

페트리 넷(Petri nets)

남 부 회*

(*강원대학교 제어계측공학과 교수)

1. 개 요

페트리(Carl Adam Petri)가 1962년 학위 논문에서 이산사건 시스템(discrete event system)의 해석으로 페트리넷(Petri nets)를 제시한 이래 많은 분야에서 응용되고 있다. 페트리넷은 상호 작용하는 동시발생 구성요소를 갖는 이산사건 시스템을 모델링하고 설계할 수 있는 그래프 이론적인 그리고 시각적인 도구이다. 페트리넷은 동시적(concurrent), 비동기적(asynchronous), 분산적(distributed), 병렬적(parallel), 비확정적(nondeterministic), 그리고 확률적(Stochastic)인 현상을 갖는 정보처리, 작업공정 시스템 등을 묘사하고 분석, 연구하는데에 유익한 도구이다. 그래픽 도구(Graphic tool)로서, 페트리넷은 플로우차트(flow chart)나 블럭다이아그램(block diagram) 또는 네트워크(network)과 유사한 시각적인 정보매체로 사용될 수 있다. 또한 정보의 흐름을 나타내는 토큰(token)은 시스템의 동적이고 동시발생적인 상황과 활동을 모의실험 하는데에 사용될 수 있다. 수학적 도구로서는 상태 방정식, 대수 방정식, 그리고 시스템의 행태를 결정짓는 다른 수학적 모델 을 정립할 수 있다.

이산사건시스템으로 이산부품의 제조공정, 통신망, 그리고 컴퓨터시스템등이 있으며, 페트리넷은 이산사건시스템의 명확한 규정(formal specification), 명확한 확인(formal verification), 그리고 성능평가(performance evaluation)에 대한 통

일된 환경을 제공한다.

페트리넷은 시스템의 관계를 시작적으로 보여 주어 시스템의 모델링을 쉽게 만든다. 시스템의 각 부분에 관한 정보를 가지고 전체와 부분에 대하여 하향식(top-down)과 상향식(bottom-up) 설계를 할 수 있게 하여준다. 페트리넷은 그래프 표현이므로 FMS(flexible manufacturing systems) 등과 같은 실시간제어(real-time control)의 구현에 사용될 수 있으며 토큰 정보로부터 시스템의 각 부분의 상태를 실시간 감시(monitoring)할 수 있다. 페트리넷은 PLC(programmable logic controller)와 같이 여러 요소시스템(subsystem)을 순서대로 동작시키고 조정한다. 시스템에 대한 페트리넷 모델로 부터 시뮬레이션(simulation)을 할 수 있고, 이로부터 성능 평가를 할 수 있다.

2. 페트리넷의 정의

페트리넷은 5개의 원소로 구성되어 있다. 여기서,

$$N = \{P, T, I, O, M_0\}$$

$P = \{P_1, P_2, \dots, P_m\}$ 는 유한한 플레이스의 집합

$T = \{t_1, t_2, \dots, t_n\}$ 는 유한한 트랜지션의 집합

플레이스(place : circle) P : 조건, 자원의 사용 가능성 또는 공정상태

트랜지션(transition : bar) T : 사건(event) 또는 공정(processing)의 시작과 끝

입력함수(input function) I : 플레이스로부터 트

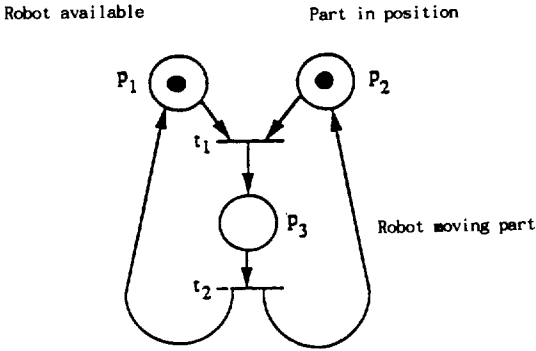


그림 1. 로봇이 부품을 가공하는 공정에 대한 페트리네트

랜지선으로의 아크(arc /arrow)

예) p_1 으로부터 t_1 으로의 아크는 p_1 을 t_1 의 입력 플레이스로 정의한다.

출력함수(output function) O :트랜지선으로부터 플레이스로의 아크

예) t_1 에서 p_3 로의 아크는 p_3 를 t_1 으로부터의 출력플레이스로 정의한다.

마킹(marking) M : 각 플레이스에 있는 토큰의 갯수(M_0 : 초기 마킹)

2.1 실행규칙

활성화 규칙(enabling rule) : 트랜지선의 모든 입력 플레이스에 토큰(token)이 있을 때, 트랜지선은 활성화(enabled)되고 점화(fire)할 수 있다.

점화 규칙(firing rule) : 활성화된 트랜지선이 점화하면, 모든 입력 플레이스에서 하나의 토큰이 제거되고, 모든 출력 플레이스에 하나의 토큰이 더해진다.

입력 플레이스가 없는 트랜지선을 생성(source) 트랜지선이라 하고, 출력 플레이스가 없는 트랜지선을 소멸(sink) 트랜지선이라 한다. 그러므로 생성 트랜지선은 무조건 활성화되어 있고, 소멸 트랜지선을 점화하는 것은 네트에서 토큰의 소멸을 가져온다.

2.2 페트리네트의 종류

- 상 페트리네트(ordinary PN) : 기본기능만을 갖는 페트리네트
- 시간 페트리네트(timed PN)
 - Timed Transition Petri Net

- Timed Place Petri Net
- 일반화된 페트리네트(generalized PN) : 즉각 및 시간지연 혼합 페트리네트
 - 즉각(immediate) 트랜지선(플레이스)
 - 시간지연(timed) 트랜지선
- 확률 페트리네트(stochastic PN) : 점화시간(firing time)이 확률변수인 경우
 - 칼라 페트리네트(colored PN) : 다수의 부품 흐름을 용이하게 표현 가능
 - 확장 페트리네트(Extended PN) : 서브네트 등을 사용하여 표현력 보강

2.3 모델링(modeling)

- 선행관계(precedence of events : sequential execution 순차적인 수행) : 공정이 어떤 특정한 순서에 의하여 진행될 때, 제약조건을 부가하게 된다. 분산되어 있는 여러 자원 사이의 관계를 나타내 주기도 한다.
- 상충(conflict) 또는 경합: 제한된 자원을 동시에 사용 요청하는 경우로, 모두 활성화되어 있으나, 어느 하나를 점화시키면 나머지 모두는 비활성화가 된다. 경합의 조정은 결정적인 방법과 확률적인 방법이 사용된다.
- 동시발생 또는 병렬처리(concurrency or parallelism) : 시스템 요소들의 상호작용 가운데 가장 흔히 발생하는 상황이다.
- 동기(synchronization) : 유연 생산시스템에서 부품들은 특정한 자원을 사용하려고 기다리고 있고, 자원은 어떤 특정한 부품을 기다리는 경우가 있다. 이러한 상황은 동기화를 함으로서 묘사된다.
- 합병(merging) : 여러 부품들이 같은 기계에서 서비스 받을 때, 예를 들어 조립과 같은 공정을 나타내기 위해 필요한 상황이다.
- 혼란(confusion) : 동시발생과 경합이 공존하는 상태이다.
- 우선순위(priorities) : 공정의 순서상, 또는 경합의 조정 등의 조작을 위하여 점화 순서를 임의로 조작하는 경우이다.

3. 페트리네트의 성질

페트리네트에는 두 가지의 성질이 있다. 초기마

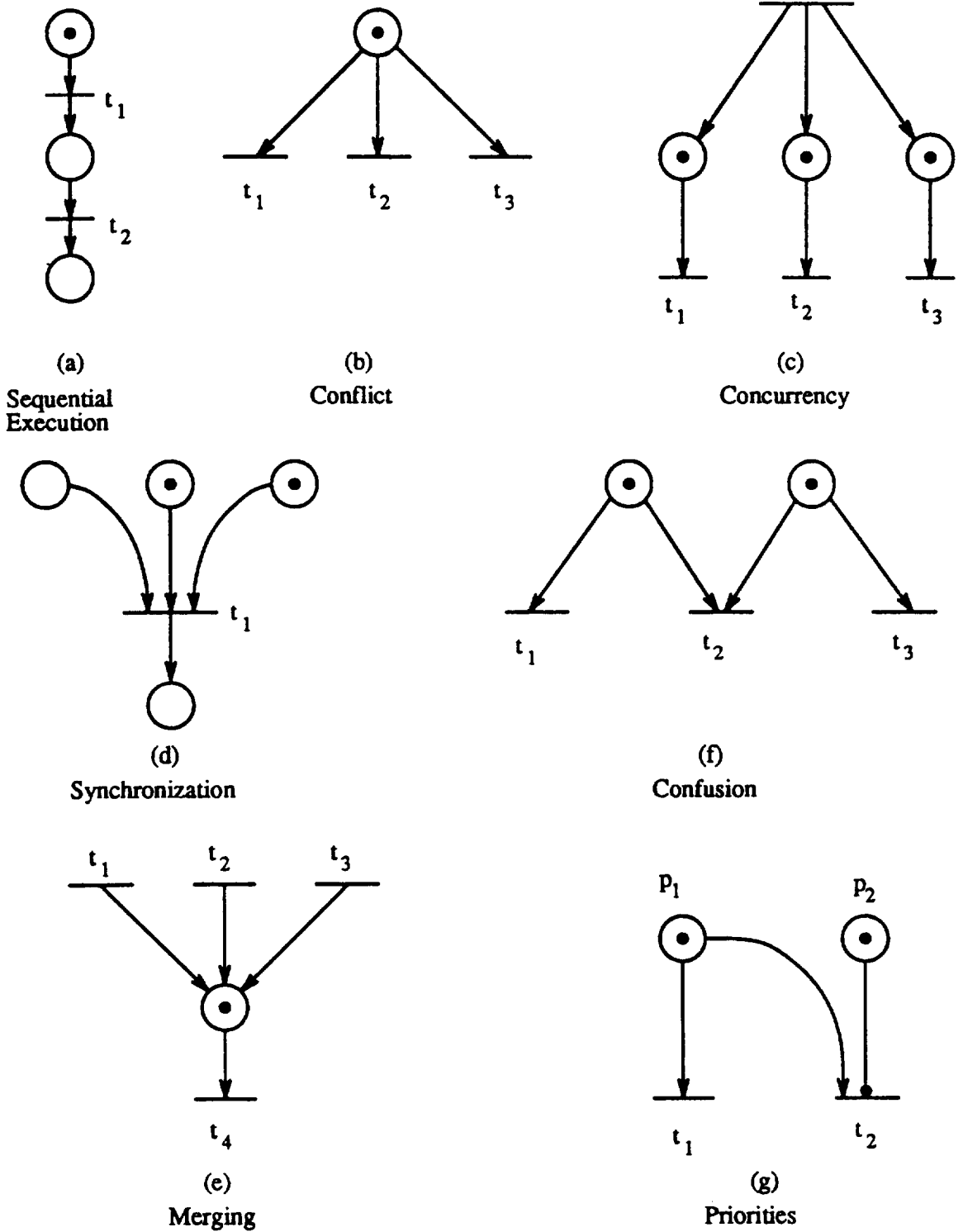


그림 2. 여러 상황에 대한 페트리네트 모델링

킹에 관계된 성질이 있고, 초기마킹에 관계치 않는 성질이 있다. 초기마킹에 의존적인 성질을 행태성질(behavioral properties)이라 하고, 그렇지 않은 성질을 구조적성질(structural properties)이라 한다.

3.1 행태적 성질

1) 도달가능성(reachability) : 도달가능성은 시스템의 동적성질을 연구하는데 근간이 된다. 활성화된 트랜지션의 점화는 점화 규칙에 따라 토큰의 분산을 변화시킨다. 일련의 점화순서는 일련의 마킹을 만든다. 초기마킹 M_0 를 임의의 마킹 M 으로 진전시키는 일련의 점화 순서가 존재할 때 도달가능하다고 말한다.

$R(M_0)$ = reachability set : M_0 에서 도달가능한 모든 마킹의 집합

활성화된 트랜지션 t_j 가 점화되면 상태가 전이되어 새로운 마킹(marking) M' 이 된다.

$$M'(P) = M(P) + O(P, t_j) - I(P, t_j)$$

$$M \xrightarrow{t_j} M'$$

2) 유한 갯수(boundedness) : 모든 플레이스의 토큰의 갯수가 k 개로 제한되어 있을 때 “ k 개로 유한하다(k -bounded)”고 말한다. 만일 모든 플레이스의 토큰 갯수가 1개로 제한되어 있다면, 안전(safe)하다고 한다. 이 성질은 시스템 내에 overflow 상태가 존재하지 않음을 뜻한다.

- 플레이스 p_i 의 토큰 갯수 $M(p_i) \leq k$

$$M(p_i) \leq k \text{ for any } M \in R[M_0]$$

3) 생동성(liveness=deadlock-free PN) : 초기 마킹 M_0 로부터 어떤 마킹 M 에 도달할때 항상 enable한 트랜지션이 존재함을 뜻한다. 교착상태(deadlock)란 시스템의 공정이 더 이상 계속될 수 없는 상태에 머무르는 경우로서, 그 이후로는 점화 가능한 트랜지션은 더 이상 없다. 도달가능 나무(reachability tree)에서 모든 마킹이 1개 이상의 활성화된 트랜지션을 가질 때 생동성이 있다고 한다.

- 모든 마킹 $R(M_0)$ 에 대하여, firable transition 존재

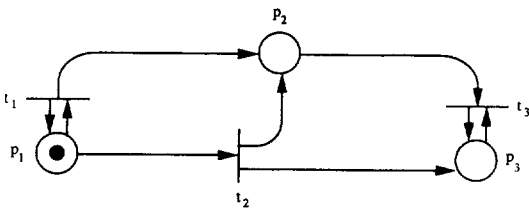
4) 귀환성(reversibility) : 도달가능 나무의 모든 마킹에서 초기마킹으로 도달가능하면 귀환성이 있다고 한다. 그러므로 귀환성이 있는 넷에서는 초기상태로 되돌아 가서 초기화가 될 수 있다. 그러나 어떤 분야에서는 귀환성이 바람직하지 못한 성질일 수도 있다.

- 초기상태 M_0 에 출발하여 $M=M_0$ 가 되는 점화 순서가 존재

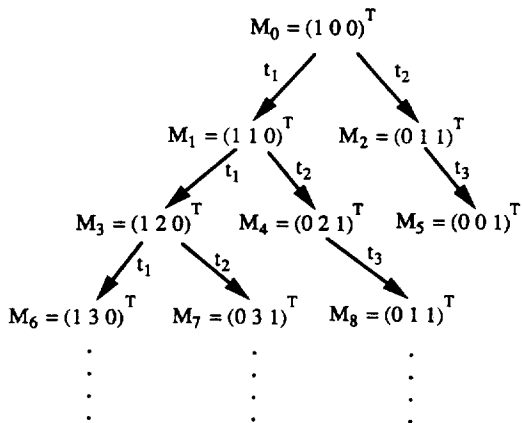
- 고장(failure) 상태에 진입했을때 재초기화 가능(error-recovery)

5) 포함관계(coverability) : 만약 마킹 M 에 대하여 $M'(p) \geq M(p)$ 한 마킹 M' 이 존재할 때 마킹 M 은 포함될 수 있다고 한다. 포함관계는 도달가능성의 특별한 예이다.

6) 일관성(consistency; persistence) : 초기 마킹 M_0 로 돌아갈 수 있다면 이 페트리넷은 일관



(a)



(b)

그림 3. 페트리넷 모델에 대한 도달가능 나무

성이 있다고 한다. 이 때 시스템은 반복적인 작업 (cyclic activity)을 수행할 수 있다.

3.2 구조적 성질

페트리네트의 기하 구조에 관계하는 구조적인 성질들은 초기마킹이나, 점화순서에 관계치 않는다. 그러므로 이러한 성질은 사건발생 행렬과 이에 관련된 방정식과 부등식을 사용하여 특징지을 수 있다. 여기서는 자기 귀환이 없는 페트리네트를 대상으로 한다.

1) 구조적 생동성 : 페트리네트 N이 생동성 있는 초기 마킹이 있을 때, 구조적인 생동성이 있다고 한다.

2) 가제어성 : N의 어떤 마킹이 다른 모든 마킹으로 도달가능하다면, N은 완전 가제어성이 있다.

4. 페트리네트 해석

4.1 점화법칙

예를 들어 그림1에서 $P=\{p_1, p_2, p_3\}$, $T=\{t_1, t_2\}$ 라 하면

$$I(p_1, t_1) = I(p_2, t_1) = I(p_3, t_2) = 1$$

$$I(p_3, t_1) = I(p_1, t_2) = I(p_2, t_2) = 0$$

따라서,

$$I(P, T) = \begin{bmatrix} I(p_1, t_1) & I(p_1, t_2) \\ I(p_2, t_1) & I(p_2, t_2) \\ I(p_3, t_1) & I(p_3, t_2) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

유사하게,

$$O(p_1, t_2) = O(p_2, t_2) = O(p_3, t_1) = 1$$

$$O(p_3, t_2) = O(p_1, t_1) = O(p_2, t_1) = 0$$

따라서,

$$O(P, T) = \begin{bmatrix} O(p_1, t_1) & O(p_1, t_2) \\ O(p_2, t_1) & O(p_2, t_2) \\ O(p_3, t_1) & O(p_3, t_2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$M_0 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

enabling rule $M(P) \geq I(P, t_j)$ 에 의해,

$$M_0 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = I(P, t_1)$$

$$M_0 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \not\geq \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = I(P, t_2)$$

따라서 t_1 은 점화가 가능하며, t_2 는 점화가 불가능하다. t_1 에 대하여 점화법칙을 사용하면

$$M_1 = M_0 + O(P, t_1) - I(P, t_1)$$

$$M_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

새로운 M_1 은

$$M_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = I(P, t_2)$$

가 된다. 따라서, t_2 는 enable하다. 다시, 활성화법칙을 사용하면

$$M_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \not\geq \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = I(P, t_1)$$

가 되어, t_1 은 enable 하지않다. t_2 에 대하여 점화법칙을 사용하면

$$M_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

4.2 사건발생행렬(Incidence Matrix)

사건발생 행렬(incidence Matrix)은 토큰 분포에 대한 정보를 제공하는 역할을 한다.

$$C = C(P, T) = O(P, T) - I(P, T)$$

$$M_k = M_{k-1} + Cu_k, k = 1, 2, 3, \dots$$

이 때, $k-1$ 번째 마킹은 제어벡터 u_k 에 의하여 k 번째 마킹으로 상태전이한다. u_k 는 $k-1$ 번째 마킹에서 k 번째 마킹으로 상태전이함에 있어 점화한 트랜지션에 해당하는 요소만이 1의 값을 갖고 나머지

요소들의 값은 0이다. 도달가능 나무에서 각 노드를 나타내는 마킹은 다음과 같은 회기방정식으로 기술할 수 있다.

$$\begin{aligned} M_1 &= M_0 + Cu_1 \\ M_2 &= M_1 + Cu_2 = M_0 + C(u_1 + u_2) \\ &\dots\dots\dots \\ M_d &= M_0 + Cf_d, \quad d = 1, 2, 3, \dots \end{aligned}$$

제어벡터의 확장개념인 점화횟수 벡터 f_d 는 각 요소가 해당 트랜지션의 점화횟수를 나타내고 있다.

$$\sum_{k=1}^d u_k = f_d$$

예로서 그림1에서

$$C(P, T) = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix}$$

k=1 때

$$u_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

t_1 를 점화하면,

$$M_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

k=2 때

$$u_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

t_2 를 점화하면,

$$M_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

d=2,

$$M_d = M_0 + Cf_2, \quad f_2 = u_1 + u_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

4.3 플레이스 불변집합(P-invariants)

플레이스 불변집합은 다음의 식을 만족하는 해이다.

$$x^T M = x^T M_0, \quad x^T C = 0$$

여기서, $x = [x_i]_{m \times 1}$, $x_i = M(p_i)$.

예로서, 그림 4에서

$$C = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11}]^T$ 라 놓으면, $x^T C = 0$ 에서,

$$\begin{aligned} -x_1 + x_7 + x_8 + x_9 &= 0 \\ x_1 - x_2 - x_8 &= 0 \\ x_2 - x_3 + x_8 &= 0 \\ x_3 - x_7 - x_8 - x_9 &= 0 \\ -x_4 + x_8 + x_{10} + x_{11} &= 0 \\ x_4 - x_5 - x_8 &= 0 \\ x_5 - x_6 + x_8 &= 0 \\ x_6 - x_8 - x_{10} - x_{11} &= 0 \end{aligned}$$

$$\begin{aligned} I_1 &= [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^T \\ I_2 &= [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]^T \\ I_3 &= [1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]^T \\ I_4 &= [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T \\ I_5 &= [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1]^T \end{aligned}$$

I_1 에서, $M(p_1) + M(p_2) + M(p_3) + M(p_7) = 1$ 이므로, 이 플레이스를 포함하는 루프회로는 safe 하다.

4.4 트랜지션 불변집합(T-invariants)

$y = [y_i]_{n \times 1}$, y_i 는 t_i 의 상태라 할 때, $M = M_0$

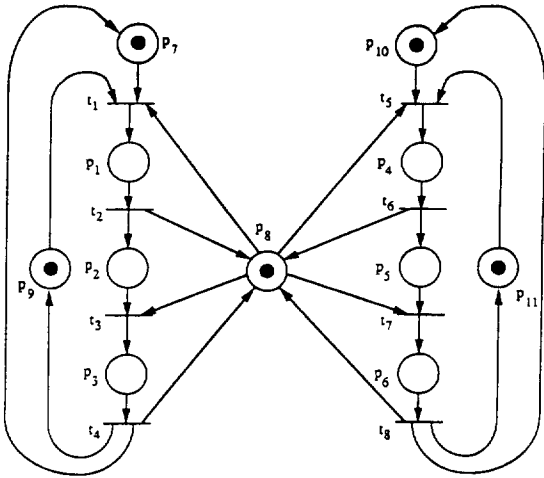


그림 4. 동일한 자원을 이용하는 두 공정에 대한 페트리네트

+Cf 이므로 $Cy = 0$ 이 되는 그러한 y 에 대하여 점화시키면 $M = M_0$ 로 귀환된다.

그림 4에서 $y = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8]^T$ 라 놓으면, $Cy = 0$ 에서

$$\begin{aligned} -y_1 + y_2 &= 0 \\ -y_2 + y_3 &= 0 \\ -y_3 + y_4 &= 0 \\ -y_5 + y_6 &= 0 \\ -y_6 + y_7 &= 0 \\ -y_7 + y_8 &= 0 \end{aligned}$$

$$\begin{aligned} y_1 &= y_2 = y_3 = y_4 \\ y_5 &= y_6 = y_7 = y_8 \end{aligned}$$

$$\begin{aligned} TI_1 &= [1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]^T \\ TI_2 &= [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1]^T \end{aligned}$$

TI_1 으로부터, 이 페트리네트의 트랜지션 1, 2, 3, 4를 점화하면 초기상태로 귀환된다.

5. 단순화 규칙

대규모 페트리네트 시스템의 해석을 용이하게 하기 위해, 우리는 흔히 주어진 모델을 더욱 간단하게 축소 또는 단순화하게 된다. 물론, 이 과정에서 원래의 시스템에 내재한 성질이 변질되어서는 아니된다.

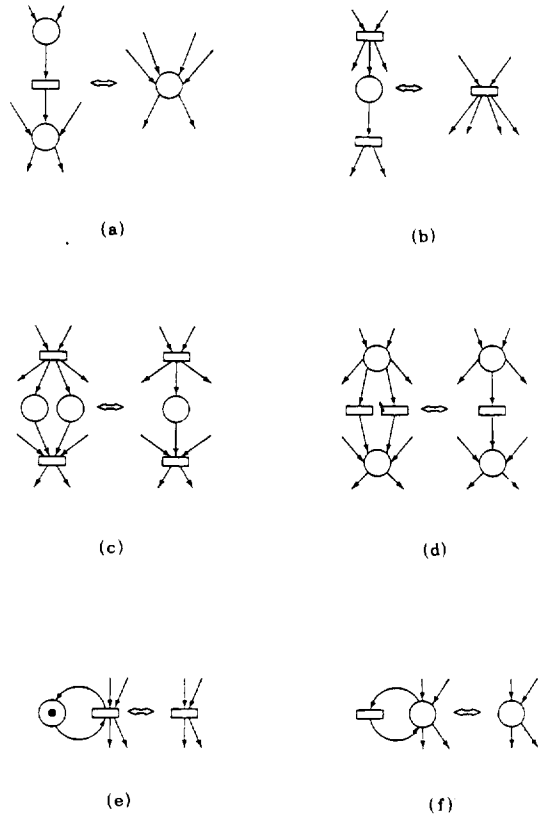


그림 5. 단순화 규칙

- 1) 직렬로 연결된 플레이스의 통합(fusion of series places)
- 2) 직렬로 연결된 트랜지션의 통합(fusion of series transitions)
- 3) 병렬로 연결된 플레이스의 통합(fusion of parallel places)
- 4) 병렬로 연결된 트랜지션의 통합(fusion of parallel transitions)
- 5) 자기 귀환을 하는 플레이스의 제거(elimination of self loop places)
- 6) 자기 귀환을 하는 트랜지션의 제거(elimination of self loop transitions)

6. 결론

지금까지 페트리네트에 대한 정의, 모델, 성질 및 해석을 살펴보았다. 실제적인 상황에서 일반적인

도구로서 사용하기 위하여 연구가 진전되고 있으며, 페트리네트는 이산사건을 해석하고 응용연구하는데 뛰어난 진가를 발휘할 것으로 기대된다.

참 고 문 헌

- [1] Marco Ajmone Marsan, "Application and Theory of Petri Nets 1993", Springer-verlag 1993.
- [2] N. Viswanadham and Y. Narahari, "Performance Modeling of Automated Manufacturing Systems", Prentice Hall, 1992.
- [3] Todao Murata, "Petri Nets: Properties, Analysis and Applications", Proceedings of the IEEE, Vol. 77, No. 4, April 1989.



남부희(南副熙)

1950년 12월 2일생. 1974년 서울대 전기공학과 졸업. 1983년 Iowa State Univ. 전기공학과 박사. 1976~1978년 울산공대 근무, 1978~1980년 충남대 근무. 1983년~현재 강원대 제어계측공학과 교수. 당학회 평의원. 1984년 IEEE Circuits and Systems Society 논문상.