Reachability Criterion with Sufficient Test Space for Ordinary Petri Net

Gi Bum Lee, Han Zandong and Jin S. Lee

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/50518

1. Introduction

Petri nets (PN) are widely recognized as a powerful tool for modelling and analyzing discrete event systems, especially systems are characterized by synchronization, concurrency, parallelism and resource sharing [1, 2]. One of the major advantages of using Petri net models is that the PN model can be used for the analysis of behaviour properties and performance evaluation, as well as for systematic construction of discrete-event simulators and controllers [3, 4]. The reachability from an initial marking to a destination marking is the most important issue for the analysis of Petri nets. Many other problems such as liveness and coverability can be deduced from this reachability problem [5, 6].

Two basic approaches are usually applied to solve the reachability problem. One is the construction of reachability tree [7, 8]. It can obtain all the reachable markings, but the computation complexity is exponentially increased with the size of a PN. The other is to solve the state equation [9]. The solution of the matrix equation provides a firing count vector that describes the relation between initial marking and reachable markings. Its major problem is the lack of information of firing sequences and the existence of spurious solutions.

Many researchers have investigated the reachability problem [10, 11]. Iko Miyazawa *et al.* have utilized the state equation to solve the reachability problem of Petri nets with parallel structures [12]. Tadashi Matsumoto *et al*. have presented a formal necessary and sufficient condition on reachability of general Petri nets with known firing count vectors [13]. Tadao Murata's paper has concentrated on presenting and analyzing Petri nets as discrete time systems. Controllability and reachability are analyzed in terms of the matrix representation of a Petri net [14].

In most cases, it is not necessary to find all reachable markings. One of the most important things is to know whether a given marking is reachable or not. If the destination marking

$\mathbf{M}_d$ is reachable from the initial marking $\mathbf{M}_0$, it is significant to find a firing sequence, which is an ordered sequence of transitions that lead $\mathbf{M}_0$ to $\mathbf{M}_d$. The following method can be utilized to find a reachable marking [15].

i. Solve the equation $\mathbf{AX}=\mathbf{M}_d-\mathbf{M}_0$ to ascertain all the solutions $\mathbf{X}_1$, $\mathbf{X}_2$, … and construct the set X={$\mathbf{X}_1$, $\mathbf{X}_2$, …}.
ii. Test if $\mathbf{X}_i$ in X is an executable solution from $\mathbf{M}_0$, i.e. there is at least one sequence S($\mathbf{X}_i$) that is a firing sequence under $\mathbf{M}_0$.
iii. If an executable solution exists, then $\mathbf{M}_d$ is reachable. On the contrary, if X=Φ or all solutions are spurious, then $\mathbf{M}_d$ is not reachable.

However, this approach is theoretic rather than practical, because there are two problems: One is that the solution of the fundamental equation $\mathbf{AX}=\mathbf{M}_d-\mathbf{M}_0$ is infinite in some cases. In that case, it is impossible to test all solution $\mathbf{X}_i$. The other is that the computation complexity of testing $\mathbf{X}_i$ increases at least exponentially as the length of S($\mathbf{X}_i$) increases.

In this chapter, the above two problems will be solved as follows: First, we construct a sufficient test space to include at least one executable solution within set X. An approach is secondly proposed to test whether there is an executable solution within the sufficient test space or not. A systematic method to search an executable solution in a sufficient test space and to enumerate the associated firing sequence is presented.

The remainder of the chapter is arranged as follows: Definitions and notations required in this chapter are given in Section 2. Section 3 describes how to determine the sufficient test space for the reachability problem. In Section 4, an algorithm is developed to determine if $\mathbf{X}_i$ is a executable solution under $\mathbf{M}_0$ and gives the associated firing sequence S($\mathbf{X}_i$). The illustrative examples are given in Section 3, Section 4, and Section 5.

2. Preliminaries

In this section, we present some definitions and notations to be necessary in the following sections.

Definition 1. Let PN=(P, T, I, O, M_0) be a marked Petri net. P={p_1, p_2, …, p_n} is the finite set of places. T={t_1, t_2, …, t_m} is the finite set of transitions. I is the input function. O is the output function. M_0 is the initial marking.

A PN is an ordinary Petri net iff I(p, t)→{0, 1} and O(t, p)→{0, 1} for any p∈P and t∈T. $\mathbf{A}=\mathbf{O}-\mathbf{I}$ is the incidence matrix, where **O** and **I** are the output and input function matrices [16]. Let $\mathbf{X}=[x_1\ x_2\ \dots\ x_m]^T$ be a column vector. If **X** is the firing count vector of S(**X**), the sequence S(**X**) is called the transition sequence associated with **X.** The transition set T(**X**) is called the support of **X** if it is composed of transitions associated with positive elements of **X**, i.e. T(**X**)={t_i|x_i>0}. p° is the set of output transitions of p, °p is the set of input transitions of p, t° is the set of output places of t, and °t is the set of input places of t.

Definition 2. C_i=<p, T_{ci}> is called a conflict structure [17] if it satisfies the following condition: T_{ci}={t|t∈p°} and |T_{ci}|≥2, where |T_{ci}| is the cardinality of T_{ci}. We note that C={C_1, C_2, …} is the set of all C_i and $T_c=T_{c1}\cup T_{c2}\cup$…. is the set of all conflict transitions.

Definition 3. For transition t_j and $\mathbf{X}$, the sub-vector $\mathbf{H}(t_j|\mathbf{X})$ is defined as: $\mathbf{H}(t_j|\mathbf{X})=e[t_j]\cdot x_j$. $e[t_j]$ is the unit m-vector which is zero everywhere except in the j-th element.

Definition 4. For the conflict structure C_i=<p, T_{ci}> and $\mathbf{X}$, the sub-vector $\mathbf{H}(C_i|\mathbf{X})$ is defined as follows:

$$\mathbf{H}(C_i\,|\,\mathbf{X}) = \sum_{t_j\ \in T_{ci}} \mathbf{H}(t_j|\mathbf{X}) \tag{1}$$

Definition 5. C_i=<p, T_{ci}> is in a spurious conflict state for $\mathbf{X}$ under $\mathbf{M}$ if there exists a firing sequence $S(\mathbf{H}(C_i|\mathbf{X}))$ under $\mathbf{M}$, i.e. the mathematic criterion is $\mathbf{M}\geq\mathbf{I}\cdot\mathbf{H}(C_i|\mathbf{X})$.

Otherwise, C_i is in an effective conflict state for $\mathbf{X}$ under $\mathbf{M}$, and the transition in T_{ci} is called the effective conflict transition for $\mathbf{X}$ under $\mathbf{M}$.

Notation 1. $N(t_j|S(\mathbf{X}))=x_j$ is the number of occurrence times of t_j in $S(\mathbf{X})$.

Notation 2. If $q=\min\{\mathbf{M}(p_i), p_i\in{}^\circ t_j\}$, we call t_j q-enabled under marking $\mathbf{M}$. This q is denoted as $E(t_j|\mathbf{M})$.

Definition 6. $\mathbf{F}=[f_1\ f_2\ \dots\ f_m]^T$ is called an actual firing vector whose j-th element is $f_j=\min\{N(t_j|S(\mathbf{X})), E(t_j|\mathbf{M})\}$. $\mathbf{F}$ can be partitioned into two parts as follows: $\mathbf{F}=\mathbf{F}_o+\mathbf{F}_c$, where $\mathbf{F}_c=[f_{c1}\ f_{c2}\ \dots\ f_{cm}]^T$ is associated with effective conflict transitions, $\mathbf{F}_o=[f_{o1}\ f_{o2}\ \dots\ f_{om}]^T$ is associated with the other transitions. $\mathbf{F}_o$ and $\mathbf{F}_c$ satisfy the following conditions:

a. If t_j is an effective conflict transition for $\mathbf{X}$ under $\mathbf{M}$, then $f_{oj}=0$ and $f_{cj}=f_j$.
b. Otherwise, $f_{cj}=0$ and $f_{oj}=f_j$.

3. Determination of the sufficient test space

If all the solutions of the equation $\mathbf{AX}=\mathbf{M}_d-\mathbf{M}_0$ are tested, It can be found whether M_d is reachable or not. But in some case, the solutions are infinite. Therefore, the tested range is determined in order to keep the method practical. This range must be finite and include at least one executable solution if it exists. This section will discuss how to determine the tested range.

Definition 7. Given the initial marking $\mathbf{M}_0$ and the destination marking $\mathbf{M}_d$ of a PN, $\mathbf{X}$ is a solution of $\mathbf{AX}=\mathbf{M}_d-\mathbf{M}_0$. If $\mathbf{M}_d$ is reachable from $\mathbf{M}_0$ under $\mathbf{X}$, then $\mathbf{X}$ is called an executable solution. Otherwise, $\mathbf{X}$ is called a spurious solution.

Definition 8. $X=\{\mathbf{X}_1, \mathbf{X}_2, \dots\}$ is the set of a solution $\mathbf{X}$, the subset $X_e=\{\mathbf{X}_{e1}, \mathbf{X}_{e2}, \dots\}$ of X is called the sufficient test space if it satisfies following conditions:

i. If $\mathbf{M}_d$ is reachable from $\mathbf{M}_0$, there must exist at least one element in X_e which is executable solution; in other words, if all elements in X_e are not executable, then all the elements in X are not executable either.
ii. X_e is a finite set.

Definition 9. The vector $\mathbf{X}$ which is a solution of $\mathbf{AX}=0$ is known as a T-invariant [18]. A solution $\mathbf{X}$ is called positive if every element of $\mathbf{X}$ is nonnegative.

Definition 10. The positive T-invariant solution $\mathbf{U}$ of $\mathbf{AU=0}$ is minimal if it satisfies the following condition: for any other T-invariant $\mathbf{U}_i$, at least one element of $\mathbf{U}$-$\mathbf{U}_i$ is negative. The set of minimal T-invariant solutions is U={$\mathbf{U}_1$, $\mathbf{U}_2$, …, $\mathbf{U}_s$}.

Definition 11. The positive particular solution $\mathbf{V}$ of $\mathbf{AV}$=$\mathbf{M}_d$-$\mathbf{M}_0$ is minimal if it satisfies the following condition: for any T-invariant $\mathbf{U}$ of PN, there must be at least one element in $\mathbf{V}$-$\mathbf{U}$ which is negative, i.e. {$\mathbf{U}$|$\mathbf{V}$-$\mathbf{U}$≥0, $\mathbf{U}$ is a T-invariant}=Φ. The set of minimal particular solutions is V={$\mathbf{V}_1$, $\mathbf{V}_2$, …, $\mathbf{V}_q$}.

The general solution of $\mathbf{AX}$=$\mathbf{M}_d$-$\mathbf{M}_0$ must be expressed by the form of one minimal particular solution and the arbitrary linear combination of the T-invariant solutions as follows:

$$\mathbf{X} = \mathbf{V}_i + \sum_{j=1}^{r} k_j U_j \tag{2}$$

where $\mathbf{V}_i$∈V, k_j is nonnegative integer.

Algorithm 1. Interpretation of the computation for X_e.

Step 1. Solve the equation $\mathbf{AX=0}$, get all the positive integer solutions U={$\mathbf{U}_1$, $\mathbf{U}_2$, …, $\mathbf{U}_s$}, where each $\mathbf{U}_j$ (1≤j≤s) is a minimal T-invariant.

Step 2. Solve the equation $\mathbf{AX}$=$\mathbf{M}_d$-$\mathbf{M}_0$, get all the positive integer particular solutions V={$\mathbf{V}_1$, $\mathbf{V}_2$, …, $\mathbf{V}_q$}, where each $\mathbf{V}_i$ (1≤i≤q) is a minimal particular solution. B={$\mathbf{B}_1$, $\mathbf{B}_2$, …, $\mathbf{B}_n$} is a subset of V.

If V= Φ, $\mathbf{M}_d$ is not reachable, then end.

Step 3. Initialization: Let X_e=V={$\mathbf{V}_1$, $\mathbf{V}_2$ …, $\mathbf{V}_q$} and X_{temp}= Φ.

If U= Φ, then end.

Otherwise, for every $\mathbf{V}_i$, if T($\mathbf{V}_i$)⊂T($\mathbf{U}_j$), then $\mathbf{V}_i$∉B. If T($\mathbf{V}_i$)⊄T($\mathbf{U}_j$), then $\mathbf{V}_i$∈B.

Go to Step 4.

Step 4. For each pair of ($\mathbf{B}_i$, $\mathbf{U}_j$), where i=1, 2, … |B|, j=1, 2, … s, and |B| is the cardinality of set B, carry out the following operations:

If $^\circ T(\mathbf{B}_i) \cap T(\mathbf{U}_j)^\circ = \Phi$, choose the next pair of ($\mathbf{B}_i$, $\mathbf{U}_j$).

If $^\circ T(\mathbf{B}_i) \cap T(\mathbf{U}_j)^\circ \neq \Phi$ and $T(\mathbf{U}_j) \subset T(\mathbf{B}_i)$, choose the next pair of ($\mathbf{B}_i$, $\mathbf{U}_j$).

If $^\circ T(\mathbf{B}_i) \cap T(\mathbf{U}_j)^\circ \neq \Phi$ and $T(\mathbf{U}_j) \not\subset T(\mathbf{B}_i)$, then $\mathbf{D}_i$=$\mathbf{B}_i$-max($\mathbf{B}_i$)·$\mathbf{U}_j$, where max($\mathbf{B}_i$) is the maximum value of elements in $\mathbf{B}_i$.

Let $\mathbf{D}_i(r)$ be the r-th elememt of $\mathbf{D}_i$.

$\mathbf{W}_i(r)=f(\mathbf{D}_i(r))$, where $f(x) = \begin{cases} \mathbf{D}_i(r), & \text{if } \mathbf{D}_i(r) > 0 \\ 0, & \text{if } \mathbf{D}_i(r) \le 0 \end{cases}$, r=1, 2, … m.

$\sum_{r=1}^{m} (\mathbf{W}_i(r) \cdot \left| \{p \mid p \in {}^\circ t_r \cap T(\mathbf{U}_j)^\circ\} \right|) = \beta$, where $\mathbf{W}_i(r)$ is the r-th element of $\mathbf{W}_i$, m=|T|.

Add $\mathbf{B}_i$+k·$\mathbf{U}_j$, k=1, 2, … β, to X_{temp}

When all pairs of ($\mathbf{B}_i$, $\mathbf{U}_j$) have been tested, go to Step 5.

Step 5. If X_{temp}= Φ, then end.

Otherwise, Let B=X_{temp}, X_e=X_e∪B, X_{temp}= Φ, go to Step 4.

Step 1 and Step 2 are to determine all the positive integer solutions **X** for equation **AX**=$\mathbf{M}_d$-$\mathbf{M}_0$. The firing count vector of any firing sequence from $\mathbf{M}_0$ to $\mathbf{M}_d$ belongs to **X**. In Step 4, if $\mathbf{B}_i$ is not an executable solution, then there must be some transitions in T($\mathbf{B}_i$) which aren't enable it, i.e. some places in °T($\mathbf{B}_i$) are lack of tokens. In this case, if {p|p∈°T($\mathbf{B}_i$)∩T($\mathbf{U}_j$)° and T($\mathbf{U}_j$) ⊄ T($\mathbf{B}_i$)}≠ Φ, then T($\mathbf{U}_j$)° may provide tokens for °t, where t∈T($\mathbf{B}_i$). Consequently, $\mathbf{B}_i$+k·$\mathbf{U}_j$ may be an executable solution, where k=1, 2, … β. Since the number of places and transitions in PN is finite, Step 4 and Step 5 only add finite elements to X_e. Since the number of minimal T-invariants is finite, the finishing condition X_{temp}= Φ, i.e. |{p|p∈°T($\mathbf{B}_i$)∩T($\mathbf{U}_j$)° and T($\mathbf{U}_j$) ⊄ T($\mathbf{B}_i$)}|= Φ, is satisfied after all the related T-invariants have been considered. As a result of the iterative process of Step 4→Step 5→Step 4, X_e includes at least one executable solution if it exists.

The following examples show how to implement the computation algorithm. These examples illustrate that suppressing any k_i in $\mathbf{B}_i$+k·$\mathbf{U}_j$, k=1, 2, … β, may eliminate some possible executable solutions.

Example 1. When the initial marking is $\mathbf{M}_0$=(1,0,0,0,0,1,0,0,0) and the destination marking is $\mathbf{M}_d$=(1,0,0,0,0,0,0,0,1) in Figure 1, calculate the sufficient test space X_e. The ● and ○ symbols are represented as the initial and destination markings respectively.

Step 1. Solve the equation **AX**=0, get the positive integer minimal T-invariant $\mathbf{U}_1$=(1,1,1,1,0,0,0,0).

Step 2. Solve the equation **AX**=$\mathbf{M}_d$-$\mathbf{M}_0$, get the positive integer minimal particular solution V={**V**}=(0,0,0,0,1,1,1,1)

Step 3. Initialization: Let X_e=V, X_{temp}= Φ, B=X_e

Step 4-1. For (**V**, $\mathbf{U}_1$),

If T($\mathbf{U}_1$) ⊄ T(**V**), then **D**=**V**-max(**V**)·$\mathbf{U}_1$, **W**(r)=f(**D**(r)),

$$\sum_{r=1}^{8}(\mathbf{W}(r)\cdot\left|\{p\,|\,p\in{}^{\circ}t_r\cap T(\mathbf{U}_1)^{\circ}\}\right|)=3.$$

Then add **V**+$\mathbf{U}_1$, **V**+2·$\mathbf{U}_1$, **V**+3·$\mathbf{U}_1$ to the set of X_{temp},

Therefore, X_{temp}={**V**+$\mathbf{U}_1$, **V**+2·$\mathbf{U}_1$, **V**+3·$\mathbf{U}_1$}

Step 5-1. If X_{temp}≠Φ, then let B=X_{temp}={**V**+$\mathbf{U}_1$, **V**+2·$\mathbf{U}_1$, **V**+3·$\mathbf{U}_1$}

X_e=X_e∪B={**V**, **V**+$\mathbf{U}_1$, **V**+2·$\mathbf{U}_1$, **V**+3·$\mathbf{U}_1$}, X_{temp}=Φ.

Go to Step4 in Algorithm 1.

Step 4-2. For any pair of ($\mathbf{B}_i$, $\mathbf{U}_1$), T($\mathbf{U}_1$)$\subset$T($\mathbf{B}_i$) is satisfied. Therefore, $X_{temp}=\Phi$

Step 5-2. If $X_{temp}=\Phi$, then end.

As a result of above sequence, $\mathbf{M}_d$ is reachable from $\mathbf{M}_0$. The firing sequence is $t_5{*}t_1{*}t_2{*}t_6{*}t_7{*}t_3{*}t_4{*}t_8$. Its firing count vector corresponds to $\mathbf{V}+\mathbf{U}_1$=(1,1,1,1,1,1,1,1) in the sufficient test space X_e. This example shows that suppressing $\mathbf{B}_i+k\cdot\mathbf{U}_j$ (k=1) in X_e may eliminate some possible executable solution.

Example 2. Consider the PN of Figure 2, given the initial marking $\mathbf{M}_0$=(1,0,0,0,0,0,0,0,1,0) and the destination marking $\mathbf{M}_d$=(0,0,0,1,0,0,0,0,1,0), calculate the sufficient test space X_e.

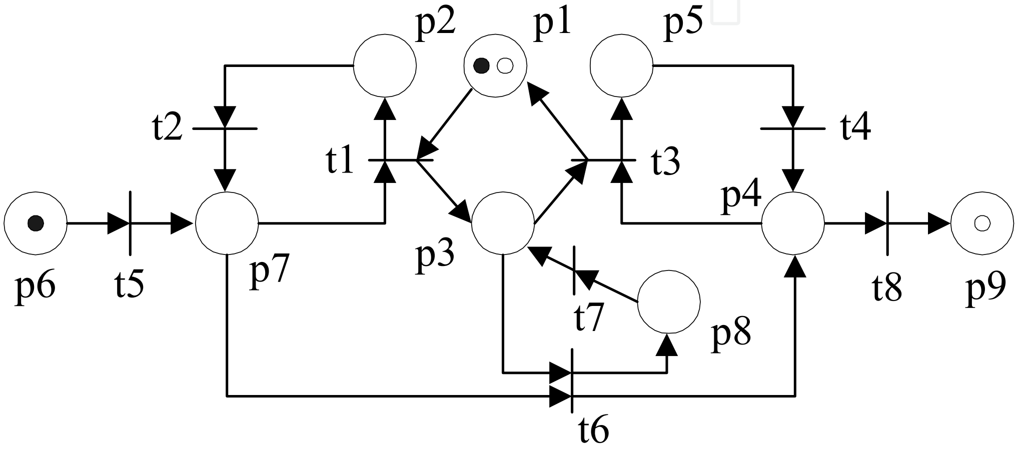

Figure 1. Petri net structure

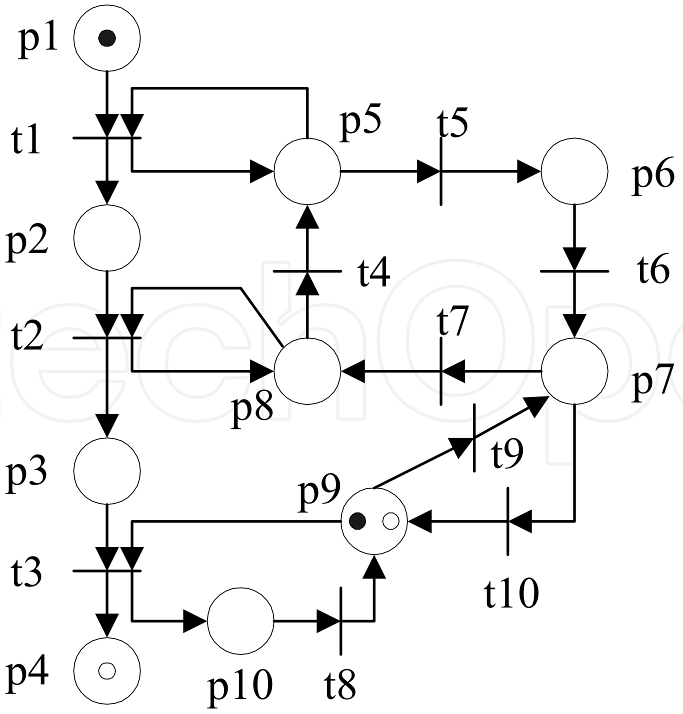

Figure 2. Petri net structure

Step 1. Solve the equation **AX**=0, two positive integer minimal T-invariants are obtained: $\mathbf{U}_1$=(0,0,0,1,1,1,1,0,0,0), $\mathbf{U}_2$=(0,0,0,0,0,0,0,0,1,1)

Step 2. Solve the equation $\mathbf{AX}=\mathbf{M}_d-\mathbf{M}_0$, get the positive integer minimal particular solutions V={**V**}={(1,1,1,0,0,0,0,1,0,0)}

The general solution can be expressed as follows:

$$\mathbf{X} = (1,1,1,0,0,0,0,1,0,0) + k_1 \cdot (0,0,0,1,1,1,1,0,0,0) + k_2 \cdot (0,0,0,0,0,0,0,0,1,1)$$

k_1 and k_2 are nonnegative integer.

Step 3. Initialization: Let X_e=V, X_{temp}=Φ, B=X_e

Step 4-1. For (**V**, $\mathbf{U}_1$),

If $T(\mathbf{U}_1) \not\subset T(\mathbf{V})$, then $\mathbf{D}=\mathbf{V}-\max(\mathbf{V})\cdot\mathbf{U}_1$, $\mathbf{W}(r)=f(\mathbf{D}(r))$,

$$\sum_{r=1}^{10}(\mathbf{W}(r)\cdot\left|\{p \mid p \in {}^{\circ}t_r \cap T(\mathbf{U}_1)^{\circ}\}\right|) = 2.$$

Then add $\mathbf{V}+\mathbf{U}_1$, $\mathbf{V}+2\cdot\mathbf{U}_1$ to X_{temp}. So $X_{temp}=\{\mathbf{V}+\mathbf{U}_1, \mathbf{V}+2\mathbf{U}_1\}$

For (**V**, $\mathbf{U}_2$),

If $T(\mathbf{U}_2) \not\subset T(\mathbf{V})$, then $\mathbf{D}=\mathbf{V}-\max(\mathbf{V})\cdot\mathbf{U}_2$, $\mathbf{W}(r)=f(\mathbf{D}(r))$,

$$\sum_{r=1}^{10}(\mathbf{W}(r)\cdot\left|\{p \mid p \in {}^{\circ}t_r \cap T(\mathbf{U}_2)^{\circ}\}\right|) = 1.$$

Then add $\mathbf{V}+\mathbf{U}_2$ to X_{temp}. So $X_{temp}=\{\mathbf{V}+\mathbf{U}_1, \mathbf{V}+2\mathbf{U}_1, \mathbf{V}+\mathbf{U}_2\}$

Step 5-1. If $X_{temp}\neq\Phi$, then let $B=X_{temp}=\{\mathbf{V}+\mathbf{U}_1, \mathbf{V}+2\mathbf{U}_1, \mathbf{V}+\mathbf{U}_2\}$,

$X_e=X_e\cup B=\{\mathbf{V}, \mathbf{V}+\mathbf{U}_1, \mathbf{V}+2\mathbf{U}_1, \mathbf{V}+\mathbf{U}_2\}$. Let's put X_{temp}=Φ.

Go to Step 4 in Algorithm 1.

Step 4-2. For ($\mathbf{V}+\mathbf{U}_1$, $\mathbf{U}_1$), because $T(\mathbf{U}_1)\subset T(\mathbf{V}+\mathbf{U}_1)$, choose the next pair.

For ($\mathbf{V}+\mathbf{U}_1$, $\mathbf{U}_2$),

If $T(\mathbf{U}_2) \not\subset T(\mathbf{V}+\mathbf{U}_1)$, then $\mathbf{D}_1=(\mathbf{V}+\mathbf{U}_1)-\max(\mathbf{V}+\mathbf{U}_1)\cdot\mathbf{U}_2$, $\mathbf{W}_1(r)=f(\mathbf{D}_1(r))$,

$$\sum_{r=1}^{10}(\mathbf{W}_1(r)\cdot\left|\{p \mid p \in {}^{\circ}t_r \cap T(\mathbf{U}_2)^{\circ}\}\right|) = 2.$$

Then add $\mathbf{V}+\mathbf{U}_1+\mathbf{U}_2$ and $\mathbf{V}+\mathbf{U}_1+2\mathbf{U}_2$ to X_{temp}. So $X_{temp}=\{\mathbf{V}+\mathbf{U}_1+\mathbf{U}_2, \mathbf{V}+\mathbf{U}_1+2\mathbf{U}_2\}$

For ($\mathbf{V}+2\mathbf{U}_1$, $\mathbf{U}_1$), because $T(\mathbf{U}_1)\subset T(\mathbf{V}+2\mathbf{U}_1)$, choose the next pair.

For ($\mathbf{V}+2\mathbf{U}_1$, $\mathbf{U}_2$),

If $T(\mathbf{U}_2) \not\subset T(\mathbf{V}+2\mathbf{U}_1)$, then $\mathbf{D}_2=(\mathbf{V}+2\mathbf{U}_1)-\max(\mathbf{V}+2\mathbf{U}_1)\cdot\mathbf{U}_2$, $\mathbf{W}_2(r)=f(\mathbf{D}_2(r))$,

$$\sum_{r=1}^{10}(\mathbf{W}_2)(r)\cdot\left|\{p\,|\,p\in {}^{\circ}t_r \cap T(\mathbf{U}_2)^{\circ}\}\right|)=3.$$

Then add $\mathbf{V}+2\mathbf{U}_1+\mathbf{U}_2$, $\mathbf{V}+2\mathbf{U}_1+2\mathbf{U}_2$, and $\mathbf{V}+2\mathbf{U}_1+3\mathbf{U}_2$ to X_{temp}. So $X_{temp}=\{\mathbf{V}+\mathbf{U}_1+\mathbf{U}_2$, $\mathbf{V}+\mathbf{U}_1+2\mathbf{U}_2$, $\mathbf{V}+2\mathbf{U}_1+\mathbf{U}_2$, $\mathbf{V}+2\mathbf{U}_1+2\mathbf{U}_2$, $\mathbf{V}+2\mathbf{U}_1+3\mathbf{U}_2\}$

For ($\mathbf{V}+\mathbf{U}_2$, $\mathbf{U}_2$), because $T(\mathbf{U}_2)\subset T(\mathbf{V}+\mathbf{U}_2)$, choose the next pair.

For ($\mathbf{V}+\mathbf{U}_2$, $\mathbf{U}_1$),

If $T(\mathbf{U}_1)\not\subset T(\mathbf{V}+\mathbf{U}_2)$, then $\mathbf{D}_3=(\mathbf{V}+\mathbf{U}_2)$-max$(\mathbf{V}+\mathbf{U}_2)\cdot\mathbf{U}_1$, $\mathbf{W}_3(r)=f(\mathbf{D}_3(r))$,

$$\sum_{r=1}^{10}(\mathbf{W}_3(r)\cdot\left|\{p\,|\,p\in {}^{\circ}t_r \cap T(\mathbf{U}_1)^{\circ}\}\right|)=3.$$

Then add $\mathbf{V}+\mathbf{U}_2+\mathbf{U}_1$, $\mathbf{V}+\mathbf{U}_2+2\mathbf{U}_1$, and $\mathbf{V}+\mathbf{U}_2+3\mathbf{U}_1$ to X_{temp}. So $X_{temp}=\{\mathbf{V}+\mathbf{U}_1+\mathbf{U}_2$, $\mathbf{V}+\mathbf{U}_1+2\mathbf{U}_2$, $\mathbf{V}+2\mathbf{U}_1+\mathbf{U}_2$, $\mathbf{V}+2\mathbf{U}_1+2\mathbf{U}_2$, $\mathbf{V}+2\mathbf{U}_1+3\mathbf{U}_2$, $\mathbf{V}+\mathbf{U}_2+3\mathbf{U}_1\}$

Step 5-2. If $X_{temp}\neq\Phi$, then let $B=X_{temp}=\{\mathbf{V}+\mathbf{U}_1+\mathbf{U}_2$, $\mathbf{V}+\mathbf{U}_1+2\cdot\mathbf{U}_2$, $\mathbf{V}+2\cdot\mathbf{U}_1+\mathbf{U}_2$, $\mathbf{V}+2\cdot\mathbf{U}_1+2\cdot\mathbf{U}_2$, $\mathbf{V}+2\cdot\mathbf{U}_1+3\cdot\mathbf{U}_2$, $\mathbf{V}+\mathbf{U}_2+3\mathbf{U}_1\}$.

So, $X_e=X_e+B=\{\mathbf{V}$, $\mathbf{V}+\mathbf{U}_1$, $\mathbf{V}+2\cdot\mathbf{U}_1$, $\mathbf{V}+\mathbf{U}_2$, $\mathbf{V}+\mathbf{U}_1+\mathbf{U}_2$, $\mathbf{V}+\mathbf{U}_1+2\cdot\mathbf{U}_2$, $\mathbf{V}+2\cdot\mathbf{U}_1+\mathbf{U}_2$, $\mathbf{V}+2\cdot\mathbf{U}_1+2\cdot\mathbf{U}_2$, $\mathbf{V}+2\cdot\mathbf{U}_1+3\cdot\mathbf{U}_2$, $\mathbf{V}+\mathbf{U}_2+3\mathbf{U}_1\}$. Let's put $X_{temp}=\Phi$.

Go to Step 4 in Algorithm 1.

Step 4-3. For any pair of ($\mathbf{B}_i$, $\mathbf{U}_j$), because $T(\mathbf{U}_j)\subset T(\mathbf{B}_i)$, $X_{temp}=\Phi$
Step 5-3. If $X_{temp}=\Phi$, then end

$\mathbf{M}_d$ is reachable from $\mathbf{M}_0$. The firing sequence is $t_9{*}t_7{*}t_4{*}t_1{*}t_5{*}t_6{*}t_7{*}t_2{*}t_4{*}t_5{*}t_6{*}t_{10}{*}t_3{*}t_8$. Its firing count vector corresponds to $\mathbf{V}+2\cdot\mathbf{U}_1+\mathbf{U}_2=(1,1,1,2,2,2,2,1,1,1)$ in the sufficient test space X_e. This example illustrates that suppressing $\mathbf{B}_i+k\cdot\mathbf{U}_j$ ($k=\beta$) in X_e may eliminate some possible executable solution.

4. Search of a firing sequence

Given the initial marking $\mathbf{M}_0$ and the destination marking $\mathbf{M}_d$ of a PN, a solution $\mathbf{X}_{ei}$ is solved from $\mathbf{AX}=\mathbf{M}_d-\mathbf{M}_0$. Then, an algorithm is developed to determine whether $\mathbf{M}_d$ is reachable from $\mathbf{M}_0$ under $\mathbf{X}_{ei}$ or not. If $\mathbf{M}_d$ is reachable from $\mathbf{M}_0$, the algorithm gives the associated firing sequence $S(\mathbf{X}_{ei})$.

Definition 12. Let $S=t_1t_2\ldots t_r$ be a finite transition sequence. The transitions appearing in S are defined by the set $Z(S)=\{t_1, t_2, \ldots, t_r\}$. The set of transitions $Z(S)$ is called a sequence component. $Z(S)$ is the set of elements that appear in a transition sequence S.

Algorithm 2. Search of a firing sequence $S(\mathbf{X}_{ei})$ under $\mathbf{M}_0$

Step 1. According to **I**, determine all the conflict structure $C_i=\langle p, T_{ci}\rangle$, and construct T_c and C.

Step 2. Initialization: Let M=$\mathbf{M}_0$, X=$\mathbf{X}_{ei}$, S=λ (λ is the sequence of length zero)

Step 3. Under M and X, calculate $\mathbf{F}$=$\mathbf{F}_o$+$\mathbf{F}_c$ from Definition 6.

If $\mathbf{F}_o \neq \mathbf{0}$, go to Step 4.

If $\mathbf{F}_o$ =$\mathbf{0}$ and $\mathbf{F}_c \neq 0$, go to Step 5.

If $\mathbf{F}$=$\mathbf{0}$, go to step 6.

Step 4. If $\mathbf{F}_o \neq 0$, then there exists an S($\mathbf{F}_o$) that has a firing sequence under M. Therefore, S($\mathbf{F}_o$) can be fired. The reachable marking is calculated by $\mathbf{M}'$=M-$\mathbf{A}\cdot\mathbf{F}_o$,

Let M=$\mathbf{M}'$, X=X–$\mathbf{F}_o$, S=S*S($\mathbf{F}_o$), where * is concatenation operation and S*S($\mathbf{F}_o$) means S followed by S($\mathbf{F}_o$). Go to Step 3.

Step 5. $\mathbf{F}_o$ =0 and $\mathbf{F}_c \neq 0$ means that all transitions in S($\mathbf{F}_c$) are effective conflict transitions. Therefore, branching occurs and the number of branches is |T($\mathbf{F}_c$)|. From here, the computation has to consider all |T($\mathbf{F}_c$)| branches.

After selecting a transition $t_j \in$T($\mathbf{F}_c$), fire it, then the reachable marking is calculated by $\mathbf{M}'$=$\boldsymbol{M}$-$\mathbf{A}\cdot\boldsymbol{e}$[$t_j$].

Let $\boldsymbol{M}$=$\mathbf{M}'$, $\boldsymbol{X}$=$\boldsymbol{X}$–$\boldsymbol{e}$[t_j], S=S*t_j.Go to Step 3

Step 6. If X=0, then $\mathbf{M}_d$ is reachable from $\mathbf{M}_0$ and S=S($\mathbf{X}_{ei}$) is one of the firing sequences, end. Otherwise, go to Step 7.

Step 7. If all the branches in Step 5 have been implemented, then $\mathbf{M}_d$ is not reachable, end. Otherwise, go to Step 5 and implement the remaining branches.

The validity of the above algorithm is proved as the following four cases:

Base: Let $\mathbf{X}$ be a solution of $\mathbf{AX}$=$\mathbf{M}_d$-$\mathbf{M}_0$. The actual firing vector $\mathbf{F}$=$\mathbf{F}_o$+$\mathbf{F}_c$ is obtained with $\mathbf{M}$ and $\mathbf{X}$. Let $t_o \in$T($\mathbf{F}_o$) and $t_c \in$T($\mathbf{F}_c$).

Case 1: If $\mathbf{F}_o \neq \mathbf{0}$ and $\mathbf{F}_c$=$\mathbf{0}$, then multiple firing of S($\mathbf{F}_o$) doesn't affect a firing sequence associated with $\mathbf{X}$ under $\mathbf{M}_0$, for the input places of T($\mathbf{F}_o$) don't affect the enabling condition of other transitions in T($\mathbf{X}$) except transitions in T($\mathbf{F}_o$).

Case 2: If $\mathbf{F}_o$=$\mathbf{0}$ and $\mathbf{F}_c \neq \mathbf{0}$, then the firing of each transition in S($\mathbf{F}_c$) is considered as a branch and implemented with respect to all branches. It means that all possibilities are involved. So, Algorithm 2 doesn't eliminate any possible firing sequence.

Case 3: If $\mathbf{F}_o$=$\mathbf{0}$ and $\mathbf{F}_c$=$\mathbf{0}$, then no transition is enabled.

Case 4: If $\mathbf{F}_o \neq \mathbf{0}$ and $\mathbf{F}_c \neq \mathbf{0}$, then the multiple firing of S($\mathbf{F}_o$) can be implemented before S($\mathbf{F}_c$). It doesn't eliminate any probability of finding a firing sequence associated with $\mathbf{X}$ under $\mathbf{M}_0$. It is proven in Proposition 1.

Proposition 1. If $\sigma \in$S($\mathbf{X}$) is a firing sequence under $\mathbf{M}_0$, then (S($\mathbf{F}_o$)*σ')$\in$S($\mathbf{X}$) is a firing sequence under $\mathbf{M}_0$ for any sequence σ'.

Proof:

Step 1. Let $T(\mathbf{F}_o)=\{t_{o1}, t_{o2}, \dots, t_{on}\}$. For a transition $t_{o1}\in T(\mathbf{F}_o)$, σ can be represented as $\sigma=\sigma_1{*}t_o{*}\sigma_2$, where $t_{o1}\notin Z(\sigma_1)$. Then $\mathbf{M}_0 \xrightarrow{\sigma_1} \mathbf{M}_1 \xrightarrow{t_{o1}} \mathbf{M}_2 \xrightarrow{\sigma_2} \mathbf{M}_d$ is a firing sequence. Since $T(\mathbf{F}_o)$ is the set of transitions possible to be enabled under $\mathbf{M}_0$, $\mathbf{M}_0$ enables t_{o1}. Therefore it is possible to put $\mathbf{M}_0 \xrightarrow{t_{o1}} \mathbf{M}_3$. By the definition of $\mathbf{F}_o$, we have $M_3(p)\geq M_0(p)$ for any $p\in{}^{\circ}Z(\sigma_1)$. So σ_1 is enabled under M_3 because σ_1 is enabled under M_0 (Monotonicity Lemma). After σ_1 firing, $\mathbf{M}_2$ is reachable from $\mathbf{M}_3$. Therefore, we have $\mathbf{M}_0 \xrightarrow{t_{o1}} \mathbf{M}_3 \xrightarrow{\sigma_1} \mathbf{M}_2$. Since σ_2 is enabled under M_2, $t_{o1}{*}\sigma_1{*}\sigma_2$ is a firing sequence under $\mathbf{M}_0$.

Step 2. Under $\mathbf{M}_3$, let's consider the new $T(\mathbf{F}_o)=\{t_{o2}, \dots, t_{on}\}\cup T(\mathbf{F}_o')$, where $T(\mathbf{F}_o')$ is the set of transition generated after t_{o1} firing and may be empty. For a transition $t_{o2}\in T(\mathbf{F}_o)$, $\sigma_1{*}\sigma_2$ can be represented as $\sigma_1{*}\sigma_2=\sigma_3{*}t_{o2}{*}\sigma_4$, where $t_{o2}\notin Z(\sigma_3)$. Then $\sigma_3{*}t_{o2}{*}\sigma_4$ is a firing sequence. By the same way described in Step 1, we can prove that $t_{o2}{*}\sigma_3{*}\sigma_4$ is a firing sequence under $\mathbf{M}_3$.

Step 3. By Step 1 and Step 2, $t_{o1}{*}t_{o2}{*}\sigma_3{*}\sigma_4$ is a firing sequence under $\mathbf{M}_0$.

Step 4. In the same way, it is proven that $t_{o1}{*}t_{o2}{*}\dots{*}t_{on}{*}\sigma_i{*}\sigma_j$ is a firing sequence under $\mathbf{M}_0$. According to the definition of $\mathbf{F}_o$, all transitions in $\{t_{o1}, t_{o2}, \dots, t_{on}\}$ can fire simultaneously under $\mathbf{M}_0$. Let's put $\sigma'=\sigma_i{*}\sigma_j$, then $(S(F_o){*}\sigma')\in S(\mathbf{X})$ is a firing sequence under $\mathbf{M}_0$.

Example 3. Let us now apply the proposed algorithm to the PN of Figure 3. Given $\mathbf{M}_0=(0,0,0,0,0,0,1,0,0)$, $\mathbf{M}_d=(0,0,0,0,1,0,1,0,0)$ and $\mathbf{X}=(1,2,1,1,1,1,1)$, determine if $\mathbf{M}_d$ is reachable or not under $\mathbf{M}_0$ and $\mathbf{X}$.

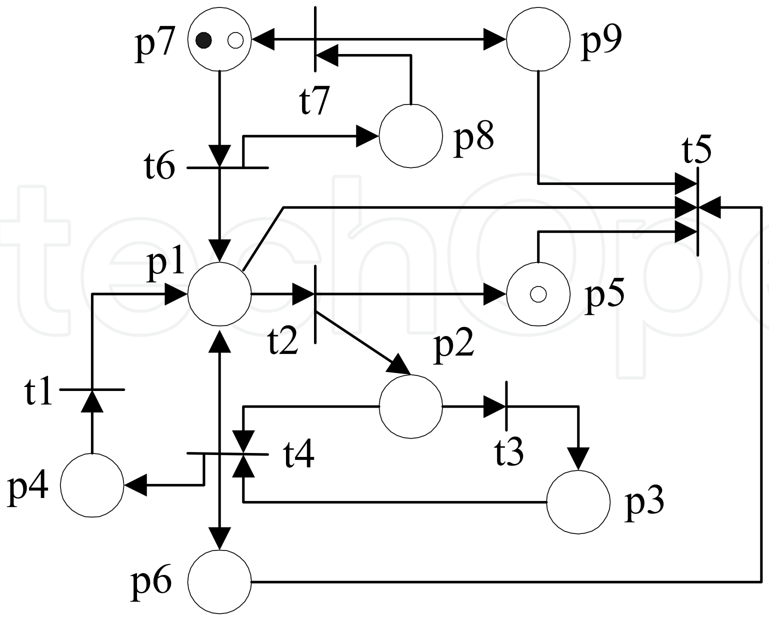

Figure 3. Petri net structure

Step 1. There are two conflict structures, C_1=<p_1, {t_2, t_5}>, C_2=<p_2, {t_3, t_4}>, T_c={t_2, t_3, t_4, t_5}.
Step 2. Initialization: M=$\mathbf{M}_0$= (0,0,0,0,0,0,1,0,0), X=**X**=(1,2,1,1,1,1,1), S=λ
Step 3. Under M and X, only t_6 is 1-enabled. Then, $\mathbf{F}_o$=(0,0,0,0,0,1,0).
Step 4. Fire S($\mathbf{F}_o$)= t_6. Then the reachable marking **M′** becomes (1,0,0,0,0,0,0,1,0)

Let $\boldsymbol{M}$=**M′**, $\boldsymbol{X}$=$\boldsymbol{X}$-$\mathbf{F}_o$=(1,2,1,1,1,0,1), S=t_6. Go to Step 3 in Algorithm 2.

Step 3-1. Under M and X, $\mathbf{F}_o$=(0,0,0,0,0,0,1).
Step 4-1. Fire S($\mathbf{F}_o$)=t_7. Then, the reachable marking becomes **M′**=(1,0,0,0,0,0,1,0,1)

Let $\boldsymbol{M}$=**M′**, $\boldsymbol{X}$=$\boldsymbol{X}$-$\mathbf{F}_o$=(1,2,1,1,1,0,0), S=t_6*t_7. Go to Step 3 in Algorithm 2.

Step 3-2. Under $\boldsymbol{M}$ and $\boldsymbol{X}$, $\mathbf{F}_o$=(0,1,0,0,0,0,0). Go to Step 4 in Algorithm 2.
Step 4-2. Fire S($\mathbf{F}_o$)=t_2 (t_2 is not an effective conflict transition because t_5 cannot enable),

then the reachable marking becomes **M′**=(0,1,0,0,1,0,1,0,1)

Let $\boldsymbol{M}$=**M′**, $\boldsymbol{X}$=$\boldsymbol{X}$-$\mathbf{F}_o$=(1,1,1,1,1,0,0), S= t_6*t_7*t_2. Go to Step 3 in Algorithm 2.

Step 3-3. Under $\boldsymbol{M}$ and $\boldsymbol{X}$, $\mathbf{F}_o$=(0,0,1,0,0,0,0). Go to Step 4 in Algorithm 2.
Step 4-3. Fire S($\mathbf{F}_o$)=t_3 (t_3 is not an effective conflict transition because t_4 cannot enable),

then the reachable marking becomes **M′**=(0,0,1,0,1,0,1,0,1)

Let $\boldsymbol{M}$=**M′**, $\boldsymbol{X}$=$\boldsymbol{X}$-$\mathbf{F}_o$=(1,1,0,1,1,0,0), S= t_6*t_7*t_2*t_3. Go to Step 3 in Algorithm 2.

Step 3-4. Under $\boldsymbol{M}$ and $\boldsymbol{X}$, **F**=**0,** go to Step 6 in Algorithm 2.

Step 6. Because X≠**0,** go to Step 7 in Algorithm 2.

Step 7. There is no effective conflict transition i.e., no branch. Consequently, $\mathbf{M}_d$ is not reachable under **X** because X≠**0**.

The above implementing process can be presented by a firing path tree as shown in Figure 4.

5. Application of Reachability Criterion

An example will be given to illustrate how to use the proposed method of Algorithm 1 and Algorithm 2 to solve the reachability problem.

Example 4. When the initial marking is $\mathbf{M}_0$=(1,0,0,0,0,0,0,0,0,1) in the PN of Figure 5, is the destination marking $\mathbf{M}_d$=(0,0,1,0,1,0,0,0,1) reachable from $\mathbf{M}_0$?

First, calculate sufficient test space using the following steps:

Step 1. Solve the equation **AX**=0, get one positive integer minimal T-invariant **U**=(0,0,0,0,0,0,1,1).
Step 2. Solve the equation **AX**=$\mathbf{M}_d$-$\mathbf{M}_0$, get the positive integer minimal particular solutions $\mathbf{V}_1$=(0,2,1,0,2,2,0,0), $\mathbf{V}_2$=(2,2,1,2,0,0,0,0) and $\mathbf{V}_3$=(1,2,1,1,1,1,0,0)
Step 3. Initialization: Let X_e={$\mathbf{V}_1$, $\mathbf{V}_2$, $\mathbf{V}_3$}, X_{temp}=Φ, B=X_e
Step 4-1. For ($\mathbf{V}_1$, **U**),

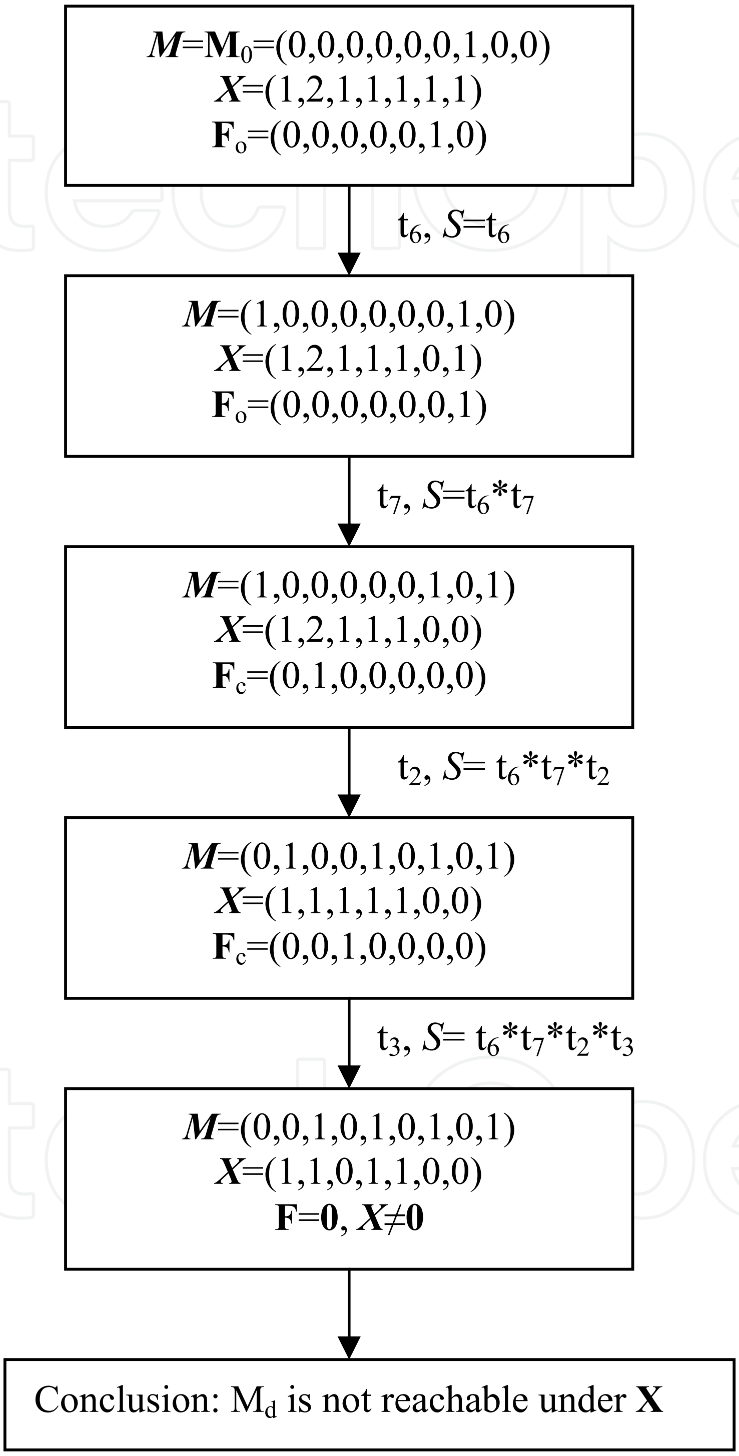
$\boldsymbol{M}=\mathbf{M}_0=(0,0,0,0,0,0,1,0,0)$
$\boldsymbol{X}=(1,2,1,1,1,1,1)$
$\mathbf{F}_o=(0,0,0,0,0,1,0)$
t_6, $S=t_6$
$\boldsymbol{M}=(1,0,0,0,0,0,0,1,0)$
$\boldsymbol{X}=(1,2,1,1,1,0,1)$
$\mathbf{F}_o=(0,0,0,0,0,0,1)$
t_7, $S=t_6{*}t_7$
$\boldsymbol{M}=(1,0,0,0,0,0,1,0,1)$
$\boldsymbol{X}=(1,2,1,1,1,0,0)$
$\mathbf{F}_c=(0,1,0,0,0,0,0)$
t_2, $S= t_6{*}t_7{*}t_2$
$\boldsymbol{M}=(0,1,0,0,1,0,1,0,1)$
$\boldsymbol{X}=(1,1,1,1,1,0,0)$
$\mathbf{F}_c=(0,0,1,0,0,0,0)$
t_3, $S= t_6{*}t_7{*}t_2{*}t_3$
$\boldsymbol{M}=(0,0,1,0,1,0,1,0,1)$
$\boldsymbol{X}=(1,1,0,1,1,0,0)$
$\mathbf{F}=\mathbf{0}$, $\boldsymbol{X}\neq\mathbf{0}$
Conclusion: M_d is not reachable under $\mathbf{X}$

Figure 4. Firing path tree on reachability of Figure 3.

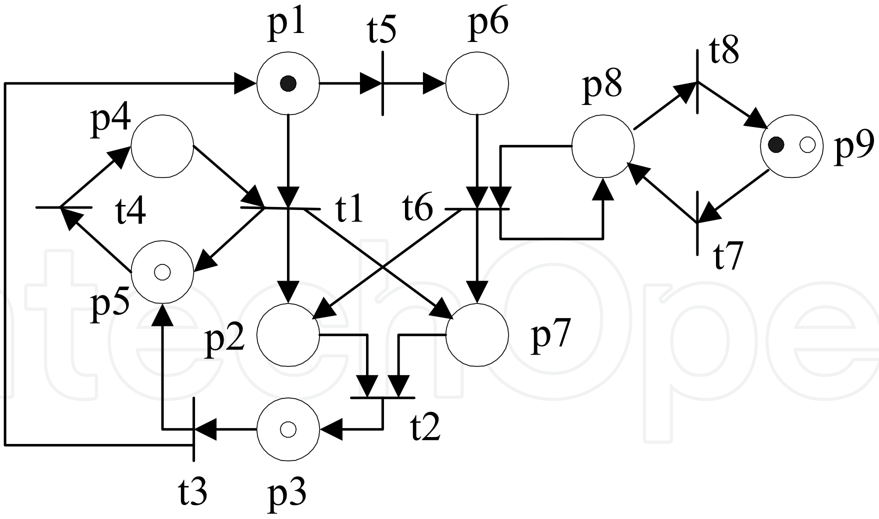

Figure 5. Petri net structure

If $T(\mathbf{U}) \not\subset T(\mathbf{V}_1)$, then $\mathbf{D}_1=\mathbf{V}_1\text{-max}(\mathbf{V}_1)\cdot\mathbf{U}$, $\mathbf{W}_1(r)=f(\mathbf{D}_1(r))$,

$$\sum_{r=1}^{8}(\mathbf{W}_1(r)\cdot\left|\{p\,|\,p\in{}^{\circ}t_r\cap T(\mathbf{U})^{\circ}\}\right|)=2$$

Then add $\mathbf{V}_1+\mathbf{U}$, $\mathbf{V}_1+2\cdot\mathbf{U}$ to X_{temp}. Then, $X_{temp}=\{\mathbf{V}_1+\mathbf{U}, \mathbf{V}_1+2\cdot\mathbf{U}\}$

For ($\mathbf{V}_2$, $\mathbf{U}$), because $^{\circ}T(\mathbf{V}_2)\cap T(\mathbf{U})^{\circ}=\Phi$, choose the next pair.

For ($\mathbf{V}_3$, $\mathbf{U}$),

If $T(\mathbf{U}) \not\subset T(\mathbf{V}_3)$, then $\mathbf{D}_3=\mathbf{V}_3\text{-max}(\mathbf{V}_3)\cdot\mathbf{U}$, $\mathbf{W}_3(r)=f(\mathbf{D}_3(r))$,

$$\sum_{r=1}^{8}(\mathbf{W}_3(r)\cdot\left|\{p\,|\,p\in{}^{\circ}t_r\cap T(\mathbf{U})^{\circ}\}\right|)=1$$

Then add $\mathbf{V}_3+\mathbf{U}$ to X_{temp}, $X_{temp}=\{\mathbf{V}_1+\mathbf{U}, \mathbf{V}_1+2\cdot\mathbf{U}, \mathbf{V}_3+\mathbf{U}\}$

Step 5-1. If $X_{temp}\neq\Phi$, then let $B=X_{temp}=\{\mathbf{V}_1+\mathbf{U}, \mathbf{V}_1+2\cdot\mathbf{U}, \mathbf{V}_3+\mathbf{U}\}$,

$X_e=X_e\cup B=\{\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3, \mathbf{V}_1+\mathbf{U}, \mathbf{V}_1+2\cdot\mathbf{U}, \mathbf{V}_3+\mathbf{U}\}$. Let's put $X_{temp}=\Phi$. Go to Step 4 in Algorithm 1.

Step 4. For any pair of ($\mathbf{B}_i$, $\mathbf{U}$), because $T(\mathbf{U}_j)\subset T(\mathbf{B}_i)$, $X_{temp}=\Phi$.

Step 5. If $X_{temp}=\Phi$, then end.

Consequently, the sufficient test space becomes $X_e=\{\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3, \mathbf{V}_1+\mathbf{U}, \mathbf{V}_1+2\cdot\mathbf{U}, \mathbf{V}_3+\mathbf{U}\}$.

Second, calculate a firing sequence in order to test if M(d) is reachable from M(0) under some element in X_e

The elements of the sufficient test space X_e are calculated separately as follows:

Step 1. For $\mathbf{X}=\mathbf{V}_1=(0,2,1,0,2,2,0,0)$

The implementing process is shown in Figure 6.

Step 2. For $\mathbf{X}=\mathbf{V}_2=(2,2,1,2,0,0,0,0)$

Carrying out the same process, the conclusion is as follows: M_d is not reachable under $\mathbf{V}_2$.

Step 3. For $\mathbf{X}=\mathbf{V}_3=(1,2,1,1,1,1,0,0)$

Carrying out the same process, the conclusion is as follows: M_d is not reachable under $\mathbf{V}_2$.

Step 4. For $\mathbf{X}=\mathbf{V}_1+\mathbf{U}=(0,2,1,0,2,2,1,1)$

Carrying out the same process shown in Figure 7, the conclusion is as follows: M_d is reachable from M_0 under $\mathbf{V}_1+\mathbf{U}$**.** $\mathbf{V}_1+\mathbf{U}$ is an executable solution in X_e, and the firing sequence is $t_5*t_7*t_6*t_2*t_3*t_5*t_6*t_2*t_8$.

As a result of calculating each element of the sufficient test space $X_e=\{\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3, \mathbf{V}_1+\mathbf{U}, \mathbf{V}_1+2\cdot\mathbf{U}, \mathbf{V}_3+\mathbf{U}\}$ individually, a firing sequence is finally found at the fourth element ($\mathbf{V}_1+\mathbf{U}$) of X_e**.** Therefore, the elements $\mathbf{V}_1+2\cdot\mathbf{U}$ and $\mathbf{V}_3+\mathbf{U}$ don't need to be calculated. Consequently, the structure of the Petri net (Figure 5) is shown to possess at least one reachable firing sequence.

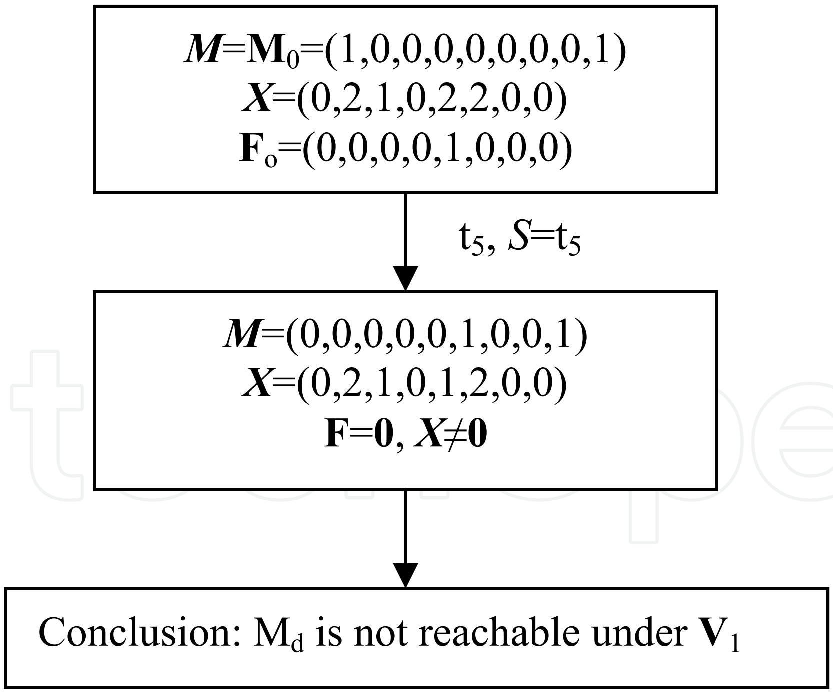

Figure 6. Firing path tree for $\mathbf{V}_1$.

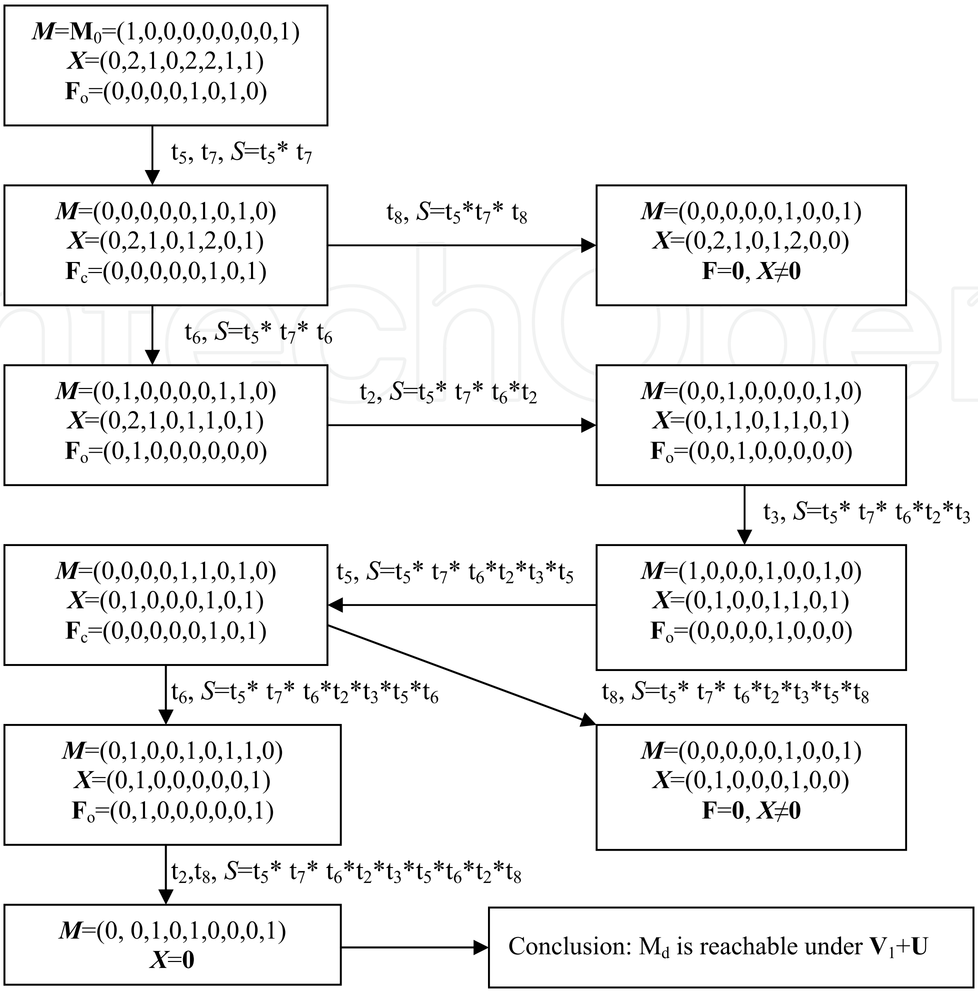
$\boldsymbol{M}=\mathbf{M}_0=(1,0,0,0,0,0,0,0,1)$
$\boldsymbol{X}=(0,2,1,0,2,2,1,1)$
$\mathbf{F}_o=(0,0,0,0,1,0,1,0)$
$t_5, t_7, S=t_5* t_7$
$\boldsymbol{M}=(0,0,0,0,0,1,0,1,0)$
$\boldsymbol{X}=(0,2,1,0,1,2,0,1)$
$\mathbf{F}_c=(0,0,0,0,0,1,0,1)$
$t_8, S=t_5*t_7* t_8$
$\boldsymbol{M}=(0,0,0,0,0,1,0,0,1)$
$\boldsymbol{X}=(0,2,1,0,1,2,0,0)$
$\mathbf{F}=\mathbf{0}, \boldsymbol{X}\neq\mathbf{0}$
$t_6, S=t_5* t_7* t_6$
$\boldsymbol{M}=(0,1,0,0,0,0,1,1,0)$
$\boldsymbol{X}=(0,2,1,0,1,1,0,1)$
$\mathbf{F}_o=(0,1,0,0,0,0,0,0)$
$t_2, S=t_5* t_7* t_6*t_2$
$\boldsymbol{M}=(0,0,1,0,0,0,0,1,0)$
$\boldsymbol{X}=(0,1,1,0,1,1,0,1)$
$\mathbf{F}_o=(0,0,1,0,0,0,0,0)$
$t_3, S=t_5* t_7* t_6*t_2*t_3$
$\boldsymbol{M}=(1,0,0,0,1,0,0,1,0)$
$\boldsymbol{X}=(0,1,0,0,1,1,0,1)$
$\mathbf{F}_o=(0,0,0,0,1,0,0,0)$
$t_5, S=t_5* t_7* t_6*t_2*t_3*t_5$
$\boldsymbol{M}=(0,0,0,0,1,1,0,1,0)$
$\boldsymbol{X}=(0,1,0,0,0,1,0,1)$
$\mathbf{F}_c=(0,0,0,0,0,1,0,1)$
$t_6, S=t_5* t_7* t_6*t_2*t_3*t_5*t_6$
$t_8, S=t_5* t_7* t_6*t_2*t_3*t_5*t_8$
$\boldsymbol{M}=(0,1,0,0,1,0,1,1,0)$
$\boldsymbol{X}=(0,1,0,0,0,0,0,1)$
$\mathbf{F}_o=(0,1,0,0,0,0,0,1)$
$\boldsymbol{M}=(0,0,0,0,0,1,0,0,1)$
$\boldsymbol{X}=(0,1,0,0,0,1,0,0)$
$\mathbf{F}=\mathbf{0}, \boldsymbol{X}\neq\mathbf{0}$
$t_2,t_8, S=t_5* t_7* t_6*t_2*t_3*t_5*t_6*t_2*t_8$
$\boldsymbol{M}=(0, 0,1,0,1,0,0,0,1)$
$\boldsymbol{X}=\mathbf{0}$
Conclusion: M_d is reachable under $\mathbf{V}_1+\mathbf{U}$

Figure 7. Firing path tree for $\mathbf{V}_1+\mathbf{F}$.

6. Conclusions

In this chapter, a new general criterion has been created to solve the reachability problems for ordinary Petri nets. This criterion is based on two processes: (i) Calculating the sufficient test space. (ii) Testing whether or not the destination marking is reachable from the initial marking under the sufficient test space. The sufficient test space significantly reduces the quantity of computation needed to search for an executable solution in X. The firing path tree shows the firing sequence of an executable solution. Consequently, if the destination marking is reachable from the initial marking, this method gives at least one firing sequence that leads from the initial marking to the destination marking. Some examples are given to illustrate how to use this method to solve the reachability problem. This algorithm can be utilized in the following fields: Path searching, auto routing, and reachability between any places in a complicated network.

Author details

Gi Bum Lee
Research Institute of Industrial Science & Technology, Pohang, Korea

Han Zandong
Tsinghua University, Beijing, P.R. China

Jin S. Lee
Pohang University of Science and Technology, Pohang, Korea

7. References

[1] Frank L. Lewis, Ayla Gurel, Stjepan Bogdan, Alper Doganalp and Octavian C. Pastravanu (1998) Analysis of deadlock and circular waits using a matrix model for flexible manufacturing systems. Automatica. 34(9): 1083-1100.

[2] Tadao Murata (1989) Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE . 77: 541-580.

[3] Gi Bum Lee, Han Zandong, Jin S. Lee (2004) Automatic generation of ladder diagram with control Petri Net. Journal of Intelligent Manufacturing. 15(2): 245-252.

[4] MengChu Zhou (1995) Petri nets in flexible and agile automation [M]. Boston: Kluwer Academic Publishers Group.

[5] Ernst W. Mayr (1984) An algorithm for the general Petri net reachability problem. SIAM J. COMPUT. 13(3): 441-449.

[6] Jeng S. Huang and Tadao Murata (1997) Classifications of Petri Net Transitions and Their Application to Firing Sequence and Reachability Problems. IEEE International Conference on Systems, Man, and Cybernatics. 1: 263-268.

[7] Jorg Desel and Javier Esparza (1995) Free choice Petri nets: Cambridge University Press.

[8] Kunihiko Hiraishi (2000) An Efficient Algorithm for Exploring State Spaces of Petri Nets with Large Capacities. IEICE Trans. Fundamentals. E83-A(11): 2188-2195.

[9] Karsten Schmidt (2001) Narrowing Petri Net State Spaces Using the State Equation. Fundamenta Informaticae. 47: 325-335.

[10] Alexander E. Kostin and Svetlana A. Tchoudaikina (1998) Yet Another Reachability Algorithm for Petri nets. SIGACT News. 29(4): 98-110.

[11] Toshiro ARAKI and Tadao KASAMI (1977) Some Decision Problems Related to the Reachability Problem for Petri nets. Theoretical Computer Science. 3: 85-104.

[12] Iko Miyazawa, Haruki Tanaka, and Takashi Sekiguchi (1996) Classification of solutions of matrix equation related to parallel structure of a Petri Net. IEEE Conference on Emerging Technologies and Factory Automation: 446-452.

[13] Tadashi Matsumoto and Yasushi Miyano (1998) Reachability criterion for Petri Nets with known firing count Vectors. IEICE Trans. Fundamentals. E81-A(4): 628-634.

[14] Jaegeol Yim, Peter C. Nelson, and Tadao Murata (1994) Predicate-Transition Net Reachability Testing Using Heuristic Search. T. IEE Japan. 114-C(9): 907-913.

[15] David (1992) Petri net and Grafcet: Prentice Hall International Ltd.

[16] Gi Bum Lee, Jin S. Lee (2002) Conversion of LD program into augmented PN graph. International Journal of Modelling & Simulation. 22(4): 201-212.

[17] Gi Bum Lee, Han Zandong, Jin S. Lee (2003) Generalized State Equation of Petri nets with Priority. International Journal of Intelligent Systems. 18(11): 1145-1153.

[18] Peterson (1981) Petri net theory and the modeling of systems: London: Prentice Hall international (UK) Ltd.