

Chapter 2

Petri Nets

Abstract This chapter presents a mathematical treatment of Petri nets, including their formal definitions, structural and behavioral properties such as invariants, siphons, traps, reachability graphs, and state equations that are necessary to understand the subjects presented in this book. A number of important subclasses of Petri nets are introduced such as state machines and marked graphs. They are essential for the development of manufacturing-oriented Petri net models and the deadlock control strategies. The basics of automata are also covered in this chapter to facilitate the reader to understand well the deadlock prevention policy based on theory of regions. The concepts of a plant model, supervisor, and controlled system are defined.

2.1 Introduction

Though Petri nets and automata lack the full modeling and decision power of Turing machines, they still rank the top popular modeling tools for DES. As for Petri nets, this is partially attributed to their capability to provide the simple, direct, faithful, and convenient graphical representation of DES. Moreover, the well-established set of mathematical approaches employing linear matrix algebra makes them particularly useful for the modeling, analysis, and control of DES [44]. This chapter presents a mathematical treatment of Petri net theory. It is fundamental for understanding of the ideas presented in the following chapters.

2.2 Formal Definitions

A Petri net is a directed bipartite graph. It consists of two components: a net structure and an initial marking. A net (structure) contains two sorts of nodes: places and transitions. There are directed arcs from places to transitions and directed arcs from

transitions to places in a net. Places are graphically represented by circles and transitions by boxes or bars. A place can hold tokens denoted by black dots, or a positive integer representing their number. The distribution of tokens over the places of a net is called a marking that corresponds to a state of the modeled system. The initial token distribution is hence called the initial marking. Let $\mathbb{N}$ denote the set of non-negative integers and $\mathbb{N}^+$ the set of positive integers.

Definition 2.1. A generalized Petri net (structure) is a 4-tuple $N = (P, T, F, W)$ where P and T are finite, non-empty, and disjoint sets. P is the set of places and T is the set of transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ iff $(x, y) \in F$, and $W(x, y) = 0$ otherwise, where $x, y \in P \cup T$.

Definition 2.2. A marking M of a Petri net N is a mapping from P to $\mathbb{N}$. $M(p)$ denotes the number of tokens in place p . A place p is marked by a marking M iff $M(p) > 0$. A subset $S \subseteq P$ is marked by M iff at least one place in S is marked by M . The sum of tokens of all places in S is denoted by $M(S)$, i.e., $M(S) = \sum_{p \in S} M(p)$. S is said to be empty at M iff $M(S) = 0$. (N, M_0) is called a net system or marked net and M_0 is called an initial marking of N .

We usually describe markings and vectors using a multiset (bag) or formal sum notation for economy of space. As a result, $\sum_{p \in P} M(p)p$ is used to denote vector M . For instance, a marking that puts four tokens in place p_2 and two tokens in place p_4 only in a net with $P = \{p_1 - p_6\}$ is denoted by $4p_2 + 2p_4$ instead of $(0, 4, 0, 2, 0, 0)^T$.

In general, (N, M_0) is directly called a net where there is no confusion. $N = (P, T, F, W)$ is called an ordinary net, denoted by $N = (P, T, F)$, if $\forall f \in F, W(f) = 1$. Note that ordinary and generalized Petri nets have the same modeling power. The only difference is that the latter may have improved modeling efficiency and convenience for some systems. For convenience, (P, T, F, W, M_0) is sometimes used to denote a marked net. It is also called a net system.

Example 2.1. Figure 2.1a shows a simple Petri net with $P = \{p_1 - p_5\}$, $T = \{t_1 - t_3\}$, $F = \{(p_1, t_1), (t_3, p_1), (p_2, t_2), (t_1, p_2), (p_3, t_3), (t_2, p_3), (p_4, t_2), (t_3, p_4), (p_5, t_1), (p_5, t_2), (t_3, p_5)\}$, $W(p_1, t_1) = W(t_3, p_1) = W(p_2, t_2) = W(t_1, p_2) = W(p_3, t_3) = W(t_2, p_3) = W(p_4, t_2) = W(t_3, p_4) = W(p_5, t_1) = 1$, $W(p_5, t_2) = 2$, and $W(t_3, p_5) = 3$. Places are graphically represented by circles and transitions are represented by boxes. It is clear that the net is not ordinary because of the multiplicity of arcs (p_5, t_2) and (t_3, p_5) .

Each of places p_1 and p_5 has three tokens, denoted by three black dots or number 3 inside. Place p_4 holds two tokens and there is no token in p_2 and p_3 . This token distribution leads to the initial marking of the net with $M_0 = 3p_1 + 2p_4 + 3p_5$. The net's alternative graphical representation is given in Fig 2.1b, where multiple arcs are replaced with an arc with its weight and multiple tokens in a place can be replaced by a corresponding number for the sake of simplicity. For example, the number of tokens in place p_1 is denoted by number 3.

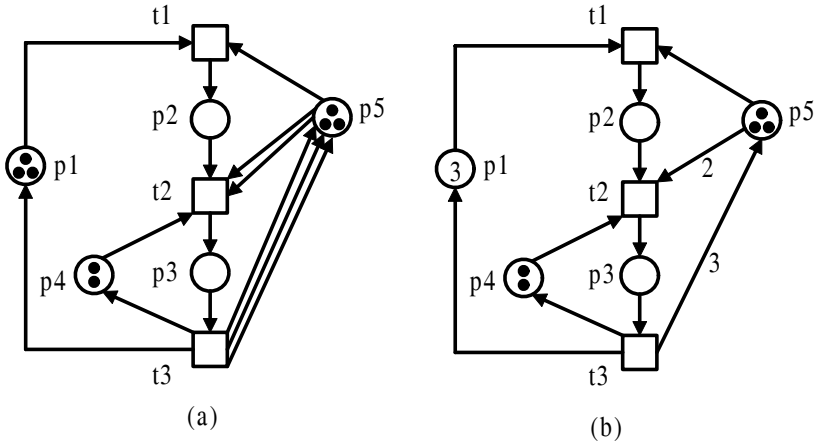


Fig. 2.1 A Petri net (N, M_0) with $M_0 = 3p_1 + 2p_4 + 3p_5$ represented by (a) multiplicity of arcs and (b) weight of arcs

Definition 2.3. Let $x \in P \cup T$ be a node of net $N = (P, T, F, W)$. The preset of x is defined as $\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$. While the postset of x is defined as $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$. This notation can be extended to a set of nodes as follows: given $X \subseteq P \cup T$, $\bullet X = \bigcup_{x \in X} \bullet x$, and $X^\bullet = \bigcup_{x \in X} x^\bullet$. Given place p , we denote $\max\{W(p, t) \mid t \in p^\bullet\}$ by $\max_{p^\bullet}$.

For $t \in T$, $p \in \bullet t$ is called an input place of t and $p \in t^\bullet$ is called an output place of t . For $p \in P$, $t \in \bullet p$ is called an input transition of p and $t \in p^\bullet$ is called an output transition of p .

Example 2.2. In Fig. 2.1a, we have $\bullet t_1 = \{p_1, p_5\}$, $\bullet t_2 = \{p_2, p_4, p_5\}$, $t_2^\bullet = \{p_3\}$, $t_3^\bullet = \{p_1, p_4, p_5\}$, $\bullet p_3 = \{t_2\}$, $p_3^\bullet = \{t_3\}$, $\bullet p_5 = \{t_3\}$, and $p_5^\bullet = \{t_1, t_2\}$. Let $S = \{p_3, p_5\}$. Then, $\bullet S = \bullet p_3 \cup \bullet p_5 = \{t_2, t_3\}$ and $S^\bullet = p_3^\bullet \cup p_5^\bullet = \{t_1, t_2, t_3\}$. It is easy to see that $\max_{p_5^\bullet} = 2$ and $\forall p \in P \setminus \{p_5\}$, $\max_{p^\bullet} = 1$.

Definition 2.4. A transition $t \in T$ is enabled at a marking M iff $\forall p \in \bullet t$, $M(p) \geq W(p, t)$. This fact is denoted by $M[t]$. Firing it yields a new marking M' such that $\forall p \in P$, $M'(p) = M(p) - W(p, t) + W(t, p)$, as denoted by $M[t]M'$. M' is called an immediately reachable marking from M . Marking M'' is said to be reachable from M if there exists a sequence of transitions $\sigma = t_0 t_1 \cdots t_n$ and markings $M_1, M_2, \dots$, and M_n such that $M[t_0]M_1[t_1]M_2 \cdots M_n[t_n]M''$ holds. The set of markings reachable from M in N is called the reachability set of Petri net (N, M) and denoted by $R(N, M)$.

Example 2.3. In Fig. 2.2a, t_1 is enabled at initial marking $M_0 = 3p_1 + 2p_4 + 3p_5$ since $\bullet t_1 = \{p_1, p_5\}$, $M_0(p_1) = 3 > W(p_1, t_1) = 1$, and $M_0(p_5) = 3 > W(p_5, t_1) = 1$. Firing t_1 leads to M_1 with $M_1(p_1) = M_0(p_1) - W(p_1, t_1) + W(t_1, p_1) = 2$, $M_1(p_2) = M_0(p_2) - W(p_2, t_1) + W(t_1, p_2) = 1$, $M_1(p_3) = M_0(p_3) - W(p_3, t_1) + W(t_1, p_3) = 0$,

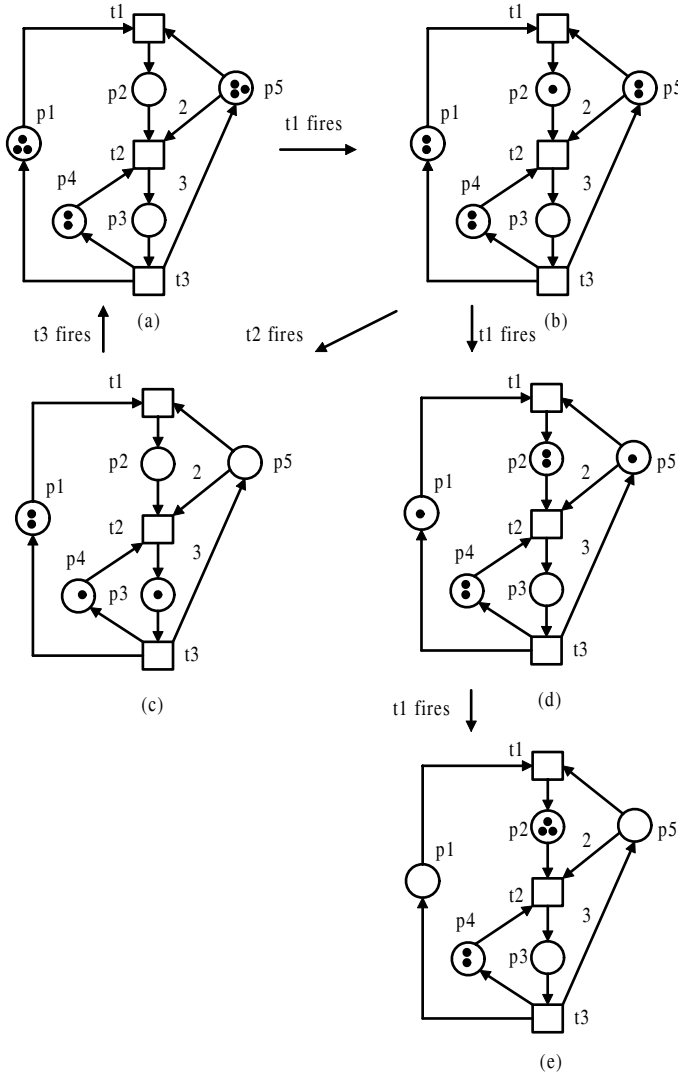


Fig. 2.2 The evolution of a Petri net: (a) (N, M_0) , (b) (N, M_1) , (c) (N, M_2) , (d) (N, M_3) , and (e) (N, M_4)

$M_1(p_4) = M_0(p_4) - W(p_4, t_1) + W(t_1, p_4) = 2$, and $M_1(p_5) = M_0(p_5) - W(p_5, t_1) + W(t_1, p_5) = 2$, as shown in Fig 2.2b.

In marking M_1 , both t_1 and t_2 are enabled. Firing t_2 at M_1 leads to M_2 as shown in Fig. 2.2c. Firing t_1 at M_1 leads to M_3 as shown in Fig. 2.2d. Only t_1 is enabled at M_3 . Figure 2.2e is the net after t_1 fires at M_3 and corresponds to M_4 . At M_2 , only t_3 is enabled. Firing it leads back to M_0 . As a result, the reachability set of the net in Fig. 2.2a is $R(N, M_0) = \{M_0, M_1, M_2, M_3, M_4\}$, where $M_0 = 3p_1 + 2p_4 + 3p_5$,

$M_1 = 2p_1 + p_2 + 2p_4 + 2p_5$, $M_2 = 2p_1 + p_3 + p_4$, $M_3 = p_1 + 2p_2 + 2p_4 + p_5$, and $M_4 = 3p_2 + 2p_4$. Note that at M_4 , no transition is enabled.

Definition 2.5. A Petri net (N, M_0) is safe if $\forall M \in R(N, M_0)$, $\forall p \in P$, $M(p) \leq 1$ is true. It is bounded if $\exists k \in \mathbb{N}^+$, $\forall M \in R(N, M_0)$, $\forall p \in P$, $M(p) \leq k$. It is said to be unbounded if it is not bounded. A net N is structurally bounded if it is bounded for any initial marking.

Note that a net is bounded iff its reachability set has a finite number of elements. The reachability set of a net (N, M_0) can be expressed by a reachability graph. A reachability graph is a directed graph whose nodes are markings in $R(N, M_0)$ and arcs are labeled by the transitions of N . An arc from M_1 to M_2 is labeled by t iff $M_1[t]M_2$.

Example 2.4. Figure 2.3 shows the reachability graph of the Petri net depicted in Fig. 2.2a. The net is bounded and its reachability graph is finite.

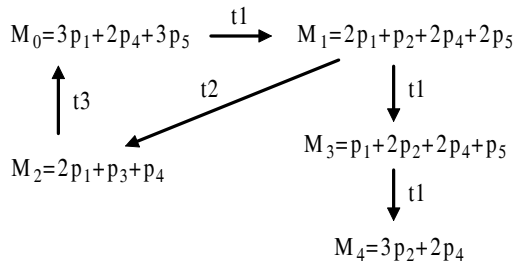


Fig. 2.3 The reachability graph of net (N, M_0) shown in Fig. 2.2a

Definition 2.6. A net $N = (P, T, F, W)$ is pure (self-loop free) iff $\forall x, y \in P \cup T$, $W(x, y) > 0$ implies $W(y, x) = 0$.

Definition 2.7. A pure net $N = (P, T, F, W)$ can be represented by its incidence matrix $[N]$, where $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. For a place p (transition t), its incidence vector, a row (column) in $[N]$, is denoted by $[N](p, \cdot)$ ($[N](\cdot, t)$).

According to the definition, it is easy to see the physical meanings of an element in an incidence matrix of a Petri net N . Specifically, $[N](p, t)$ indicates that p receives (loses) $|[N](p, t)|$ tokens if $[N](p, t) > 0$ ($[N](p, t) < 0$) after t fires. The number of tokens in p does not change if $[N](p, t) = 0$ after t fires. Vector $[N](p, \cdot)$ shows the token variation in p with respect to the firing of each transition once in the net N . Let $S \subseteq P$ be a subset of places in net N . $[N](S, \cdot)$ is used to denote $\sum_{p \in S} [N](p, \cdot)$.

Example 2.5. The incidence matrix of the net in Fig. 2.1a is shown below:

$$[N] = \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \\ -1 & -2 & 3 \end{pmatrix}.$$

$[N](p_1, t_1) = -1$ implies that p_1 loses a token after firing t_1 . $[N](p_1, t_3) = 1$ indicates that p_1 gets a token after t_3 fires. $[N](p_1, t_2) = 0$ means that the number of tokens in p_1 does not change after t_2 fires. Note that $[N](p_5, \cdot) = (-1, -2, 3)$. It implies that firing t_1 removes one token from p_5 , firing t_2 removes two tokens from p_5 , and firing t_3 deposits three tokens into p_5 .

Let $S = \{p_3, p_5\}$. $[N](S, \cdot) = [N](p_3, \cdot) + [N](p_5, \cdot) = (-1, -1, 2)$. It indicates that firing t_1 or t_2 removes one token from S , and firing t_3 puts two tokens into S .

It is important to note that the change of the number of tokens in a place p caused by firing some transition t does not depend on the current marking. Instead, it is completely determined by the structure of a net. In this sense, the incidence matrix suffices to characterize the relative change of tokens for every place when a transition fires.

The incidence matrix $[N]$ of a net N can be naturally divided into two parts $[N]^+$ and $[N]^-$ according to the token flow by defining $[N] = [N]^+ - [N]^-$, where $[N]^+(p, t) = W(t, p)$ and $[N]^-(p, t) = W(p, t)$ are called input (incidence) matrix and output (incidence) matrix, respectively. Note that the input and output matrices can completely describe a net structure, but it is not the case for incidence matrices in general. Two nets that have the same incidence matrices may have different net structures. This case likes an expression $a - b = c - d$ but $a = c$ and $b = d$ are not necessarily true. However, if there are no self-loops in a Petri net, its incidence matrix can completely determine its structure.

Example 2.6. For the net in Fig. 2.1a, its input matrix and output matrix are as follows:

$$[N]^+ = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 3 \end{pmatrix}, \quad [N]^- = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 0 \end{pmatrix}.$$

Accordingly, the enabling condition of a transition t can be rewritten as $M \geq [N]^-(\cdot, t)$.

Definition 2.8. Given a Petri net (N, M_0) , $t \in T$ is live under M_0 iff $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $M'[t]$. (N, M_0) is live iff $\forall t \in T$, t is live under M_0 . (N, M_0) is dead under M_0 iff $\nexists t \in T$, $M_0[t]$. (N, M_0) is deadlock-free (weakly live or live-locked) iff $\forall M \in R(N, M_0)$, $\exists t \in T$, $M[t]$.

Definition 2.9. Petri net (N, M_0) is quasi-live iff $\forall t \in T$, there exists $M \in R(N, M_0)$ such that $M[t]$ holds.

A live Petri net guarantees deadlock-freedom no matter what firing sequence is chosen but the converse is not true. However, this property is costly to verify.

Example 2.7. The net shown in Fig. 2.4a is deadlock-free since transitions t_1 and t_2 are live, while the net in Fig. 2.4b is live since all transitions are live. The net in Fig. 2.2e is dead since no transition is enabled under the current marking M_4 .

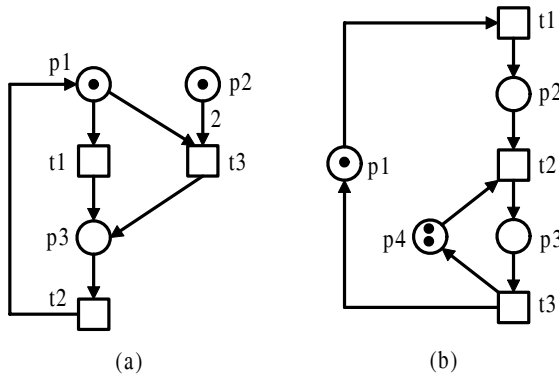


Fig. 2.4 Two Petri nets: (a) is deadlock-free and (b) is live

Definition 2.10. Let $N = (P, T, F, W)$ be a net and σ be a finite sequence of transitions. The Parikh vector of σ is $\vec{\sigma} : T \rightarrow \mathbb{N}$ which maps t in T to the number of occurrences of t in σ . Define $\vec{t}_1 = (1, 0, \dots, 0)^T$, $\vec{t}_2 = (0, 1, 0, \dots, 0)^T$, and $\vec{t}_k = (0, 0, \dots, 0, 1)^T$ assuming $k = |T|$.

Example 2.8. Let $\sigma_1 = t_1 t_3 t_2 t_4 t_5 t_2$ and $\sigma_2 = t_1$ be two sequences of transitions of some net N with $|T| = 6$. Their Parikh vectors are $\vec{\sigma}_1 = (1, 2, 1, 1, 1, 0)^T$ and $\vec{\sigma}_2 = (1, 0, 0, 0, 0, 0)^T$, respectively. Clearly, we have $\vec{\sigma}_2 = \vec{t}_1 = (1, 0, 0, 0, 0, 0)^T$. For the transition sequence $\sigma = t_1 t_1 t_1$ in the net shown in Fig. 2.1a, $\vec{\sigma} = (3, 0, 0)^T$.

It is trivial that for each transition t , we have $[N](\cdot, t) = [N]\vec{t}$. Note that $M[t]M'$ leads to $M' = M + [N](\cdot, t)$. Consequently, if $M[t]M'$, we have $M' = M + [N]\vec{t}$. For an arbitrary finite transition sequence σ such that $M[\sigma]M'$, we have

$$M' = M + [N]\vec{\sigma}. \quad (2.1)$$

Equation 2.1 is called the state equation of a Petri net (N, M) , which presents an algebraic description of the marking change in a Petri net. In other words, it is a compact way to express the interrelation between markings and numbers of transition occurrences in a transition sequence. Such a linear algebraic expression is

very helpful because it allows one to apply the concepts and results of linear algebra to the domain of Petri nets.

Any reachable marking fulfils the state equation but the converse is not true. In this sense, the state equation provides a necessary condition for a marking M to be reachable from an initial marking M_0 . That is to say, if marking M is reachable from M_0 , the state equation $M = M_0 + [N]\vec{\sigma}$ must have a vector solution for σ with its components in $\mathbb{N}$. Conversely, if the marking equation is not soluble, marking M is not reachable from M_0 .

Example 2.9. In Fig. 2.1a, $\sigma = t_1 t_1 t_1$ is a firable transition sequence with $\vec{\sigma} = (3, 0, 0)^T$. From Fig. 2.2d, we have $M_0[\sigma]M_4$, which can be verified by (2.1) as follows:

$$M_0 + [N]\vec{\sigma} = \begin{pmatrix} 3 \\ 0 \\ 0 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \\ -1 & -2 & 3 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 0 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} -3 \\ 3 \\ 0 \\ 0 \\ -3 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \\ 0 \\ 2 \\ 0 \end{pmatrix} = M_4.$$

Let $\sigma = t_1 t_2 t_3$. It is a firable transition sequence with $\vec{\sigma} = (1, 1, 1)^T$. From Fig. 2.2, we have $M_0[\sigma]M_0$ that can be verified by (2.1) as follows.

$$M_0 + [N]\vec{\sigma} = \begin{pmatrix} 3 \\ 0 \\ 0 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \\ -1 & -2 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 0 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 0 \\ 2 \\ 3 \end{pmatrix} = M_0.$$

Definition 2.11. Let (N, M_0) be a net system. Its linearized reachability set by using the state equation over the real numbers is defined as $R^S(N, M_0) = \{M | M = M_0 + [N]Y, M \geq 0, Y \geq 0\}$.

We have $R(N, M_0) \subseteq R^S(N, M_0)$ since the state equation does not check whether there is a sequence of intermediate markings such that some transition sequence σ is actually firable. The markings in $R^S(N, M_0) \setminus R(N, M_0)$ are called spurious markings (with respect to the state equation).

Although the reachability set derived from the state equation may contain spurious markings, in some cases its linear description facilitates the analysis of a Petri net.

For example, the verification of predicate $\min\{M(S) | M \in R(N, M_0)\} \geq k_1$ is difficult due to a potentially huge number of reachable markings in $R(N, M_0)$, where S is a subset of places and k_1 is a non-negative integer. However, the minimal number of tokens holding by S in $R^S(N, M_0)$ can be found by solving the following linear programming problem (LPP):

$$\begin{aligned} & \text{MIN } M(S) \\ & \text{s.t.} \\ & M = M_0 + [N]Y \end{aligned}$$

$$\begin{aligned} M &\geq 0 \\ Y &\geq 0 \end{aligned}$$

It is known that an LPP can be solved in polynomial time. Let k_2 be a feasible solution of the above LPP. Obviously, we have $k_2 \leq \min\{M(S) | M \in R(N, M_0)\}$. If $k_1 \leq k_2$, one gets $k_1 \leq k_2 \leq \min\{M(S) | M \in R(N, M_0)\}$, leading to the truth of this predicate. Certainly, if $k_1 > k_2$, one cannot give a definite answer to the truth of this predicate.

Example 2.10. $S = \{p_1, p_3, p_4\}$ is a set of places in the net shown in Fig 2.5, where $M_0 = p_3$. A question is whether S can be always marked. By solving an LPP, we have $\min\{M(S) | M = M_0 + [N]Y, M \geq 0, Y \geq 0\} = 1$. This leads to the fact that S can never be emptied, i.e., under any reachable marking, there is at least one place that is marked.

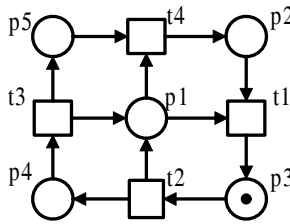


Fig. 2.5 A Petri net (N, M_0)

2.3 Structural Invariants

One important feature of Petri nets is that their structural properties can be obtained by linear algebraic techniques [13, 18, 42]. These properties that depend on only the topological structure of a Petri net and are independent of the initial marking are called invariants. Invariants are an important means for analyzing the behavior of a Petri net from a structural viewpoint.

Definition 2.12. A P -vector is a column vector $I : P \rightarrow \mathbb{Z}$ indexed by P and a T -vector is a column vector $J : T \rightarrow \mathbb{Z}$ indexed by T , where $\mathbb{Z}$ is the set of integers.

We denote column vectors where every entry equals 0(1) by $\mathbf{0}(\mathbf{1})$. I^T and $[N]^T$ are the transposed versions of vector I and matrix $[N]$, respectively. A $P(T)$ -vector is non-negative if no element in it is negative.

Definition 2.13. P -vector I is called a P -invariant (place invariant) iff $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$. T -vector J is called a T -invariant (transition invariant) iff $J \neq \mathbf{0}$ and $[N]J = \mathbf{0}$.

Definition 2.14. P -invariant I is a P -semiflow if every element of I is non-negative. $\|I\| = \{p | I(p) \neq 0\}$ is called the support of I . $\|I\|^+ = \{p | I(p) > 0\}$ denotes the positive support of P -invariant I and $\|I\|^- = \{p | I(p) < 0\}$ denotes the negative support of I . I is called a minimal P -invariant if $\|I\|$ is not a superset of the support of any other one and its components are mutually prime.

Definition 2.15. T -invariant J is a T -semiflow if every element of J is non-negative. $\|J\| = \{t | J(t) \neq 0\}$ is called the support of J . $\|J\|^+ = \{t | J(t) > 0\}$ denotes the positive support of T -invariant J and $\|J\|^- = \{t | J(t) < 0\}$ denotes the negative support of J . J is called a minimal T -invariant if $\|J\|$ is not a superset of the support of any other one and its components are mutually prime.

Note that a set of numbers is mutually prime if their common divisor is one. For example, 4, 7, and 16 are mutually prime. But 4, 6, and 16 are not since 2 is their common divisor. A P -invariant corresponds to a set of places whose weighted token count is a constant for any reachable marking. It follows immediately from the state equation.

Theorem 2.1. Let (N, M_0) be a net with P -invariant I and M be a reachable marking from M_0 . Then

$$I^T M = I^T M_0.$$

A fundamental property of a T -invariant follows immediately from the state equation.

Theorem 2.2. Let (N, M_0) be a net with a transition sequence σ such that $M_0[\sigma]M$. $M = M_0$ iff $\vec{\sigma}$ is a T -invariant of N .

Note that for a specific marked net, the existence of a T -invariant does not imply that there exists a transition sequence whose Parikh vector is the T -vector such that it is firable and its firing leads the net from the initial marking back to it. Furthermore, it is easy to see that any linear combination of $P(T)$ -invariants of a net is still a $P(T)$ -invariant of the net.

Property 2.1. If I is a P -semiflow of a net, $\bullet\|I\| = \|I\|\bullet$.

Example 2.11. In the net shown in Fig. 2.1a, there are three minimal P -invariants: $I_1 = p_1 + p_2 + p_3$, $I_2 = p_3 + p_4$, and $I_3 = p_2 + 3p_3 + p_5$, since $\forall i \in \{1, 2, 3\}$, $I_i^T[N] = \mathbf{0}^T$. $\forall M \in R(N, M_0)$, $I_1^T M = I_1^T M_0 = M_0(p_1) + M_0(p_2) + M_0(p_3) = 3$. This indicates that the token count in places p_1 , p_2 , and p_3 keeps three under any reachable marking, which can be verified from the reachability graph, which is identical with the one shown in Fig 2.3.

The net has a unique T -invariant $J = t_1 + t_2 + t_3$ and the transition sequence $\sigma = t_1 t_2 t_3$ is firable. As a result, $M_0[t_1]M_1[t_2]M_2[t_3]M_0$.

Since I_1 and I_2 are P -invariants, $I = I_1 - I_2 = p_1 + p_2 - p_4$ is a P -invariant as well. Note that I is not a P -semiflow due to its negative component. Moreover, one can get $\|I_1\| = \{p_1, p_2, p_3\}$, $\|I\|^+ = \{p_1, p_2\}$, and $\|I\|^- = \{p_4\}$. It is easy to see that $\bullet\|I_1\| = \bullet p_1 \cup \bullet p_2 \cup \bullet p_3 = \{t_3\} \cup \{t_1\} \cup \{t_2\} = \{t_1, t_2, t_3\}$ and $\|I_1\|\bullet = p_1\bullet \cup p_2\bullet \cup p_3\bullet = \{t_1\} \cup \{t_2\} \cup \{t_3\} = \{t_1, t_2, t_3\}$. $\|I_1\|\bullet = \bullet\|I_1\|$ will not be surprising since I_1 is a P -semiflow.

A Petri net is strongly connected if $\forall x, y \in P \cup T$, there is a sequence of nodes $x, a, b, \dots, c, y$ such that $(x, a), (a, b), \dots, (c, y) \in F$, where $\{a, b, \dots, c\} \subseteq P \cup T$. A string $x_1 \dots x_n$ is called a path of N iff $\forall i \in \mathbb{N}_{n-1}, x_{i+1} \in x_i^\bullet$, where $\forall x \in \{x_1, \dots, x_n\}, x \in P \cup T$. An elementary path from x_1 to x_n is a path whose nodes are all different (except, perhaps, x_1 and x_n). A path $x_1 \dots x_n$ is called a circuit iff it is an elementary path and $x_1 = x_n$.

The liveness of a Petri net is close to its connectedness. A result is given in [17]: Each connected net with a live and bounded marking is strongly connected. A result that establishes a bridge between strong connectedness and invariants is given as follows owing to [18]:

Theorem 2.3. *Each connected net with a positive place invariant and positive transition invariant is strongly connected.*

2.4 Siphons and Traps

P -invariants that can be derived from the state equation of a Petri net are marking invariants. The token count in their corresponding places stays constant, i.e., the invariant law associated with a P -invariant holds for any reachable marking. In a Petri net, siphons and traps are also structural objects that involve marking invariants. However, the invariant laws associated with them do not hold under any reachable marking, but once they become true they remain true for any subsequently reachable markings. A siphon remains empty once it loses all tokens. A trap remains marked once it has any token in it. Siphons and traps have been extensively investigated and used for the structural analysis of a Petri net. They also play an important role in the liveness analysis of a net, particularly in ordinary ones.

Definition 2.16. A non-empty set $S \subseteq P$ is a siphon iff ${}^\bullet S \subseteq S^\bullet$. $S \subseteq P$ is a trap iff $S^\bullet \subseteq {}^\bullet S$. A siphon (trap) is minimal iff there is no siphon (trap) contained in it as a proper subset. A minimal siphon S is said to be strict if ${}^\bullet S \subsetneq S^\bullet$.

Property 2.2. Let S_1 and S_2 are two siphons (traps). Then, $S_1 \cup S_2$ is a siphon (trap).

Example 2.12. In the net shown in Fig. 2.1a, $S_1 = \{p_1, p_2, p_3\}$, $S_2 = \{p_4, p_3\}$, $S_3 = \{p_2, p_3, p_5\}$, and $S_4 = \{p_3, p_5\}$ are siphons, among which S_1, S_2 , and S_4 are minimal since the removal of any place from each of these sets leads to the fact that the resultant set is not a siphon any more. Note that ${}^\bullet S_1 = S_1^\bullet$, ${}^\bullet S_2 = S_2^\bullet$, and ${}^\bullet S_3 = S_3^\bullet$. S_1, S_2 , and S_3 are also traps. By ${}^\bullet S_4 = \{t_2, t_3\}$ and $S_4^\bullet = \{t_1, t_2, t_3\}$, we have ${}^\bullet S_4 \subset S_4^\bullet$. $S_4 = \{p_3, p_5\}$ is therefore a strict minimal siphon.

Corollary 2.1. *If I is a P -semiflow, then $\|I\|$ is both a siphon and trap.*

Note that the converse of Corollary 2.1 is not true since a P -invariant depends on not only the topological structure of a net but also the weights attached to the arcs. However, a siphon or trap depends on the topological structure only. For example,

$S = \{p_1, p_2\}$ in Fig. 2.6 is both a siphon and trap. However, it is not the support of a P -semiflow. In this sense, the converse of Corollary 2.1 is true in the domain of ordinary nets.

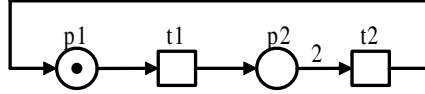


Fig. 2.6 A siphon and trap in a net without P -semiflow

If a siphon contains the support of a P -semiflow and the support is initially marked, then it can never be emptied. In addition, traps and siphons have the following marking invariant laws.

Property 2.3. Let $M \in R(N, M_0)$ be a marking of net (N, M_0) and S a trap. If $M(S) > 0$, then $\forall M' \in R(N, M)$, $M'(S) > 0$.

This property implies that once a trap is marked under a marking, it is always marked under the subsequent markings that are reachable from the current one.

Property 2.4. Let $M \in R(N, M_0)$ be a marking of net (N, M_0) and S a siphon. If $M(S) = 0$, then $\forall M' \in R(N, M)$, $M'(S) = 0$.

Property 2.4 indicates that once a siphon loses all its tokens, it remains unmarked under any subsequent markings that are reachable from the current marking. An empty siphon S causes that no transition in $S^\bullet$ is enabled. Due to the definition of siphons, all transitions connected to S can never be enabled once it is emptied. The transitions are therefore dead, leading to the fact that the net containing these transitions is not live.

As a result, deadlock-freeness and liveness of a Petri net are closely related to its siphons, which is shown by the following known results [16].

Theorem 2.4. Let (N, M_0) be an ordinary net and Π the set of its siphons. The net is deadlock-free if $\forall S \in \Pi$, $\forall M \in R(N, M_0)$, $M(S) > 0$.

This theorem states that an ordinary Petri net is deadlock-free if no (minimal) siphon eventually becomes empty.

Theorem 2.5. Let (N, M) be an ordinary net that is in a deadlock state. Then, $\{p \in P \mid M(p) = 0\}$ is a siphon.

This result means that if an ordinary net is dead, i.e., no transition is enabled, then the unmarked places form a siphon.

Example 2.13. The net shown in Fig. 2.7a is a famous example as first discussed by Zhou et al. [61, 62] and later by Chu and Xie [12] and many other researchers [6]. It has four minimal siphons $S_1 = \{p_1, p_2, p_3, p_4\}$, $S_2 = \{p_3, p_5\}$, $S_3 = \{p_2, p_4, p_6\}$,

and $S_4 = \{p_4, p_5, p_6\}$. S_1, S_2 and S_3 are also traps that are initially marked. Note that S_4 is a strict minimal siphon since $\bullet S_4 = \{t_2, t_3, t_4\}$ and $S_4^\bullet = \{t_1, t_2, t_3, t_4\}$, leading to the truth of $\bullet S_4 \subsetneq S_4^\bullet$.

In Fig. 2.7a, $\sigma = t_1 t_2 t_1$ is a firable transition sequence whose firing leads to a new marking as shown in Fig. 2.7b. The net in Fig. 2.7b is dead since no transition is enabled in the current marking. The unmarked places p_1, p_4, p_5 , and p_6 form a siphon $S = \{p_1, p_4, p_5, p_6\}$ that is not minimal since it contains S_4 . The emptiness of S disables every transition in $S^\bullet$ such that no transition in this net is enabled. As a result, the net is dead.

Based on Theorem 2.5, we can achieve the following results.

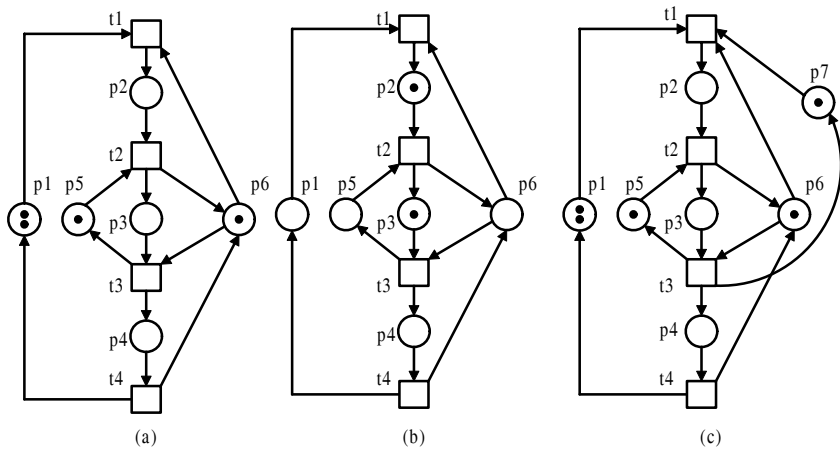


Fig. 2.7 (a) A Petri net [61], (b) a dead marking, and (c) a controlled siphon

Corollary 2.2. A deadlocked ordinary Petri net contains at least one empty siphon.

Corollary 2.3. Let $N = (P, T, F, W)$ be a deadlocked net under marking M . Then, it has at least one siphon S such that $\forall p \in S, \exists t \in p^\bullet$ such that $W(p, t) > M(p)$.

Definition 2.17. A siphon S is said to be controlled in a net system (N, M_0) iff $\forall M \in R(N, M_0), M(S) > 0$.

Clearly, any siphon that contains a marked trap is controlled since it can never be emptied. In an ordinary Petri net, a siphon that is controlled does not imply a deadlock. This is not the case in a generalized Petri net. For example, there are two minimal siphons in the generalized Petri net shown in Fig 2.4a. They are $S_1 = \{p_1, p_3\}$ and $S_2 = \{p_2\}$. Both of them can never be unmarked. However, the insufficient number of tokens in S_2 disables t_3 . In fact, t_3 is a dead transition in the net. Hence, the net is not live even though each siphon is always marked. Chapter 3 shows that a siphon in a generalized Petri net does not lead to dead transitions if it is max-controlled [4].

For a siphon that can be emptied in a net, some external control mechanism can be exerted on the net such that it becomes controlled. In Fig. 2.7a, S_4 is a strict minimal siphon whose emptiness leads to the deadlock of the net. To prevent S_4 from being unmarked, a place p_7 is added with $\bullet p_7 = \{t_3\}$ and $p_7^\bullet = \{t_1\}$, as shown in Fig. 2.7c. The initial marking of p_7 is one. Such an additional place is called a *monitor* or control place in terms of its role. In Fig. 2.7c, the addition of p_7 leads to an extra minimal siphon $S_5 = \{p_2, p_3, p_7\}$ that is a marked trap. As a result, no siphon can be emptied in the net and it is deadlock-free (actually, live). This example motivates one to explore the mechanism to make a siphon controlled by adding a monitor.

When we talk about siphon control, we are usually concerned with minimal siphons since the controllability of a minimal siphon implies that of those containing it.

A natural problem is to decide whether a set of places S in a Petri net is a minimal siphon. It is shown in [2] that the decision can be done in polynomial time with complexity $O(m^2 + mn^2)$, where $m = |S^\bullet|$ and $n = |S|$.

Definition 2.18. Let $N = (P, T, F, W)$ be a Petri net with $P_X \subseteq P$ and $T_X \subseteq T$. $N_X = (P_X, T_X, F_X, W_X)$ is called a subnet generated by $P_X \cup T_X$ if $F_X = F \cap [(P_X \times T_X) \cup (T_X \times P_X)]$ and $\forall f \in F_X, W_X(f) = W(f)$.

Property 2.5. Let S be a minimal siphon in a net N . The subnet generated by $S \cup \bullet S$ is strongly connected.

The following definition is from [12, 32].

Definition 2.19. Siphon S in an ordinary net system (N, M_0) is invariant-controlled by P -invariant I under M_0 iff $I^T M_0 > 0$ and $\forall p \in P \setminus S, I(p) \leq 0$, or equivalently, $I^T M_0 > 0$ and $\|I\|^+ \subseteq S$.

If S is controlled by P -invariant I under M_0 , S cannot be emptied, i.e., $\forall M \in R(N, M_0)$, S is marked under M .

Example 2.14. In Fig. 2.7c, one can verify that $I_1 = p_3 + p_5$, $I_2 = p_2 + p_4 + p_6$, and $I_3 = p_2 + p_3 + p_7$ are P -invariants. As a result, $I = I_1 + I_2 - I_3 = p_4 + p_5 + p_6 - p_7$ is a P -invariant as well. It is easy to see that siphon $S_4 = \{p_4, p_5, p_6\}$ is controlled by P -invariant I since $\|I\|^+ = \{p_4, p_5, p_6\} = S_4$ and $I^T M_0 = M_0(p_4) + M_0(p_5) + M_0(p_6) - M_0(p_7) = 2 - 1 = 1 > 0$. The controllability of $S_4 = \{p_4, p_5, p_6\}$ implies that of siphon $S = \{p_1, p_4, p_5, p_6\}$ that is not minimal. Note that $\bullet S_4 = \{t_2, t_3, t_4\}$. The subset generated by $S_4 \cup \bullet S_4$ is shown in Fig. 2.8. It is clearly strongly connected since $S_4 = \{p_4, p_5, p_6\}$ is a minimal siphon.

In essence, the controllability of siphon S by adding a monitor is ensured by the fact that the number of tokens leaving S is limited by a marking invariant law imposed on the Petri net, which is implemented by a P -invariant whose support contains the monitor.

In order to test whether a siphon S is controlled by a P -invariant I , it is sufficient to solve the following system of linear homogeneous inequalities and equations:

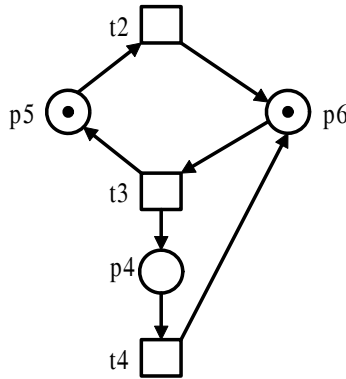


Fig. 2.8 A subnet generated by a minimal siphon and its preset

$$\begin{aligned} I^T[N] &= \mathbf{0}^T \\ I^T M_0 &> 0 \\ I(p) &\leq 0, \forall p \in P \setminus S \end{aligned}$$

For the above system, the existence of a solution can be proved through *Phase I* of the simplex algorithm applied to the following LPP:

$$\begin{aligned} &\text{maximize } \mathbf{0}^T I \\ &\text{s. t.} \\ &I^T[N] = \mathbf{0}^T \\ &I^T M_0 > 0 \\ &I(p) \leq 0, \forall p \in P \setminus S \end{aligned}$$

Phase I of the simplex algorithm computes a *basic feasible solution* of the set of constraints of the LPP if it exists.

An empty or insufficiently marked siphon in a Petri net can cause some transitions not to be enabled. A siphon in an ordinary Petri net can be made invariant-controlled as defined above. The case in a generalized Petri net is much more complicated and is treated as follows.

Definition 2.20. Let (N, M_0) be a net system and S be a siphon of N . S is said to be max-marked at a marking $M \in R(N, M_0)$ iff $\exists p \in S$ such that $M(p) \geq \max_p \bullet$.

Definition 2.21. A siphon is said to be max-controlled iff it is max-marked at any reachable marking.

Definition 2.22. (N, M_0) satisfies the maximal cs-property (maximal controlled-siphon property) iff each minimal siphon of N is max-controlled.

The following results are owing to [4]. In case of no confusion, maximal cs-property is called cs-property for the sake of simplification.

Property 2.6. If (N, M_0) satisfies the cs-property, it is deadlock-free.

Property 2.7. If (N, M_0) is live, it satisfies the cs-property.

A siphon satisfying the max-controlled property can be always marked sufficiently to allow firing a transition once at least. In order to check and use the cs-property, Barkaoui et al. [4] propose the conditions to determine whether a given siphon is max-controlled.

Proposition 2.1. Let (N, M_0) be a Petri net and S be a siphon of N . If there exists a P -invariant I such that $\forall p \in (||I||^- \cap S), \max_{p^\bullet} = 1, ||I||^+ \subseteq S$ and $I^T M_0 > \sum_{p \in S} I(p)(\max_{p^\bullet} - 1)$, then S is max-controlled.

Example 2.15. Figure 2.9a shows a generalized net and $I_1 = p_2 + p_6$ and $I_2 = p_2 + 3p_3 + p_5$ are its two P -invariants. Trivially, $I = I_2 - I_1 = 3p_3 + p_5 - p_6$ is also a P -invariant. Let $S = \{p_3, p_5\}$ be a set of places. Since $\bullet S \subseteq S^\bullet$, S is a strict minimal siphon. Next we show that it is max-controlled by P -invariant I . It is clear that $||I||^- \cap S = \emptyset$ and $||I||^+ = S$. We then check the truth of $I^T M_0 > \sum_{p \in S} I(p)(\max_{p^\bullet} - 1)$. $I^T M_0 = M_0(p_5) + 3M_0(p_3) - M_0(p_6) = 3 - 1 = 2$. $\sum_{p \in S} I(p)(\max_{p^\bullet} - 1) = I(p_3)(\max_{p_3^\bullet} - 1) + I(p_5)(\max_{p_5^\bullet} - 1)$. Considering $\max_{p_3^\bullet} = 1$ and $\max_{p_5^\bullet} = 2$, we have $\sum_{p \in S} I(p)(\max_{p^\bullet} - 1) = 1$. Therefore, $I^T M_0 > \sum_{p \in S} I(p)(\max_{p^\bullet} - 1)$ and S is max-controlled. Figure 2.9b shows the reachability graph of the net in Fig. 2.9a.

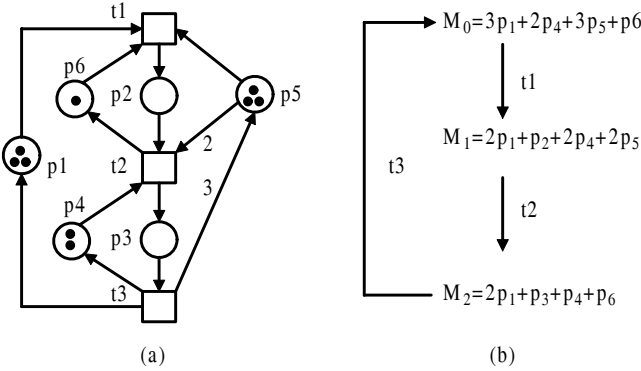


Fig. 2.9 A max-controlled siphon in a net (N, M_0) (a) Petri net model and (b) reachability graph

By comparing the net in Fig. 2.9a with Fig. 2.1a as well as their reachability graphs as shown in Fig. 2.9b and Fig. 2.3, respectively, one concludes that the addition of p_6 removes two markings M_3 and M_4 in Fig. 2.3: one is a deadlock marking and the other is a marking that inevitably leads the system to deadlock.

Remark 2.1. The number of siphons (minimal siphons) grows fast with respect to the size of a Petri net and in the worst case grows exponentially with a net size. However, many deadlock control approaches depend on the complete or partial enumeration of siphons in a plant net model [23, 33, 34, 40, 50–52, 58, 59]. It is well known that the complete siphon enumeration is time-consuming. Extensive studies have been

conducted on the siphon computation, leading to a variety of methods [1, 14, 22, 31, 35, 53, 54]. A recent work [15] by Cordone et al. claims that their proposed siphon computation method can find more than 2×10^7 siphons in less than one hour.

2.5 Subclasses of Petri Nets

There are a number of interesting subclasses of ordinary Petri nets. The reasons that they are interesting are twofold. First, they play an important role in the development of certain application of Petri nets [11, 12]. Second, some relevant analysis problems in these classes can be solved in polynomial time [3, 21, 36]. For example, the problem of deciding whether a free-choice Petri net is live and bounded can be solved in $O(nm)$ [21], where n and m are the number of places and transitions of the net, respectively. In turn, many analysis problems of live and bounded free-choice nets are also shown to have polynomial time complexity [16].

Definition 2.23. A Petri net $N = (P, T, F)$ is called a state machine iff $\forall t \in T, |\bullet t| = |t \bullet| = 1$.

In a state machine, each transition has exactly one input place and exactly one output place. Each transition allows tokens to flow from one place to another. Multiple transitions may allow tokens to flow from their respective places to the same place. In addition, a single token in a place p enables all transitions in $p \bullet$. Firing any of them disables the others. This is called a conflict. Note that all finite automata can be described as the state machines of Petri nets.

Theorem 2.6. A state machine (N, M_0) is live iff N is strongly connected and M_0 marks at least one place.

Definition 2.24. A Petri net $N = (P, T, F)$ is said to be a marked graph iff $\forall p \in P, |\bullet p| = |p \bullet| = 1$.

In a marked graph, each place has exactly one input transition and exactly one output transition. A transition may have multiple input places and output places. In this sense, a marked graph allows concurrent and synchronization structure. A state machine admits no synchronization and a marked graph allows no conflict.

Theorem 2.7. A marked graph (N, M_0) is live iff M_0 places at least one token on each circuit in N .

Definition 2.25. A Petri net is a free-choice net iff $\forall p_1, p_2 \in P, p_1 \bullet \cap p_2 \bullet \neq \emptyset \Rightarrow |p_1 \bullet| = |p_2 \bullet| = 1$.

In a free-choice net, every arc from a place is either a unique outgoing arc or a unique incoming arc to a transition. A free-choice net allows both conflict and synchronization, i.e., state machines and marked graphs fall under the class of free-choice nets.

Theorem 2.8. A free-choice net (N, M_0) is live iff every siphon in it contains a marked trap.

Definition 2.26. A Petri net is an extended free-choice net iff $\forall p_1, p_2 \in P, p_1^\bullet \cap p_2^\bullet \neq \emptyset \Rightarrow p_1^\bullet = p_2^\bullet$.

Definition 2.27. A Petri net is an asymmetric choice net iff $\forall p_1, p_2 \in P, p_1^\bullet \cap p_2^\bullet \neq \emptyset \Rightarrow p_1^\bullet \subseteq p_2^\bullet$ or $p_2^\bullet \subseteq p_1^\bullet$.

Theorem 2.9. An asymmetric choice net (N, M_0) is live if (but not only if) every siphon in N contains a marked trap.

Example 2.16. Figure 2.10 shows some subclasses of Petri nets.

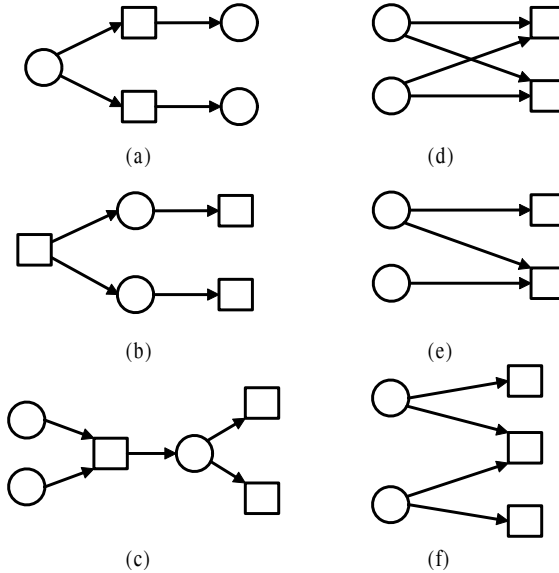


Fig. 2.10 Subclasses of Petri nets: (a) a state machine but not marked graph, (b) a marked graph but not state machine, (c) a free-choice net, (d) an extended free-choice net, (e) an asymmetric net, and (f) a Petri net

2.6 Petri Nets and Automata

Since the reachability graph of a Petri net is an automaton, this section presents some basics of finite-state automata [26], which are helpful to understand what is presented in this book.

Definition 2.28. A (deterministic) finite-state automaton is a 5-tuple $G = (Q, \Sigma, \delta, q_0, Q_m)$, where Q is a finite set of states, Σ is a finite alphabet of symbols that we refer to as event labels, $\delta: Q \times \Sigma$ the (partial) transition function, q_0 the initial state, and $Q_m \subseteq Q$ the set of marker states.

δ is a partial function since $\delta(q, \alpha)$ may not be defined for all $(q, \alpha) \in Q \times \Sigma$. When $\delta(q, \alpha)$ is defined, it implies that $\exists q' \in Q$ and $\alpha \in \Sigma$, the occurrence of event α transits the automaton from states q to q' .

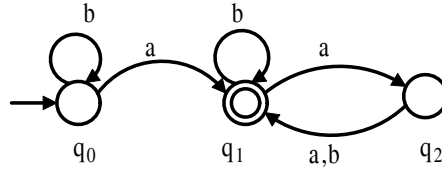


Fig. 2.11 An automaton

The operation of a finite-state automaton is always illustrated in a state diagram. Graphically, the initial state is marked with an input arrow and the marker states are denoted by double circles. For instance, Figure 2.11 shows an automaton G , where $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, the initial state is q_0 , $Q_m = \{q_1\}$, $\delta(q_0, b) = q_0$, $\delta(q_0, a) = q_1$, $\delta(q_1, b) = q_1$, $\delta(q_1, a) = q_2$, $\delta(q_2, a) = q_1$, and $\delta(q_2, b) = q_1$.

The behavior of a system modeled by an automaton can be characterized by the language that the automaton speaks, i.e., a set of sequences of symbols of events from Σ , which are physically possible. For example, $\sigma = abab$ is a possible sequence of events in the automaton in Fig. 2.11. The set of all finite sequences over Σ is denoted by Σ^* , which includes the empty string whose length is zero and which is denoted by ϵ .

Definition 2.29. A labeled Petri net is a net with a labeling function $l: T \rightarrow 2^\Sigma \cup \{\epsilon\}$, where Σ is the set of events and ϵ is a null event. A net is said to be free-labeled if each transition $t \in T$ is labeled by a single event $a \in \Sigma$ and different transitions bear different labels.

The reachability graph of a free-labeled Petri net corresponds to a deterministic automaton. A finite automaton can easily be converted into a labeled Petri net by inserting a transition that is labeled by the symbol between two connected states. The states in the automaton are differently numbered by places. Figure 2.12 is the equivalent labeled Petri net of the automaton depicted in Fig. 2.11.

For supervisory control of DES in a Petri net formalism, we are more concerned with a free-labeled Petri net representation. Unfortunately, it is shown that not all finite automata admit a free-labeled Petri net representation. It remains unanswered what finite automata do have a free-labeled Petri net realization. Figure 2.13 shows two finite automata that have no such realizations.

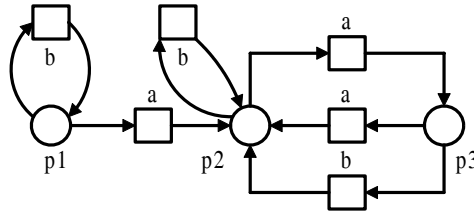


Fig. 2.12 The equivalent labeled Petri net of a finite automaton (a) Petri net A and (b) Petri net B

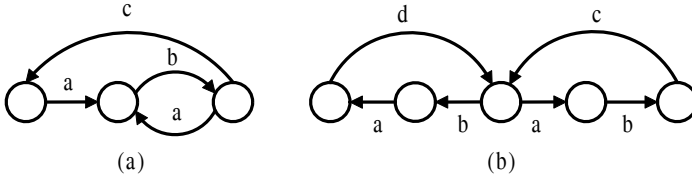


Fig. 2.13 Two finite automata without Petri net realizations

2.7 Plants, Supervisors, and Controlled Systems

In traditional supervisory control theory of DES, a system to be controlled is called a plant or a plant net model if Petri nets are used as a formalism. The external agent that forces the system to behave to satisfy given control specifications and requirements is usually called a supervisor. In a Petri net formalism, a supervisor is a Petri net that usually consists of a set of *monitors*, sometimes called control places, and a set of transitions of the plant net model. There are no places of the plant model in its supervisor. The role of the monitors in a supervisor is to supervise the plant such that its behavior satisfies the control specifications. The compound of a plant net model and its Petri net supervisor is called the controlled (net) system of the plant, whose behavior does not violate the given control specifications and requirements. To formally define a controlled system, it is necessary to first define a class of compositions of two Petri nets via shared transitions. This composition is also called synchronous synthesis of Petri nets.

Definition 2.30. Let (N_1, M_1) and (N_2, M_2) be two nets with $N_i = (P_i, T_i, F_i, W_i)$, $i = 1, 2$, satisfying $P_1 \cap P_2 = \emptyset$. (N, M) with $N = (P, T, F, W)$ is said to be a synchronous synthesis net resulting from the merge of (N_1, M_1) and (N_2, M_2) , denoted by $(N_1, M_1) \otimes (N_2, M_2)$, iff

1. $P = P_1 \cup P_2$
2. $T = T_1 \cup T_2$
3. $F = F_1 \cup F_2$
4. $W(f) = W_i(f)$ if $f \in F_i$, $i = 1, 2$
5. $M(p) = M_i(p)$ if $p \in P_i$, $i = 1, 2$.

Definition 2.31. Let (N_1, M_1) , (N_2, M_2) , $\dots$, and (N_k, M_k) be k nets satisfying $P_i \cap P_j = \emptyset$, $\forall i, j \in \mathbb{N}_k$, $i \neq j$. The synchronous synthesis of the k Petri nets (N_1, M_1) , (N_2, M_2) , $\dots$, and (N_k, M_k) is defined as $(N, M) = (N_k, M_k) \otimes (\bigotimes_{i=1}^{k-1} (N_i, M_i))$.

In a Petri net formalism, a supervisor is a Petri net that usually consists of a set of monitors and a set of transitions, which is a subset of the set of transitions in the plant net model. The controlled system is the synchronous synthesis of a plant net model and its supervisor via shared transitions.

Definition 2.32. Let (N_p, M_p) with $N_p = (P, T, F, W)$ be a plant model and (N_{sup}, M_{sup}) with $N_{sup} = (P_V, T_V, F_V, W_V)$ its supervisor, where $P \cap P_V = \emptyset$ and $T_V \subseteq T$. The controlled system of the plant model is $(N_p, M_p) \otimes (N_{sup}, M_{sup})$.

Example 2.17. The Petri net shown in Fig. 2.14a is a plant model. The control specification is that the number of tokens in place p_2 is not greater than one at any reachable marking. The net depicted in Fig. 2.14b is a supervisor that can implement this control specification, where p_3 is a monitor, $P_V = \{p_3\}$, and $T_V = \{t_1, t_2\}$. Figure 2.14c shows the controlled system that can be obtained by synchronous synthesis of the nets in Fig. 2.14a, b. It is easy to verify that the number of tokens in p_2 can never be greater than one.

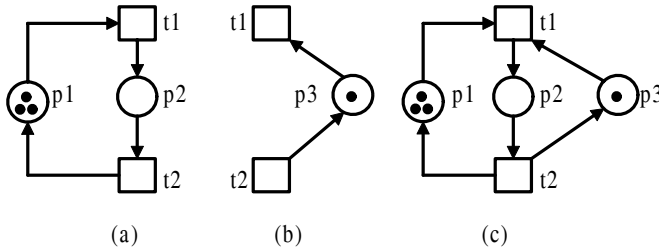


Fig. 2.14 (a) a plant net model, (b) the supervisor, and (c) the controlled system

2.8 Bibliographical Remarks

All the material covered in this chapter can be found in standard books [16, 39, 43] and survey papers [37, 38]. A good paper on siphons is [2], which presents an effective characterization of minimal siphons and traps from the viewpoint of graph theory. The algorithms calculating siphons and traps can be found in [5, 9, 10, 14, 15, 22, 29, 31, 48, 54, 57, 60]. For a general introduction to the subclasses of Petri nets, we refer readers to [37]. Good surveys of Petri nets from a system theory view can be found in [24, 46].

For a more extensive discussion of the original framework of DES supervisory control based on formal languages and automata, we refer readers to the tutorial surveys, papers and books [7, 8, 25, 27, 30, 41, 49].

Problems

2.1. It is known that the siphons are closely related to the deadlock or the existence of dead transitions in a Petri net. Suppose that (N, M_0) is a net without siphons. Is it live? Results can be found in [55].

2.2. INA [47] is a widely used tool that supports the behavioral and structural analysis of Petri nets. Let us define the size of a net (N, M_0) as $||\mathcal{N}|| = |P| + |T| + \sum_{p \in P} M_0(p)$. By using INA, compute the reachability graphs for a number of Petri nets with different sizes 5, 10, 20, ..., and 100, and observe the relationship between the CPU-time and the size of a Petri net.

2.3. Figure 2.15 shows the reduced version of the reachability graph of the net in Fig. 2.1a. It is clear that M_4 is a deadlock marking and M_3 is a marking that definitely leads the system to a deadlock state. These are “bad” states, which the system is not allowed to enter. M_1 is called a *dangerous marking* since, at this marking, the system may enter M_3 if supervisory control is not properly imposed.

Therefore, M_0 , M_1 , and M_2 form the good behavior of the system. An intuitive idea is to design an online supervisor that supervises the system such that if the system reaches M_1 , it disables t_1 and directs the system to M_2 .

Combining with the results for Problem 2.2, discuss the disadvantages of this intuitive control idea. Try to implement this idea by some programming language and check the size of the problem that can be processed by your computer.

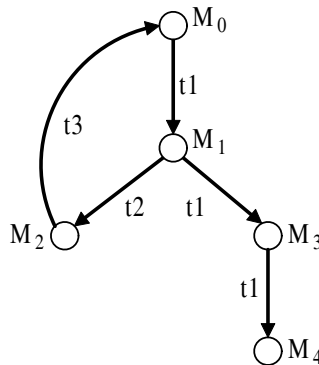


Fig. 2.15 A reachability graph

2.4. Prove Corollary 2.3.

2.5. Find and compare the strict minimal siphons in Fig 2.16a, b. Change the initial markings of places p_1 and p_4 and verify the liveness of the two nets by INA. This verification may find an interesting problem about deadlocks and siphons in a generalized Petri net.

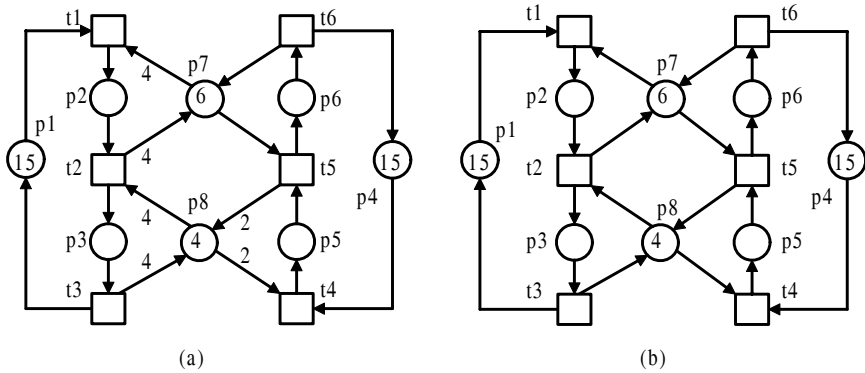


Fig. 2.16 Two Petri nets (a) a generalized one and (b) an ordinary one

2.6. The reachability graph of a Petri net (N, M_0) can be constructed using the following algorithm that terminates in a finite number of steps if its reachability set is finite. Starting with M_0 , all the enabled transitions can be fired. These firings can lead to new markings that may enable other transitions. Taking each of the new markings as a new root, we can recursively generate all the reachable markings. The following reachability graph generation algorithm can be found in [37, 39].

Algorithm 2.1 Reachability graph

- 1: The root node is M_0 . This node has initially no label.
- 2: **while** There are nodes with no label **do**
- 3: Consider a node M with no label.
- 4: (a) For each transition t enabled at M :
- 5: Let $M' = M + [N](\cdot, t)$.
- 6: **if** There does not exist a node M' in the graph **then**
- 7: Add it.
- 8: Add an arc t from M to M' .
- 9: **end if**
- 10: (b) Label the node M "old".
- 11: **end while**
- 12: Remove all labels from nodes.

Implement the algorithm in a programming language and find the reachability graphs for all the Petri nets in Chap. 2 and those used in Problem 2.2. Check the

maximal size of a reachability graph that your computer can process in reasonable time. Compare the CPU-times needed by your program and INA.

Let $M, M' \in R(N, M_0)$ be two reachable markings of a Petri net (N, M_0) with $N = (P, T, F, W)$. M is said to cover M' if $M \geq M'$, i.e., $\forall p \in P, M(p) \geq M'(p)$.

For an unbounded Petri net, its reachability graph can grow indefinitely. To reduce and keep the size of the graph finite, a special symbol $\bar{\omega}$ is usually introduced, which represents a number of tokens that can be made arbitrarily large. For any finite integer a , $\bar{\omega}$ is subject to the following four rules:

$$\begin{aligned} a &\leq \bar{\omega}, \\ \bar{\omega} &\leq \bar{\omega}, \\ \bar{\omega} + a &= \bar{\omega}, \\ \bar{\omega} - a &= \bar{\omega}. \end{aligned}$$

By using the above notations, a special graph called a coverability graph can be constructed using the algorithm stated in [19]. If there is no symbol $\bar{\omega}$ in a graph, it is also a reachability graph. A coverability graph is finite and contains every reachable marking from an initial marking M_0 , which is either explicitly represented by a node, or is covered by a node through the use of $\bar{\omega}$. For details, the reader can be referred to [19] and [37].

Additional work related to the check of liveness of unbounded Petri nets can be found in [20, 28, 45, 56]. The liveness analysis problem of generalized unbounded Petri nets remains open.

References

1. Abdallah, I.B., ElMaraghy, H.A., ElMekkawy, T. (1997) A logic programming approach for finding minimal siphons in S^3PR nets applied to manufacturing systems. In *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp.1710–1715.
2. Barkaoui, K., Lemaire, B. (1989) An effective characterization of minimal deadlocks and traps in Petri nets based on graph theory. In *Proc. 10th Int. Conf. on Applications and Theory of Petri Nets*, pp.1–21.
3. Barkaoui, K., Minoux, M. (1992) A polynomial time graph algorithm to decide liveness of some basic classes of Petri nets. In *Proc. 13th Int. Conf. on Applications and Theory of Petri Nets, Lecture Notes in Computer Science*, vol.616, pp.62–75.
4. Barkaoui, K., Pradat-Peyre, J.F. (1996) On liveness and controlled siphons in Petri nets. In *Proc. 17th Int. Conf. on Applications and Theory of Petri Nets Lecture Notes in Computer Science*, vol.1091, pp.57–72.
5. Boer, E.R., Murata, T. (1994) Generating basis siphons and traps of Petri nets using the sign incidence matrix. *IEEE Transactions on Circuits and Systems I—Fundamental Theory and Applications*, vol.41, no.4, pp.266–271.
6. Bogdan, S., Lewis, F.L., Kovacic, Z., Mireles, J. (2006) *Manufacturing Systems Control Design*. London: Springer.
7. Cassandras, C.G., Lafortune, S. (1999) *Introduction to Discrete Event Systems*. Boston, MA: Kluwer.
8. Cassandras, C.G., Lafortune, S. (2008) *Introduction to Discrete Event Systems*. Springer.
9. Chao, D.Y. (2006) Computation of elementary siphons in Petri nets for deadlock control. *Computer Journal*, vol.49, no.4, pp.470–479.
10. Chao, D.Y. (2006) Searching strict minimal siphons for SNC-based resource allocation systems, *Journal of Information Science and Engineering*, vol.23, no.3, pp.853–867.

11. Cheung, K.S. (2004) New characterization for live and reversible augmented Petri nets. *Information Processing Letters*, vol.92, no.5, pp.239–243.
12. Chu, F., Xie, X.L. (1997) Deadlock analysis of Petri nets using siphons and mathematical programming. *IEEE Transactions on Robotics and Automation*, vol.13, no.6, pp.793–804.
13. Colom, J.M., Campos, J., Silva, M. (1990) On liveness analysis through linear algebraic techniques. In *Proc. of Annual General Meeting of ESPRIT Basic Research Action 3148 Design Methods Based on Nets DEMON*.
14. Cordone, R., Ferrarini, L., Piroddi, L. (2003) Some results on the computation of minimal siphons in Petri nets. In *Proc. 42nd IEEE Conf. on Decision and Control*, pp.3754–3759.
15. Cordone, R., Ferrarini, L., Piroddi, L. (2005) Enumeration algorithms for minimal siphons in Petri nets based on place constraints. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol.35, no.6, pp.844–854.
16. Desel, J., Esparza, J. (1995) *Free Choice Petri Nets*. London: Cambridge University Press.
17. Desel, J., Reisig, W. (1998) Place/transition Petri nets. In *Lectures on Petri Nets I: Basic Models, Lecture Notes in Computer Science*, vol.1491, W. Reisig and G. Rozenberg (Eds.), pp.122–174.
18. Desel, J. (1998) Basic linear algebraic techniques for place/transition nets. In *Lectures on Petri Nets I: Basic Models, Lecture Notes in Computer Science*, vol.1491, W. Reisig and G. Rozenberg (Eds.), pp.257–308.
19. Desrocher, A.A., Al-Jaar, R.Y. (1995) *Applications of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis*, Piscataway, NJ: IEEE Press.
20. Ding, Z.J., Jiang, C.J., Zhou, M.C. (2008) Deadlock checking for one-place unbounded Petri nets based on modified reachability trees. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol.38, no.3, pp.881–882.
21. Esparza, J., Silva, M. (1992) A polynomial-time algorithm to decide liveness of bounded free choice nets. *Theoretical Computer Sciences*, vol.102, no.1, pp.185–205.
22. Ezpeleta, J., Couvreur, J.M., Silva, M. (1993) A new technique for finding a generating family of siphons, traps, and st-components: Application to colored Petri nets. In *Advances in Petri Nets, Lecture Notes in Computer Science*, vol.674, G. Rozenberg (Ed.), pp.126–147.
23. Ezpeleta, J., Colom, J.M., Martinez, J. (1995) A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Transactions on Robotics and Automation*, vol.11, no.2, pp.173–184.
24. Giua, A., Seatzu, C. (2007) A systems theory view of Petri nets. In *Advances in Control Theory and Applications, Lecture Notes in Control and Information Science*, vol.353, C. Bonivento et al. (Eds.), pp.99–127.
25. Holloway, L.E., Krogh, B.H., Giua, A. (1997) A survey of Petri net methods for controlled discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications*, vol.7, no.2, pp.151–190.
26. Hopcroft, J.E., Motwani, R., Ullman, J.D. (2000) *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., New York: Addison-Wesley.
27. Hruz, B., Zhou, M.C (2007) *Modeling and Control of Discrete-Event Dynamic Systems: With Petri Nets and Other Tools*. London: Springer.
28. Jeng, M.D, Peng, M.Y. (1999) Augmented reachability trees for 1-place-unbounded generalized Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.29, no.2, pp.173–183.
29. Jeng, M.D., Peng, M.Y., Huang, Y.S. (1999) An algorithm for calculating minimal siphons and traps in Petri nets. *International Journal of Intelligent Control and Systems*, vol.3, no.3, pp.263–275.
30. Kumar, R. Garg, V. (1995) *Modeling and Control of Logical Discrete Event Systems*. Boston, MA: Kluwer.
31. Lautenbach, K. (1987) Linear algebraic calculation of deadlocks and traps. In *Concurrency and Nets*, K. Voss, H. J. Genrich and G. Rozenberg (Eds.), pp.315–336.
32. Lautenbach, K., Ridder, H. (1993) Liveness in bounded Petri nets which are covered by T-invariants. In *Proc. 13th Int. Conf. on Applications and Theory of Petri Nets, Lecture Notes in Computer Science*, vol.815, R. Valette (Ed.), pp.358–375.

33. Li, Z.W., Zhou, M.C. (2004) Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.34, no.1, pp.38–51.
34. Li, Z.W., Wei, N. (2007) Deadlock control of flexible manufacturing systems via invariant-controlled elementary siphons of Petri nets. *International Journal of Advanced Manufacturing Technology*, vol.33, no.1–2, pp.24–35.
35. Li, Z.W., Zhou, M.C. (2008) On siphon computation for deadlock control in a class of Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, A*, vol.38, no.3, pp.667–679.
36. Minoux, M., Barkaoui, K. (1990) Deadlocks and traps in Petri nets as horn-satisfiability solutions and some related polynomially solvable problems. *Discrete Mathematics*, vol.29, no.2–3, pp.195–210.
37. Murata, T. (1989) Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE*, vol.77, no.4, pp.541–580.
38. Peterson, J.L. (1977) Petri nets. *Computing Surveys*, vol.9, no.3, pp.223–252.
39. Peterson, J.L. (1981) *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ: Prentice-Hall.
40. Piroddi, L., Cordone, R., Fumagalli, I. (2008) Selective siphon control for deadlock prevention in Petri nets, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 38, no. 6, pp.1337–1348.
41. Ramadge, P., Wonham, W.M. (1989) The control of discrete event systems. *Proceedings of the IEEE*, vol.77, no.1, pp.81–89.
42. Recalde, L., Teruel, E., Silva, M., (1998) On linear algebraic techniques for liveness analysis of P/T systems. *Journal of Circuits, Systems, and Computers*, vol.8, no.1, pp.223–265.
43. Reisig, W. (1985) *Petri Nets: An Introduction*. New York: Springer.
44. Reutenauer, C. (1990) *The Mathematics of Petri Nets*. Translated by I. Varig, Englewood Cliffs, NJ: Prentice-Hall.
45. Ru, Y., Wu, W.M., Hadjicostis, C.N. (2006) Comments on “A modified reachability tree approach to analysis of unbounded Petri nets”. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol.36, no.5, p.1210.
46. Silva, M., Teruel, E. (1996) A systems theory perspective of discrete event dynamic systems: The Petri net paradigm. In P. Borne, J. C. Gentina, E. Craye, and S. El Khattabi, (Eds.), *Symposium on Discrete Events and Manufacturing Systems*, IMACS Multiconference, Lille, France, pp.1–12.
47. Starke, P. H. (2003) *INA: Integrated Net Analyzer*. <http://www2.informatik.hu-berlin.de/~starke/ina.html>.
48. Tanimoto, S., Yamauchi, M., Watanabe, T. (1996) Finding minimal siphons in general Petri nets. *IEICE Transactions on Fundamentals*, vol.E79-A, no.11, pp.1817–1824.
49. Thistle, J.G. (1996) Supervisory control of discrete event systems. *Mathematical and Computer and Modeling*, vol.23, no.11–12, pp.25–53.
50. Tricas, F., Garacía-Vallés, F., Colom, J.M., Ezpeleta, J. (1998) A partial approach to the problem of deadlocks in processes with resources. Research Report, GISI-RR-97-05, Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain.
51. Tricas, F., García-Vallés, F., Colom, J.M., Ezpeleta, J. (1998) A structural approach to the problem of deadlock prevention in processes with shared resources. In *Proc. 4th Workshop on Discrete Event Systems*, pp.273–278.
52. Tricas, F., Ezpeleta, J. (1999) A Petri net solution to the problem of deadlocks in systems of processes with resources. In *Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation*, pp.1047–1056.
53. Tricas, F., Ezpeleta, J. (2003) Some results on siphon computation for deadlock prevention in resource allocation systems modeled with Petri nets. In *Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation*, pp.322–329.
54. Tricas, F., Ezpeleta, J. (2006) Computing minimal siphons in Petri net models of resource allocation systems: A parallel solution. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.36, no.3, pp.532–539.

55. Tsuji, K., Murata, T. (1993) On reachability conditions for unrestricted Petri nets. In *Proc. IEEE Int. Symp. on Circuits and Systems*, pp.2713–2716.
56. Wang, F.Y., Gao, Y.Q., Zhou, M.C. (2004) A modified reachability tree approach to analysis of unbounded Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol.34, no.1, pp.303–308.
57. Watanabe, T., Yamauchi, M., Tanimoto, S. (1998) Extracting siphons containing specified set of places in Petri nets. In *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp.142–147.
58. Xing, K.Y., Hu, B.S., Chen, H.X. (1996) Deadlock avoidance policy for Petri-net modelling of flexible manufacturing systems with shared resources. *IEEE Transactions on Automatic Control*, vol.41, no.2, pp.289–295.
59. Xing, K.Y., Hu, B.S. (2005) Optimal liveness Petri net controllers with minimal structures for automated manufacturing systems. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, pp.282–287.
60. Yamauchi, M., Watanabe, T. (1999) Algorithms for extracting minimal siphons containing specified places in a general Petri net. *IEICE Transactions on Fundamentals*, vol.E82-A, no.11, pp.2566–2575.
61. Zhou, M.C., DiCesare, F. (1991) Parallel and sequential exclusions for Petri net modeling for manufacturing systems. *IEEE Transactions on Robotics and Automation*, vol.7, no.4, pp.515–527.
62. Zhou, M.C., DiCesare, F. (1993) *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Boston, MA: Kluwer.