \| ... \| (RE_n)^\alpha \| RE_{n+1}$.

Example 4: We take the index decomposition net Σ_1 shown in Fig. 2a as an example. Two new S-Nets Σ_{11} and Σ_{12} are constructed and shown in Fig. 5a and b, respectively. Σ_{11} is the S-Net shown in Fig. 4, so Exp $(P(\Sigma_{11})) = (P_{111} \circ P_{112})^\alpha$. Σ_{12} is an S-graph and Exp $(P(\Sigma_{12})) = P_{113}$*, where the basic process section P_{113} is shown in Fig. 6. So, the process expression of Σ_1 is Exp $(P(\Sigma_1))$ = Exp $(P(\Sigma_{11}))$ || Exp $(P(\Sigma_{12})) = (P_{111} \circ P_{112})^\alpha \| P_{113}$*.

PROJECTION AND SYNCHRONIZATION SHUFFLE OF PROCESSES

In order to analyze the process characteristics during the decomposition, the projection and synchronization shuffle operations of processes are defined in this section. Convenient to define, some notations are introduced first.

Let X be an alphabet. For any character $\alpha \in X$ and any word $\omega \in X^*$, $\#(\alpha, \omega)$ is the number of α occurring in ω.

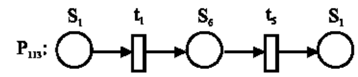


Fig. 6: The basic process section of Σ_{12}

For any language $L \subseteq X^*$, $\delta(L) = \{x \in X | \exists \omega \in L: \#(x, \omega) > 0\}$. For example, $\delta(a+ab) = \delta(aa+abbbb) = \{a, b\}$. Obviously, $\delta(L) \subseteq X$.

Definition 16: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ $(i \in \{1, 2,..., k\})$ be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$ and SN(*) be the set of occurrence net, where $* \in \{\Sigma, \Sigma_1, ..., \Sigma_k\}$. For $\forall N \in SN(\Sigma)$, if Π: $\varphi(SN(\Sigma)) \rightarrow \varphi_i(SN(\Sigma_i))$ satisfies the following conditions,

(1) $\forall b \in B$, to delete b and its input and output arcs if $\varphi(b) \in S_j$ and $j \neq i$ and
(2) $\forall e \in E$, to delete e and its input and output arcs if $\varphi(e) \in E_j$-E_i and $j \neq i$,

Π is the projection from $\varphi(SN(\Sigma))$ to $\varphi_i(SN(\Sigma_i))$ and denoted by $\Pi_{\Sigma \rightarrow \Sigma_i}$.

For any basic process section, it is one part of the occurrence net of a Petri net. If SN(*) in Definition 16 is extended to contain all the basic process sections and other conditions are invariant, the projection of basic process section can be also obtained.

Definition 17: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ $(i \in \{1, 2\})$ be two Petri nets and Exp $(P(\Sigma_i))$ be the process expression of Σ_i. Exp $(P(\Sigma_1))\Theta$Exp $(P(\Sigma_2))$ is the synchronization shuffle of Exp $(P(\Sigma_1))$ and Exp $(P(\Sigma_2))$ iff:

$$\mathrm{Exp}(\Sigma_1)\Theta\mathrm{Exp}(\Sigma_2) = \{\omega \in \delta(\mathrm{Exp}(\Sigma_\Delta))^* \mid \delta(\mathrm{Exp}(P(\Sigma_\Delta))) = \delta(\mathrm{Exp}(P(\Sigma_1))) \cup \delta(\mathrm{Exp}(P(\Sigma_2))),\ \Pi_{\Sigma_\Delta \rightarrow \Sigma_i}(\omega) \in \mathrm{Exp}(P(\Sigma_i))\}$$

Definition 17 only presents the synchronization shuffle of two process expressions, which can be extended into the case of $k(k \geq 2)$ expressions, denoted by $\underset{1 \leq i \leq k}{\Theta} \mathrm{Exp}(P(\Sigma_i))$. It can be defined by:

$$\underset{1 \leq i \leq k}{\Theta} \mathrm{Exp}(P(\Sigma_i)) = \mathrm{Exp}(P(\Sigma_1))\Theta\mathrm{Exp}(P(\Sigma_2))\Theta \cdots \Theta\mathrm{Exp}(P(\Sigma_k)) = ((\mathrm{Exp}(P(\Sigma_1))\Theta\mathrm{Exp}(P(\Sigma_2)))\Theta \cdots \Theta\mathrm{Exp}(P(\Sigma_k)))$$

PROCESS CHARACTERISTIC ANALYSIS DURING DECOMPOSITION

Lemma 5: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ $(i \in \{1, 2, ..., k\})$ be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$ and BP(*) be the set of the basic process sections of *, where $* \in \{\Sigma, \Sigma_1, ..., \Sigma_k\}$. For each $BP \in BP(\Sigma)$, $\Pi_{\Sigma \rightarrow \Sigma_i} BP \in BP(\Sigma_i)$.

Proof: It is easy to prove.

Lemma 5: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ ($i\in\{1, 2\}$) be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$. $Exp(P(\Sigma)) = Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$.

Proof: First with $Exp(P(\Sigma))\subset Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$ and then $Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))\subset Exp(P(\Sigma))$.

Now, we prove that $Exp(P(\Sigma))\subset Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$.

For any $\varphi(N)\in Exp(P(\Sigma))$, let $N = (B, E; G)$. Since $\varphi(N)$ is a process of Σ, according to the definition of $\Pi_{\Sigma\to\Sigma_i}$,

(1)
$\Pi_{\Sigma\to\Sigma_i}(\varphi(B)) \subseteq S_i \wedge \Pi_{\Sigma\to\Sigma_i}(\varphi(E)) \subseteq T_i \wedge \Pi_{\Sigma\to\Sigma_i}(\varphi_i(G)) \subseteq F_i$ and

(2)
For any $e \in \Pi_{\Sigma\to\Sigma_i}(E) \subseteq E$, since $\varphi({}^\bullet e) = {}^\bullet\varphi(e) \wedge \varphi(e^\bullet) = \varphi(e)^\bullet, \varphi({}^\bullet e) \cap S_i = {}^\bullet\varphi(e) \cap S_i \wedge \varphi(e^\bullet) \cap S_i = \varphi(e)^\bullet \cap S_i$ and $\Pi_{\Sigma\to\Sigma_i}\varphi({}^\bullet\Pi_{\Sigma\to\Sigma_i}(e)) = {}^\bullet(\Pi_{\Sigma\to\Sigma_i}\varphi(\Pi_{\Sigma\to\Sigma_i}(e))) \wedge \Pi_{\Sigma\to\Sigma_i}\varphi(\Pi_{\Sigma\to\Sigma_i}(e)^\bullet) = (\Pi_{\Sigma\to\Sigma_i}\varphi(\Pi_{\Sigma\to\Sigma_i}(e)))^\bullet$

(3)
For any $b_1, b_2 \in \Pi_{\Sigma\to\Sigma_i}(B) \subseteq B$, ${}^\bullet b_1 \neq {}^\bullet b_2 \wedge b_1^\bullet \neq b_2^\bullet$ since $\varphi(b_1) = \varphi(b_2)$. So, $\varphi(b_1) \cap P_i = \varphi(b_2) \cap P_i$ and ${}^\bullet b_1 \cap T_i \neq {}^\bullet b_2 \cap T_i \wedge b_1^\bullet \cap T_i \neq b_2^\bullet \cap T_i$. With $\Pi_{\Sigma\to\Sigma_i}\varphi(\Pi_{\Sigma\to\Sigma_i}(b_1)) = \Pi_{\Sigma\to\Sigma_i}\varphi(\Pi_{\Sigma\to\Sigma_i}(b_2))$, ${}^\bullet\Pi_{\Sigma\to\Sigma_i}(b_1) \neq {}^\bullet\Pi_{\Sigma\to\Sigma_i}(b_2) \wedge \Pi_{\Sigma\to\Sigma_i}(b_1)^\bullet \neq \Pi_{\Sigma\to\Sigma_i}(b_2)^\bullet$ and

(4)
For any $s \in S_i \subseteq S$, $|\{b \mid {}^\bullet b \neq \phi \wedge \varphi(b) = s\}| = M_0(s)$, so $|\{\Pi_{\Sigma\to\Sigma_i}(b) \mid {}^\bullet(\Pi_{\Sigma\to\Sigma_i}(b)) \neq \phi \wedge \Pi_{\Sigma\to\Sigma_i}\varphi(b) = s\}| = M_{0i}(s)$, where $M_{0i} = \Gamma_{S\to S_i}(M_0)$

Thus, $\Pi_{\Sigma\to\Sigma_i}\varphi(N) \in Exp(P(\Sigma_i))(i \in \{1,2\})$. With the definition of Θ, $\varphi(N)\in Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$ and $Exp(P(\Sigma))\subset Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$.

Now, we prove that $Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))\subset Exp(P(\Sigma))$.

For any $\varphi(N)\in Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$, let $|\varphi(N)|$ be the number of the basic process sections contained by $\varphi(N)$. The conclusion can be proved by induction on $|\varphi(N)|$.

(1) While $|\varphi(N)| =1$, $\varphi(N)$ is a basic process section and $\Pi_{\Sigma\to\Sigma_i}(\varphi(N)) \in BP(\Sigma_i)$. With Lemma 5, $\varphi(N) \in BP(P(\Sigma)) \subset Exp(P(\Sigma))$, so $Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))\subset Exp(P(\Sigma))$.

(2) Suppose that the conclusion is correct while $|\varphi(N)|\le k$, i.e., $Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))\subseteq Exp(P(\Sigma))$.

Now, we prove it is also correct while $\widehat{\varphi(N)} \in Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$ and $|\widehat{\varphi(N)}| = k+1$. With the definition of process expression, there are $\varphi(N)\in Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$ and $\overline{\varphi(N)}$ such that $\widehat{\varphi(N)} = \varphi(N) \circ \overline{\varphi(N)}$, where o is the connection operation between basic process sections and $|\varphi(N)| = k$ and $|\overline{\varphi(N)}| = 1$. Now, it is proved by two cases of $\overline{\varphi(N)}$.

Case 1: If $\overline{\varphi(N)} \in BP(\Sigma_i) - BP(\Sigma_j)$, where $i, j\in\{1, 2\}$ and $i \neq j$, then

(a) $\widehat{\varphi(B)} = \varphi(B) \circ \overline{\varphi(B)}$ and $\varphi(B) \circ \overline{\varphi(B)} \subseteq S_1 \cup S_2 \cup S_i$, so $\widehat{\varphi(B)} \subseteq S_1 \cup S_2 \cup S_i$, i.e., $\widehat{\varphi(B)} \subseteq P_1 \cup P_2$ and $\widehat{\varphi(E)} = \varphi(E) \circ \overline{\varphi(E)}$ and $\varphi(E) \circ \overline{\varphi(E)} \subseteq T_1 \cup T_2 \cup T_i$, so $\widehat{\varphi(E)} \subseteq T_1 \cup T_2 \cup T_i$, i.e., $\widehat{\varphi(E)} \subseteq T_1 \cup T_2$ and $\widehat{\varphi(G)} = \varphi(G) \circ \overline{\varphi(G)}$ and $\varphi(G) \circ \overline{\varphi(G)} \subseteq F_1 \cup F_2 \cup F_i$, so $\widehat{\varphi(G)} \subseteq F_1 \cup F_2 \cup F_i$, i.e., $\widehat{\varphi(G)} \subseteq F_1 \cup F_2$.

(b) For any $\hat{e} \in \hat{E}$ such that $\hat{E} = E \cup \overline{E}$, while $\hat{e} \in E$, $\varphi({}^\bullet\hat{e}) = {}^\bullet\varphi(\hat{e})$ and $\varphi(\hat{e}^\bullet) = \varphi(\hat{e})^\bullet$ with the supposition. While $\hat{e} \in \overline{E}$, $\overline{\varphi}({}^\bullet\hat{e}) = {}^\bullet\overline{\varphi}(\hat{e})$ and $\overline{\varphi}(\hat{e}^\bullet) = \overline{\varphi}(\hat{e})^\bullet$ since $\overline{\varphi(N)} \in BP(\Sigma_i) - BP(\Sigma_j)$. So $\varphi({}^\bullet\hat{e}) = {}^\bullet\varphi(\hat{e})$ and $\varphi(\hat{e}^\bullet) = \varphi(\hat{e})^\bullet$

(c) For any $\hat{b}_1, \hat{b}_2 \in \hat{B}$ such that $\hat{B} = B \cup \overline{B}$, it can be proved $\widehat{\varphi(b_1)} = \widehat{\varphi(b_2)} \Rightarrow {}^\bullet\hat{b}_1 \neq {}^\bullet\hat{b}_2 \wedge \hat{b}_1^\bullet \neq \hat{b}_2^\bullet$ with a similar method of (b)

(d) For any $s \in S$ such that $S = S_1 \cup S_2$, $|\{b \mid {}^\bullet b = \phi \wedge \varphi(b) = s\}| = M_0(s)$ i.e., $|\{b \mid {}^\bullet\hat{b} = \phi \wedge \widehat{\varphi(b)} = s\}| = M_0(s)$.

With (1a – d), $\overline{\varphi(N)} \in Exp(P(\Sigma))$. So, $Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2)) \subseteq Exp(P(\Sigma))$.

Case 2: If $\overline{\varphi(N)} \in BP(\Sigma_1) \cap BP(\Sigma_2)$, then

Since $\widehat{\varphi(B)} = \varphi(B) \circ \overline{\varphi(B)}$ and $\varphi(B) \circ \overline{\varphi(B)} \subseteq S_1 \cup S_2 \cup S_1 \cup S_2$, $\widehat{\varphi(B)} \subseteq S_1 \cup S_2 \cup S_1 \cup S_2$ and $\widehat{\varphi(B)} \subseteq S_1 \cup S_2$

$$\text{Since } \widehat{\varphi(E)} = \varphi(E) \circ \overline{\varphi(E)} \text{ and } \varphi(E) \circ \overline{\varphi(E)} \subseteq T_1 \cup T_2 \cup T_1 \cup T_2, \widehat{\varphi(E)} \subseteq T_1 \cup T_2 \cup T_1 \cup T_2 \text{ and } \widehat{\varphi(E)} \subseteq T_1 \cup T_2$$

$$\text{Since } \widehat{\varphi(G)} = \varphi(G) \circ \overline{\varphi(G)} \text{ and } \varphi(G) \circ \overline{\varphi(G)} \subseteq F_1 \cup F_2 \cup F_1 \cup F_2, \widehat{\varphi(G)} \subseteq F_1 \cup F_2 \cup F_1 \cup F_2 \text{ and } \widehat{\varphi(G)} \subseteq F_1 \cup F_2$$

Other parts can be proved like the cases in Case 1. So $\overline{\varphi(N)} \in Exp(P(\Sigma))$ i.e., $Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2)) \subseteq Exp(P(\Sigma))$

With Case 1 and 2, $Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2)) \subseteq Exp(P(\Sigma))$ is proved based on the supposition.

$Exp(P(\Sigma)) = Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$ has been proved since $Exp(P(\Sigma)) \subseteq Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2))$ and $Exp(P(\Sigma_1))\Theta Exp(P(\Sigma_2)) \subseteq Exp(P(\Sigma))$.

Theorem 5: Let $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ $(i \in \{1, 2, ..., k\})$ be the index decomposition net of a Petri net $\Sigma = (S, T; F, M_0)$. $Exp(P(\Sigma)) = \underset{1 \le i \le k}{\Theta} Exp(P(\Sigma_i))$.

Proof: Based on Lemma 5, the theorem can be proved by induction on k.

Theorem 5 is useful to obtain the process expression of a Petri net by the operation Θ between processes based on the process expressions of the index decomposition S-Nets. The method will be presented in next section.

A CONSTRUCTION METHOD FOR THE PROCESS EXPRESSION OF PETRI NET

Using the decomposition method, a Petri net Σ can be decomposed into a set of S-Nets $\Sigma_1, \Sigma_2, ..., \Sigma_k$. The methods to obtain the process expressions of all kinds of S-Nets are presented in earlier. Theorem 5 indicates the relation between the process expression of the original Petri net and its index decomposition net systems. $\underset{1 \le i \le k}{\Theta} Exp(P(\Sigma_i))$ is the process expression of Σ, so a method to obtain the process expression of a Petri net can be presented, which is shown in Algorithm 1.

Algorithm 1 presents a method to construct the process expression of a given Petri net. The correctness and termination can be proved by the analysis in the last sections. It benefits the semantic expression of the process of a Petri net especially a structure-complex net system. This method adopts a top to down process, in which a Petri net is decomposed into a set of S-Nets since the process expression of each S-Net is easy to obtain. Based on the process expressions of these S-Nets, the process expression of the original Petri net can be expressed by the operation of Θ between processes.

Example 5: We take the Petri net Σ shown in Fig. 1 as an example to show the method proposed in Algorithm 1.

The first step in Algorithm 1 is to decompose Σ into a set of S-Nets, which has been discussed earlier and the index decomposition sets are shown in Fig. 2.

The second step is to obtain the process expression of each S-Net. Σ_1 has been discussed in example 4 and its process expression is $Exp(P(\Sigma_1)) = (P_{111} \circ P_{112})^{\alpha} \| P_{113}{*}$. Σ_3 has been discussed in example 2 and its process expression is $Exp(P(\Sigma_3)) = P_{31}{*}$.

Now, it is only necessary to obtain the process expression of Σ_2. There is a primitive transition t_4 and a goal transition t_2 in Σ_2. First, Σ_2 is transformed into an S-Net Σ'_2 without goal transition. Σ'_2 is an S-Net satisfying the conditions in Theorem 4, so two new S-Nets are constructed first, which are Σ'_{21} and Σ'_{22} shown in Fig. 7. Σ'_{21} is an S-graph and Σ'_{22} is an S-Net with one and only one primitive transition and with empty initial markings. According to methods, as mentioned

Algorithm 1: To obtain the process expression of a Petri net

Input: A Petri net $\Sigma = (S, T; F, M_0)$
Output: The process expression $Exp(P(\Sigma))$ of Σ
(1) Decompose Σ into a set of S-Nets $\Sigma_1, \Sigma_2, ..., \Sigma_k$. The main steps include,
(1a) Define a function f: $S \to \{1, 2, ..., k\}$ such that
$\forall s_1, s_2 \in S, (s_1^{\bullet} \cap s_2^{\bullet} \neq \phi) \vee ({}^{\bullet}s_1 \cap {}^{\bullet}s_2 \neq \phi) \to f(s_1) \neq f(s_2)$

(1b) Decompose Σ into $\Sigma_1, \Sigma_2, ..., \Sigma_k$ based on f. For each $\Sigma_i = (S_i, T_i; F_i, M_{0i})$, it satisfies
$S_i = \{s \in S | f(s) = i\}$; $T_i = \{t \in T | \exists s \in S_i, t \in {}^{\bullet}s \cup s^{\bullet}\}$; $F_i = ((S_i \times T_i) \cup (T_i \times S_i)) \cap F$ and $M_{0i} = \Gamma_{s \to s_i} M_0$

(1c) $\Sigma_i = (S_i, T_i; F_i, M_{0i})$ $(i \in \{1, 2, ..., k\})$ is the index decomposition S-Nets.
(2) Obtain the process expression $Exp(P(\Sigma_i))$ of each S-Net.
(3) Output $Exp(P(\Sigma)) \leftarrow \underset{1 \le i \le k}{\Theta} Exp(P(\Sigma_i))$.

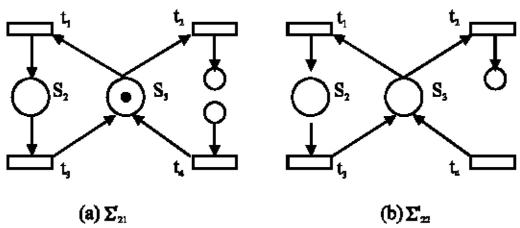


Fig. 7: Two new S-Nets Σ'_{21} and Σ'_{22}

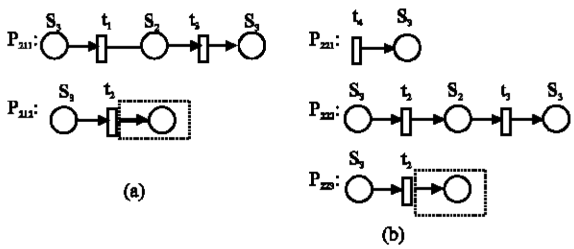


Fig. 8: The basic process section of Σ_{123}

earlier the process expression of Σ'_{21} is $\text{Exp}(P(\Sigma'_{21})) = P_{211}{*}\circ P_{212}$, where the basic process sections P_{211} and P_{212} are shown in Fig. 8a. The part in P_{212} containing by a broken-line quadrangle should be deleted, since it is added in order to reduce the goal transition. The process expression of Σ'_{22} is $\text{Exp}(P(\Sigma'_{22})) = (P_{221}\circ P_{222} \circ P_{223})^{\alpha}$. Similarly, the part in P_{223} contained by a broken-line quadrangle should also be deleted. So, the process expression of Σ'_2 is:

$$\text{Exp}(P(\Sigma_2)) = \text{Exp}(P(\Sigma'_{21}))\|\,\text{Exp}(P(\Sigma'_{22})) = (P_{211}{*} \circ P_{212})\|\,(P_{221} \circ P_{222} \circ P_{223})^{\alpha}$$

The third step in Algorithm 1 is to output the process expression of Petri net Σ,

$$\begin{aligned}\text{Exp}(P(\Sigma)) &= \text{Exp}(P(\Sigma_1))\Theta\text{Exp}(P(\Sigma_2))\Theta\,\text{Exp}(P(\Sigma_3))\\ &= (P_{111} \circ P_{112})^{\alpha} \,\|\, P_{113} * \Theta(P_{211} * \circ P_{212})\\ &\quad \|\,(P_{221} \circ P_{222} \circ P_{223})^{\alpha}\Theta P_{31} *\end{aligned}$$

CONCLUSION

A construction method for the process expression of a Petri net is proposed based on the index decomposition. With this decomposition method, a Petri net is decomposed into a set of S-Nets. The process properties of the S-Nets are easy to analyze since they are well-formed and their process expressions are easier to obtain than the original Petri net. With the process expressions of the index decomposition nets, the process expression of the original Petri net can be presented by the synchronization shuffle operation between processes.

There have been lots of works to obtain the process expression of a Petri net (Wu, 1996; Wu *et al*., 2000; Zeng and Wu, 2003). The method introduce in Wu (1996) only suits the bounded Petri net and the method of Wu *et al*. (2000) only suits the unbounded but fair Petri net. Zeng and Wu (2003) presents the method can be used for all kinds of Petri nets, however it is necessary to obtain the process net system first. Since the structure of the process net system is usually complex, it must decompose the process net system so as to obtain its language expression. The process of the method in Zeng and Wu (2003) is very complex. Compared with the related work, the construction method proposed in this work is easier to realize and it is useful for obtaining the process expressions of all kinds of Petri nets.

The semantics of the process expression containing Θ is not obvious. In the future, the reduction and optimization for the process expression containing Θ must be considered. The applications of the process expression for analyzing the properties such as liveness and fairness are also research work in the future.

ACKNOWLEDGMENTS

Supported by the National Science Foundation of China under Grant No.60603090 and 90718011, the Excellent Young Scientist Foundation of Shandong Province of China under Grant No. 2006BS01019 and the Taishan Scholar Program of Shandong Province.

REFERENCES

Duan, H. and Q.T. Zeng, 2004. Liveness analysis for S-net. Mini-Micro Syst., 25: 1975-1978.

Garg, V.K. and M.T. Ragunath, 1992. Concurrent regular expressions and their relationship to Petri nets. Theor. Comput. Sci., 96: 258-304.

Lu, R.Q., 1992a. P/R nets and P/R processes (I). Science in China (Series E), pp: 35: 21-31.

Lu, R.Q., 1992b. P/R nets and P/R processes (II). Science in China (Series E), 35: 148-157.

Murata, T., 1989. Petri Nets: Properties, analysis and applications. Proceedings IEEE, pp: 77.

Wu, Z.H., 1996. Process expression of bounded petri net. Sci. China (Series E), 39: 37-49.

Wu, Z.H., P.L. Wang and M.X. Zhao, 2000. Process expression of unbounded fair Petri net. Chin. J. Comput., 23: 337-344.

Yuan, C.Y., 2005. Principle and Application of Petri Net. Beijing: Publishing House of Electronics Industry.

Zeng, Q.T. and Z.H. Wu, 2002a. Decomposition method of petri net based on the index of places. Comput. Sci., 29: 15-17.

Zeng, Q.T. and Z.H. Wu, 2002b. The language characteristics analysis of analogous S-graph. Comput. Sci., 29: 120-122.

Zeng, Q.T. and Z.H. Wu, 2002c. Process net system of petri net. Chin. J. Comput., 25: 1308-1315.

Zeng, Q.T. and Z.H. Wu, 2003. Process expression of unbounded Petri net. Chin. J. Comput., 26: 1629-1636.

Zeng, Q.T. and Z.H.Wu, 2004a. Methods for behavior descriptions of structure-complex petri nets based on decomposition. J. Control Theor. Appl., 2: 93-98.

Zeng, Q.T. and Z.H. Wu, 2004b. Invariant decomposition conditions for Petri net. Mini-Micro Syst., 25: 1671-1675.