

# 자원 공유 플레이스 추이적 행렬을 이용한 효율적인 교착상태 확인 정책

김종욱<sup>1</sup> · 이종근<sup>1†</sup>

## An Efficient Deadlock Detection Policy Using the Transitive Matrix of Resource Share Places

Jongwoog Kim · Jongkun Lee

### ABSTRACT

A deadlock is a condition in which the excessive demand for the resources being used by others causes activities to stop and it is one of the important problems in process control system to detect and prevent deadlocks. While the reachability analyze methods or divide subnets for analyzing methods are well used, it requires a lot of times and effects to analyze the detection of the deadlock status. Furthermore, it could not be effective if the model is complex or huge. Therefore, it is necessary to develop a new and more efficient deadlock detection algorithm. In this paper, a deadlock detection conditions after analyzed the Petri Net using the transitive matrix has been proposed to solve these problems. For presenting the results, the suggested deadlock detection algorithm was also adapted to an illustrated FMS (Flexible Manufacturing System) model.

**Key words** : Algorithm, Conflict, Deadlock, Petri-nets, Resource share place, Transitive matrix

### 요 약

여러 개의 작업이 동시에 작동 할 때 서로 다른 작업에서 공유자원을 사용하기 위하여 상대방의 작업이 끝나기를 기다리는 상태를 교착상태라 하며 이는 시스템의 효율성과 경제성 제고에 가장 중요한 문제 중 하나이다. 지금까지 이러한 교착상태 확인은 도달성(reachability)기법이나 서브 넷(Subnet)으로 분리하여 분석하는 기법 등이 가장 많이 사용 되었으나, 분석에 필요한 시간과 노력, 복잡성과 효율성에 있어서 효과적이지 못한 단점을 가지고 있다. 본 연구는 패트리 넷(Petri Net)에서 모든 플레이스(Place) 간의 관계를 나타내는 추이적(Transitive)행렬을 이용하여 교착상태가 나타날 가능성이 있는 자원공유 플레이스를 분석하여 교착상태를 검증하는 정책을 제시한다. 이를 위하여 교착상태를 확인할 수 있는 자원공유 플레이스의 관계를 정의하고, 유연생산시스템 모델을 이용하여 제안한 알고리즘의 성과를 검증한다.

**주요어** : 알고리즘, 병합, 교착, 유연생산시스템, 패트리 넷, 자원공유 플레이스, 추이적 행렬

## 1. 서 론

교착상태란 시스템의 공정 중에 포함 된 여러 작업에서 공동으로 사용되는 공유 자원을 각 작업에서 상호적으로 사용을 기다려 시스템의 작업 공정이 중단 된 상태를 의미하며, 현대의 많은 기술 시스템인 자동생산시스템, 데

이터 통신, 다중처리 운영시스템과 분산 데이터 베이스 시스템 등에서 가장 잘 알려진 문제이다. 다양한 시스템에서 교착상태 발생은 시스템 관리와 작업과정 등에서 효율적인 처리 과정을 중단시키거나 무력화 시키며 특히 자원 공유 체제를 붕괴 시키는 중대한 결함을 야기시킨다<sup>[17]</sup>. 이러한 교착 문제를 해결하기 위한 교착상태 확인 및 회피 방법에 대하여 많은 연구가 제안 되어 왔는데, 대표적인 분석기법으로 패트리 넷(Petri Net)을 활용하는 연구가<sup>[1-3,5-7,9-13,15,17,19,22-25]</sup> 활발하다. 패트리 넷은 비동기적이며, 동시 발생적인 이벤트에 의해 시스템의 상태가 변화하는 이산사건 시스템(DES: Discrete Event System)을 모델링 하는데 아주 적절한 도구이며, 분산 시스템 환경에서, 공

2008년 7월 22일 접수, 2008년 9월 10일 채택

<sup>1)</sup> 창원대학교 컴퓨터공학과

주 저 자 : 김종욱

교신저자 : 이종근

E-mail: jklee@changwon.ac.kr

유된 자원간의 상호 배제와 같은 특징들을 모델링 하는데 활용되고 있다<sup>[3,14,19,24]</sup>. 패트리 넷은 상태를 나타내는 플레이스(place)(○로 표시)와 변이를 나타내는 트랜지션(transition)(■로 표시) 그리고 변화의 흐름을 표현하는 토큰(token)(●로 표시)으로 구성 되어 있는데, 패트리 넷의 마킹(marking)은 토큰의 흐름을 나타내는 개념으로 모델링 된 시스템의 작업 흐름 상태를 분석 할 수 있다. 마킹은 트랜지션의 점화를 통해 변화하므로 트랜지션의 점화 순서를 분석하면 원하는 시스템의 상태 분석과 모델링의 검증이 가능하다. 모델링에서 중요한 것은 마킹의 흐름이 원활하게 순환되는 것인데, 플레이스에서 서로가 변이 대기상태가 되어 마킹의 흐름이 정지 될 경우 이를 교착상태라 한다. 이는 패트리 넷의 성질 분석하는데 아주 중요한 요소 중의 하나이며, 특히 이산사건 시스템에서의 교착상태 확인과 회피에 아주 주요한 성질로 이용되고 있다. 교착상태를 확인하는 패트리 넷 이용 모델에서는 크게 siphon 이용 분석과<sup>[5,8,11,19,22,23]</sup> 도달성 그래프(reachability graph) 분석방법이<sup>[10,15,17,21,25]</sup> 많이 사용되고 있다. siphon이란 입력 플레이스에 의하여 삼입 된 토큰이 다시 입력 플레이스로 주입되는 특성을 가진 서브 넷 개념으로 이러한 과정을 통하여 토큰의 수가 줄어들어 교착상태의 발생 여부를 확인 할 수 있는 알고리즘이다. 특히 siphon 분석 기법은 초기 마킹과 연관 된 아크(arc)를 추가하여 독립적인 서브 넷으로 나누고 이를 분석하고 교착상태를 제거하고 다시 서브 넷들을 통합하는 기법으로 제시 되었는데, 초기모드에서 siphon 특성을 분석하고 siphon 서브 넷을 찾기 위한 노력이 절대적으로 필요하다. 도달성 그래프 기법은 가장 손쉽게 활용 되는 기법으로 마킹의 흐름을 그래프로 나타내거나 수식으로 표기하여 교착상태를 확인하는 가장 고전적인 기법이다. Unfolding 기법을 이용하여 도달성 트리를 축소화하여 도달성 트리에 나타날 수 있는 교착상태를 분석하는 연구가 [17]에서 제안되었으며, 인접행렬을 이용한 알고리즘으로 DAPN<sup>[25]</sup>이 있는데, 이는 패트리 넷 모델을 인접 행렬로 표시하고 각 작업 별로 행렬을 분리하여 토큰의 흐름으로 교착상태를 확인하는 기법이나 알고리즘이 복잡한 단점을 가진다. 또한 추이적 행렬로 모델링 한 후 마킹의 흐름을 도달성 트리로 분석하여 교착상태를 확인 및 회피하는 알고리즘을 [10, 15]에서는 제안하였다. 그러나 초기 모델이 크거나 복잡한 경우 도달성 그래프기<sup>[25]</sup> 거대하여 지고 복잡하여져 분석에 많은 시간과 노력이 필요하게 된다. 본 연구에서는 패트리 넷에서의 플레이스와 트랜지션간의 관계를 표현하는 추이적 행렬<sup>[14,15,16]</sup>을 이용하여 자원공유 플레이스간

의 산술적 해석으로 교착상태를 확인하는 알고리즘을 제안한다. 추이적 행렬이란, 패트리 넷의 플레이스와 트랜지션간의 관계를 모두 표현함으로 토큰의 흐름을 행렬을 통하여 분석이 가능하며, 모든 형태의 플레이스 확인이 가능하고, 또한 흐름의 제반 내용을 표를 통하여 분석 가능한 특징을 가지고 있다. 따라서, 교착 가능한 플레이스는 자원공유 플레이스임에 착안하여 공유 플레이스간의 관계를 수리적으로 분석하여 교착상태 발생 가능한 위치를 확인함으로 시스템의 성질 분석 및 검증이 용이하도록 알고리즘을 제안한다.

패트리 넷의 정의와 추이적 행렬의 요약은 2장에서 정리하고, 교착상태를 찾기 위한 수리적 해석 방법이 예제와 함께 3장에 기술된다. 4장에서는 제안 알고리즘의 실용성을 확인하기 위하여 유용생산시스템(FMS: Flexible Manufacturing System)의 사례 모델에 적용하여 그 성능을 검증한다.

## 2. 패트리 넷

패트리 넷은 일반적으로 시스템의 모델링과 분석에 많이 사용되며, 다음과 같이 정의 할 수 있다<sup>[4,9,14,16,18,20]</sup>.

패트리 넷 PN은 5-튜플로 구성된다:

$$PN = (P, T, I, O, Mo)$$

$P = \{p_1, p_2, \dots, p_n\}$ : 플레이스의 유한 집합,

$T = \{t_1, t_2, \dots, t_k\}$ : 트랜지션의 유한 집합,

$I(t_j \in T \rightarrow I(t_j) \in P)$ : 트랜지션의 입력 함수,

$O(t_j \in T \rightarrow O(t_j) \in P)$ : 트랜지션의 출력 함수,

$Mo: P \rightarrow \{0, 1, 2, 3, \dots\}$ : 초기 마킹  $P \cap T = \emptyset$ .

트랜지션  $t \in T$  는 M에서 점화가능하며 이를  $M[t >$  로 표현한다, 만일  $\forall p \in \bullet t: M(p) > 0$  가 된다, 여기서  $\bullet t$  는 트랜지션 t의 입력 플레이스,  $t \bullet$ 는 트랜지션 t의 출력 플레이스를 의미한다. 만일  $M[t >$  이면 트랜지션 t는 점화 가능하며, 새로운 마킹  $M'$ 을 가지며,  $M[t > M'$ , with  $M'(p) = M(p) - 1$ 로 표시된다, 만일  $p \in \bullet t$ ;  $M'(p) = M(p) + 1$  만일  $p \in t \bullet$ ; 그렇지 않으면  $M(p) = M'(p)$ ,  $\forall p \in P$ 에서.

초기 마킹  $Mo$ 에서 도달 가능한 마킹의 집합은  $R(PN, Mo)$ 로 표시된다.

$(PN, Mo)$ 는 5-튜플  $PN = \langle P, T, I, O, M \rangle$ 를 갖는 패트리 넷이라고 하자.

트랜지션  $t \in T$ 는  $\forall M \in R(PN, Mo)$ ,  $\exists M' \in R(PN, M)$ ,  $M'[t >$  이면 마킹  $Mo$ 로부터 생존(live) 가능하다. 만

일  $\exists t \in T, \text{Mo}[t >$  이면 PN는 교착상태이다. 만일  $\forall M \in R(\text{PN}, \text{Mo}), \exists t \in T, M[t >$ 이면  $(\text{PN}, \text{Mo})$ 는 교착자유상태(deadlock free)이다. 만일  $\forall t \in T, \exists M \in R(\text{PN}, \text{Mo})$ 이며  $M[t >$  가 성립하면  $(\text{PN}, \text{Mo})$ 는 부분생존한다. 만일  $\forall t \in T$ , 트랜지션  $t$ 가  $\text{Mo}(\text{PN}, \text{Mo})$ 에서 생존하면  $(\text{PN}, \text{Mo})$ 은 생존한다. 만일  $\exists k \in \mathbb{N}, \forall M \in R(\text{PN}, \text{Mo}), \forall p \in P, M(p) \leq k$ 를 만족하면  $(\text{PN}, \text{Mo})$ 는 유한하다.

PN 구조의 행렬  $C$ 는  $C = \langle P, T, B, B^+ \rangle$ 이라 하면, 여기서  $P = \{p_1, p_2, \dots, p_n\}$ 는 플레이스의 유한집합,  $T = \{t_1, t_2, \dots, t_k\}$ 는 트랜지션의 유한집합,  $B$  와  $B^+$  는  $n$ 열과  $k$ 행의 행렬로 다음과 같이 정의된다.

$B^- [i, j] = \#(P_i, I(t_j))$ , 입력함수 행렬,

$B^+ [i, j] = \#(P_i, O(t_j))$ , 출력함수행렬.

또한,  $B = B^+ - B^-$  는 유사(incidence)행렬이라 한다.

### 2.1 추이적 행렬

추이적 행렬에 대한 일반 정의를 정리해 보면 다음과 같다<sup>[9,10,14-16]</sup>.

동차방정식  $B^T \cdot x = 0$ 에서 행 벡터(vector)  $x^T = (x_1, x_2, \dots, x_k) \geq 0$  ( $\Delta M = 0$ )는 T-상태불변(T-invariant)이라 하며, 여기서  $x^T$ 는  $x$ 의 변형이다. 하나의 정수 값  $y = (y_1, y_2, \dots, y_n)^T$ 가 변형동차방정식에서 존재한다면  $B \cdot y = 0$ 는 S-상태불변(S-invariant)라 한다.

이 식을 활용하여 플레이스 추이적 행렬을 다음과 같이 얻을 수 있다.

$B = B(B^+)^T$  : 플레이스 추이적 행렬

이제  $L_{BP}$ 를 추이적 행렬이라고 하면:

$$L_{BP} = B^- \text{diag}(t_1, t_2, \dots, t_k) (B^+)^T,$$

여기서  $t_i (i = 1, 2, \dots, k)$ 는:

$$|t_i| = \begin{cases} 1 & \text{fire } t_i \\ 0 & \text{not fire } t_i \end{cases} \text{ 이다,}$$

여기서  $|t_i|$ 는 행렬에서 트랜지션  $t_i$ 가 갖는 값을 의미한다. 즉,  $t_i$ 가 점화하면 1이며 점화하지 않으면 0으로 표시된다.

$L_{BP}$ 는 모든 플레이스간의 관계를 각각의 트랜지션을 이용하여 직접적으로 표현 된다.

$L^*_{BP}$ 를 라벨추이적 행렬이라 할 때, 만일 어떤 트랜지션  $t_k$ 가 같은 행에서  $s$ 번 나타나게 된다면 우리는  $t_k/s$ 로 표시한다.

도달 마킹  $M_R(K+1)$ 은 초기마킹  $K(k)$ 으로부터 도달되어지는 마킹으로 정수형  $m$ -벡터라 한다. 도달 마킹의

표현 형태는 다음과 같다.

$$M_R(k+1)^T = M(k)^T M_{PR}$$

여기에서, 만일  $\sum_i p_i M_{PR}$  이 정수(혹은 음수) 값이면,  $\sum_i p_i M_{PR} = 1$ (혹은 0)로 한다. 이 의미는  $P_i(k)$ 의 토큰은 마킹  $M(k)$ 에서 도달 마킹  $M_R(K+1)$ 으로 전달이 된다는 의미를 갖는다(정수이면 전달 가능, 음수이면 전달 불가). 도달 마킹  $M_R(K+1)$ 은 단지 토큰의 흐름을 나타내는 관계식이므로, 도달 마킹  $M_R(K+1)$ 은  $M(k+1)$ 에 꼭 기인 할 필요는 없다는 의미도 갖는다. 모델을 간단히 하기 위하여 또한 모델의 효율성을 위하여,  $|P_i(k)|=1$  혹은 0으로 제한한다.

추이적 행렬에서 열 벡터는 플레이스  $p_i$ 에게 행 벡터의 플레이스에서 토큰이 입력되는 관계를 나타낸 것이며 행 벡터는 플레이스  $p_i$ 에서 열 벡터의 플레이스로 토큰이 출력되는 관계를 나타내는 것이다. 지금의 추이적 행렬은 기본 패트리 넷으로 나타낸 모델에만 적용되므로 모든 토큰과 아크의 기본 개수는 1이며 복수개의 토큰이나 아크는 고려하지 않는다.

추이적 행렬의 행 방향은 제어흐름을 표현한다. 따라서  $\sum(t_k/s_i)$ 의 값에 따라 제어흐름을 알 수 있다.

만약  $\#(p_i, O(t_k)) < \#(p_i, E(t_k))$ 이면 해당 플레이스의 병행적인 흐름이 유지되거나 확장되지 않고, 축소됨을 뜻한다.

이를 유형별로 정리하면 다음과 같다.

각각의  $\sum(t_k/s_i)$ 에 대하여,

- (1)  $\#(p_i, O(t_k)) = \#(p_i, E(t_k))$ 인 경우 : 토큰  $t_k$ 의 점화로 인한 제어 영역이 그대로 유지된다.
- (2)  $\#(p_i, O(t_k)) < \#(p_i, E(t_k))$  인 경우  $\sum(t_k/s_i) > 1$  : 토큰  $t_k$ 의 점화로 인한 제어 영역이 넓어지게 된다.
- (3)  $\#(p_i, O(t_k)) < \#(p_i, E(t_k))$  인 경우  $\sum(t_k/s_i) < 1$  : 토큰  $t_k$ 의 점화로 인한 제어 영역이 좁아지게 된다.

추이적 행렬의 열방향은 각 트랜지션의 점화 가능성을 표현한다. 또한, 열방향의 자원 공유 플레이스에는 토큰이 존재한다.

그리고 자원 공유 플레이스의 열방향에 표현된 트랜지션의 수는 그 플레이스로 입력되는 트랜지션의 수를 말하므로  $\sum(t_k/s_i)$ 의 값에 따라 해당 트랜지션의 점화 여부를 알 수 있는데,  $\sum(t_k/s_i)$ 의 값이 1이면 점화가 가능하다. 그 이유는 추이적 행렬은 기본 패트리 넷(Ordinary Petri Nets)을 기본으로 하므로 모든 플레이스의 토큰은 항상 1이어야 하기 때문이다.

### 2.2 교착상태

교착상태란 시스템의 흐름에서 서로가 대기상태가 되어 마킹의 흐름이 정지 될 경우 이를 교착상태라 한다. 또한 이러한 상태는 주로 자원공유 플레이스에서 많이 발생되는 성질이 있다. 즉 자원공유 플레이스에서 입출력 토큰의 수가 음수이거나 정수가 되지 못하면 자력으로 접화가 불가하므로 교착상태 가능성이 있다.

#### 성질1:교착 노드

- 1) 마킹 M으로부터 어떠한 트랜지션으로 점화가능하지 못하다면, 그 트랜지션은 교착 트랜지션이라 한다.
- 2) 마킹 M으로부터 어떠한 플레이스로 마킹이 전이가 되지 못하면 그러한 플레이스를 교착 플레이스라 한다.

#### 성질2:교착상태

패트리 넷에서 교착 노드가 하나 이상 나타나면 마킹 M은 교착상태라 한다.

**성질3:**Rc(p<sub>i</sub>)를 행의 플레이스 p<sub>i</sub>의 토큰 수의 합이고, Rr(p<sub>i</sub>)를 열 플레이스 p<sub>i</sub>의 토큰 수의 합이라고 하면:

$$Rc(p_i) = \sum_{i=1}^n (p_i) = \sum_{i=1}^n (f \bullet p_i),$$

$$Rr(p_i) = \sum_{i=1}^n (p_i) = \sum_{i=1}^n (f \bullet p_i),$$

여기서 n은 플레이스의 수,

f는 L\*<sub>BP</sub>의 함수

만일 ∃ t 이면 f = 1 아니면 f = 0.

그러면, Dr(p<sub>i</sub>) = Rr(p<sub>i</sub>) - Rc(p<sub>i</sub>)은 플레이스 P<sub>i</sub>의 교착상태 조건이라 한다.

여기서, Dr(p<sub>i</sub>) = k, 만일 k ≥ 0 이고 k가 정수이면 이 PN은 교착자유상태이고 그렇지 않으면 교착상태이다.

증명: 추이적 행렬은 기본 패트리 넷을 기초로 작성 되는(입력과 출력이 각각 1이며 플레이스의 토큰도 1이어야 한다) 플레이스 행렬이므로 모든 행과 열의 합이 정수이어야 하며, 음수가 되면 점화가 불가하게 된다. 따라서, 정수이며 음수가 아닌 토큰의 합이어야 교착자유상태이다.

### 3. 추이적 행렬을 이용한 교착상태

#### 3.1 교착상태 확인

교착상태는 자원공유 플레이스에서 발생하게 된다. 그림 1

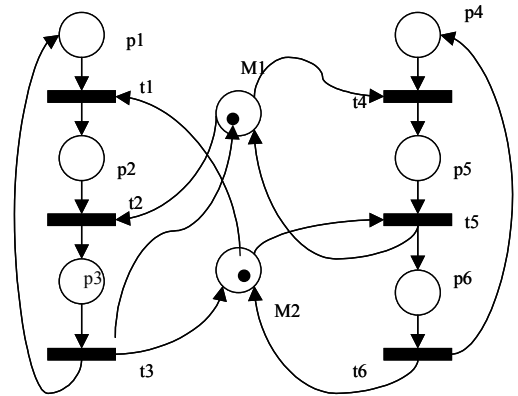


그림 1. 예제 패트리 넷

표 1. 그림 1의 추이적 행렬

	P1	p2	p3	p4	p5	p6	M1	M2	
L* <sub>BP</sub> =	0	t1/2	0	0	0	0	0	0	P1
	0	0	t2/2	0	0	0	0	0	P2
	t3	0	0	0	0	0	t3	t3	P3
	0	0	0	0	t4/2	0	0	0	P4
	0	0	0	0	0	t5/2	t5/2	0	P5
	0	0	0	t6	0	0	0	t6	P6
	0	0	t2/2	0	t4/2	0	0	0	M1
	0	t1/2	0	0	0	t5/2	t5/2	0	M2

과 같은 예처럼, 두 개의 트랜지션 t2와 t5는 플레이스 M1과 M2에서 교착상태가 발생된다. 즉 자원 공유플레이스 M1과 M2가 각각 트랜지션 t1과 t4에서 점화를 하면 트랜지션 t2와 t5에서 M1과 M2의 토큰을 기다리는 교착상태가 발생하게 된다. 그러나, M1이 t4에서 점화를 하지 않고 t1이 점화 완료 될 때까지 기다린다면 t2는 점화가 가능하다. FMS에서는 동시처리를 원칙으로 고려하므로 대기시간의 발생도 교착상태 발생으로 본다.

그림 1에서 자원공유 플레이스는 M1과 M2이므로 각 공유 플레이스의 행과 열의 토큰의 합을 각각 Rc(p<sub>i</sub>)과 Rr(p<sub>i</sub>)로 정의하면:

- 1) 자원공유 플레이스 M1:

$$Rr(M1) : 1+1/2+1/2=2,$$

$$Rc(M1) : 1/2+1/2=1 \text{ 이며}$$

교착상태 판별조건 Dr(p<sub>i</sub>)는

$$Dr(M1) = 2-1=1$$

따라서 M1은 교착상태 플레이스가 아님.

2) 자원공유 플레이스 M2:

$$Rr(M2) : 1+1=2,$$

$$Rc(M2) : 1/2+1/2 =1(1/2)$$

$$Dr(M2) = 2-1(1/2)=1/2$$

판별식이 정수가 아니므로 공유 플레이스 M2는 교착 플레이스가 되고 (그림 1) 낮은 교착상태가 된다.

이제 도달함수를 이용하여 교착상태를 확인하여 보면:

$$M_R(k+1)^T = M(K)^T M_{PR}$$

$$M_R(k+1)^T = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1) \begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ t3 & 0 & 0 & 0 & 0 & 0 & 0 & t3 \\ 0 & 0 & 0 & 0 & t4/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t5/2 & t5/2 & 0 \\ 0 & 0 & 0 & t6 & 0 & 0 & 0 & t6 \\ 0 & 0 & t2/2 & 0 & t4/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & t5/2 & t5/2 & 0 \end{bmatrix}$$

$$= (0 \ 1 \ 1/2 \ 0 \ 1 \ 1/2 \ 0)$$

즉, 다음 마킹의 값들이 모두 1이하가 되어 부족한 마킹의 값을 기다리는 교착상태가 되어 점화가 불가능하게 된다.

일반적으로 널리 알려진 또 하나의 예제를 살펴보면<sup>[18]</sup> (그림 2(a)), 플레이스 p3과 p4가 자원공유 플레이스가 되며, 이 예제의 추이적 행렬은 그림 2(b)와 같다.

앞의 예제(그림 3)와 같이 플레이스 p3과 p4간의 관계식을 살펴보면:

1) 플레이스 p3:

$$Rr(P3) : 1/2+1=1(1/2),$$

$$Rc(P3) : 1$$

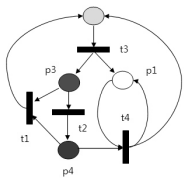
$$Dr(P3) = 1(1/2)-1=1/2$$

2) 플레이스 p4:

$$Rr(P4) : 1/2+1/2+1/2=1(1/2),$$

$$Rc(P4) : 1$$

$$Dr(P4) = 1(1/2)-1=1/2$$



(a) 패트리 넷

$$L_{BP}^* = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & \Sigma \\ t4/2 & t4/2 & 0 & 0 & p_1^1 \\ t3 & 0 & t3 & 0 & p_2^2 \\ 0 & t1/2 & 0 & t2 & p_3^{1(1/2)} \\ t4/2 & t1/2, t4/2 & 0 & 0 & p_4^{1(1/2)} \\ \Sigma & 2 & 2 & 1 & 1 \end{bmatrix}$$

(b) 추이적 행렬

그림 2. FCS 구조

플레이스 p3과 p4의 판별식이 모두 정수가 아니므로, 플레이스 p3과 p4는 각각 교착상태의 플레이스가 되며 이 예제는 교착상태가 된다. 이제, 마킹의 흐름을 이용하여 검증을 하여 보면 다음의 마킹 흐름과 같이 플레이스 p1의 토큰이 부족하게 되어 교착이 됨을 알 수 있다.

$$M_R(k+1)^T = (0 \ 1 \ 0 \ 0) \begin{bmatrix} t4/2 & t4/2 & 0 & 0 \\ t3 & 0 & t3 & 0 \\ 0 & t1/2 & 0 & t2 \\ t4/2 & t1/2, t4/2 & 0 & 0 \end{bmatrix}$$

$$= (1 \ 0 \ 1 \ 0) \begin{bmatrix} t4/2 & t4/2 & 0 & 0 \\ t3 & 0 & t3 & 0 \\ 0 & t1/2 & 0 & t2 \\ t4/2 & t1/2, t4/2 & 0 & 0 \end{bmatrix}$$

$$= (1/2 \ 1 \ 0 \ 1)$$

### 3.2 교착자유상태

이 절에서는 간단한 교착자유상태의 예를 살펴본다. 5개의 플레이스 중에 p3가 자원공유 플레이스이며 작업의 흐름은 2개가 되며(그림 3), 이 모델의 추이적 행렬은 표 2가 된다.

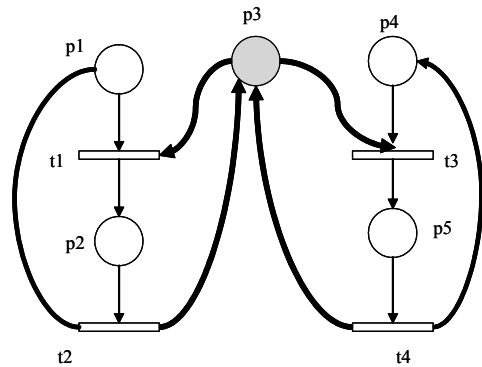


그림 3. 교착자유상태 모델

표 2. (그림 3.3)의  $L_{BP}^*$

	P1	p2	p3	p4	p5	
	0	t1/2	0	0	0	P1
	t2	0	t2	0	0	P2
$L_{BP}^*$	0	t1/2	0	0	t3/2	P3
	0	0	0	0	t3/2	P4
	0	0	t4	t4	0	P5

표 2를 이용하여 자원공유 플레이스 p3를 정리하면:

$$Rr(p3) = 1+1=2,$$

$$Rc(p3) = 1+1=2$$

$$Dr(p3) = 2-2=0$$

판별식이 0이 되어 토큰의 흐름에 토큰 수의 변동이 없음을 알 수 있다. 따라서, 플레이스 p3는 교착자유상태이며, 마킹의 흐름을 통하여 검증을 하여 보면:

$$M_R(k+1)^T = (1\ 0\ 1\ 1\ 0) \begin{bmatrix} 0 & t1/2 & 0 & 0 & 0 \\ t2 & 0 & t2 & 0 & 0 \\ 0 & t1/2 & 0 & 0 & t3/2 \\ 0 & 0 & 0 & 0 & t3/2 \\ 0 & 0 & t4 & t4 & 0 \end{bmatrix}$$

$$= (0\ 1\ 0\ 0\ 1) \begin{bmatrix} 0 & t1/2 & 0 & 0 & 0 \\ t2 & 0 & t2 & 0 & 0 \\ 0 & t1/2 & 0 & 0 & t3/2 \\ 0 & 0 & 0 & 0 & t3/2 \\ 0 & 0 & t4 & t4 & 0 \end{bmatrix}$$

$$= (1\ 0\ 2\ 1\ 0)$$

위의 마킹 흐름과 같이 모든 플레이어의 점화가 가능하며 특히 플레이스 p3는 계속해서 토큰의 수가 증가하게 되어 제한적이지 않은 성질을 갖게 됨을 확인 할 수가 있다.

### 3.3 교착상태 감지 알고리즘

앞 절서 보인 교착상태 감지 알고리즘을 정리하면 다음과 같다:

Algorithm: detection

Input: PN = <P,T,I,O,M>

Output: PN is deadlock free or not

- (1) Define  $L^*_{BP}$  of an initial Petri net PN.
- (2) Find all resource share places in each column  $L^*_{BP}$
- (3) Find all resource share places in each row  $L^*_{BP}$ .
- (4) Calculate  $Dr(pi)$  : deadlock condition using the  $L^*_{BP}$
- (5) Repeat (2)-(4) for all resource share places.
- (6) If  $\sum_{i=1}^n Dr(p_i) = k$ , if  $k \geq 0$  and  $k$  is integer value then this net is deadlock free, if not this PN is deadlock.

## 4. 사례분석

### 4.1 예제모델

앞 장에서 제안한 알고리즘의 실용성을 확인하기 위하여 이 장에서는 교착 연구가 활발한 분야인 FMS의 사례 모델에<sup>[8]</sup> 적용하여 그 성능을 검증한다(그림 4).

FMS의 사례 모델은 2개의 기기 M1과 M2가 존재하며, 한 개의 로봇과 두 개의 이동 포트가 있다. 또한 두 개의 작업 Job 1과 Job 2가 존재하는데 Job1은 기기M1→M2의 순으로 Job2는 기기M2→M1의 순으로 각각 작업에 사용된다. 이러한 작업의 내용을 정리하면 다음과 같다.

Job 1: {M1, M2}

Job 2: {M2,M1}

또한 사례 모델의 패트리 넷 모델은 다음과 같으며(그림 4) 추이적 행렬은 표 3과 같다:

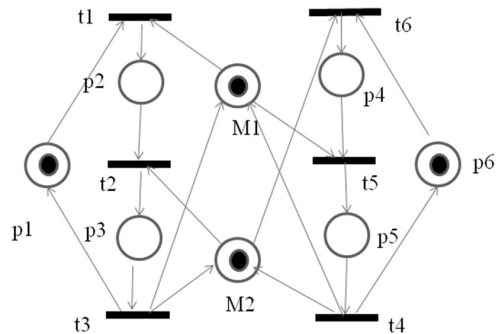


그림 4. 사례모델

표 3. 그림 4의  $L^*_{BP}$

$$L^*_{BP} = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

사례 모델에서의 자원공유 플레이스 M1과 M2를 각각 정리하면 다음과 같다.

1) 플레이스 M1:

$$Rr(M1) = 1+1=2,$$

$$Rc(M1) = 1/2+1/2+1/2=1(1/2)$$

$$Dr(M1) = 2-1(1/2)=1/2$$

2) 플레이스 M2:

$$Rr(M2) = 1+1=2$$

$$Rc(M2) = 1/2+1/2=1$$

$$Dr(M2) 2-1=1$$

여기서 플레이스 M1은 M2와 달리 판별식이 정수가 아니므로 M1은 교착상태가 되어 사례 모델은 교착상태이다.

이제 마킹의 흐름을 통한 검증을 하여 보면 플레이스 p2와 p4의 경우 정상적으로 토큰을 가지게 되나, p3, p5, p6의 경우 비정상적이 되어 교착상태가 발생하게 된다.

$$M_R(k+1)^T = (1 \ 0 \ 1 \ 1 \ 0) \begin{bmatrix} 0 & t1/2 & 0 & 0 & 0 \\ t2 & 0 & t2 & 0 & 0 \\ 0 & t1/2 & 0 & 0 & t3/2 \\ 0 & 0 & 0 & 0 & t3/2 \\ 0 & 0 & t4 & t4 & 0 \end{bmatrix}$$

$$= (0 \ 1 \ 0 \ 0 \ 1) \begin{bmatrix} 0 & t1/2 & 0 & 0 & 0 \\ t2 & 0 & t2 & 0 & 0 \\ 0 & t1/2 & 0 & 0 & t3/2 \\ 0 & 0 & 0 & 0 & t3/2 \\ 0 & 0 & t4 & t4 & 0 \end{bmatrix}$$

$$= (1 \ 0 \ 2 \ 1 \ 0)$$

#### 4.2 제안 알고리즘의 특성

본 연구에서 제안한 추이적 행렬을 이용한 교착상태 확인 알고리즘은 행렬 자체에서 교착여부 확인이 가능한 장점을 가진다. 즉 기존의 siphon이나 도달성 트리의 경우 패트리 넷에서 마킹의 흐름을 전부 확인하여야 교착 여부 확인이 가능한데, 제안한 알고리즘은 공유 플레이스를 기준으로 수리적인 판별식으로 분석하는 관계로 모든 마킹의 흐름을 분석하는 노력과 시간의 절약이 가능하다. 기존의 추이적 행렬을 이용한 알고리즘의 경우<sup>[10,15,25]</sup> 작업별로 구분 되어진 서브 넷의 마킹 흐름을 도달성 트리 형식으로 분석하여 교착상태를 결정하였으나, 제안한 알고리즘은 이러한 작업 별 서브 넷의 분리 작업 없이 행렬에서 수리적 산술법으로 교착여부 확인이 가능하여 더욱 빠

르게 교착여부를 확인 할 수가 있었다. 그러나 제안한 알고리즘에서는 교착상태여부를 빠르게 인지가 가능한 반면 교착상태 회피를 위한 알고리즘은 아직 제시되지 못하는 한계점이 있다.

### 5. 결 론

본 논문에서 추이적 행렬을 기반으로 하여 패트리 넷의 교착상태 문제 분석과 추이적 행렬을 기반으로 하는 자원공유 플레이스의 교착상태 조건을 제안하였다. 교착상태 문제는 충돌 플레이스에 기반한 두 개 이상의 트랜지션 관계에서 발생하였다. 그러므로 입력 토큰의 수가 출력 토큰의 수 보다 작으면 공유 플레이스에서 교착이 발생하였고, 충돌 플레이스내 입력토큰과 출력토큰의 관계를 이용한 교착상태 조건을 제안하였다. 또한 추이적 행렬을 이용하여 공유 플레이스의 교착상태 여부를 판단하는 수리적 판별식을 제안하였으며 그 실효성을 위하여 여러 예제에 적용하여 교착상태 및 교착자유상태 여부를 검증하였다. 제안 된 교착상태 감지 알고리즘은 다른 분석 알고리즘에 비해 쉽고 간단하게 교착 여부를 판별 가능한 특성을 가지나 회피를 위한 후속 작업 즉 회피와 방지 알고리즘이 아직 제시가 되지 못한 한계점을 갖는다. 향후 다른 교착상태 감지 알고리즘과의(siphon, 도달성 트리 등) 성능분석을 통한 활용성 검증, 추이적 행렬을 이용한 일반적 패트리 넷 모델 확장 및 FMS의 교착상태회피에 의한 사이클릭 스케줄링(cyclic scheduling) 연구로 확장해 나갈 필요가 있다.

### 참 고 문 헌

1. F.Basil, A.Giua, C.Seatzu,(2003), "Observer-based state-feedback control of time Petri nets with deadlock recovery: theory and implementation," *In Proc. Of Symp. On Discrete events in Industrial and manufacturing systems, CESA2003.*
2. Corbett JC(1996), "Evaluating Deadlock detection methods for concurrent software," *International Journal of IEEE Tr. Sof. Eng.* 22(3), pp. 161-180.
3. Damasceno BC. And Xie X.(1999), "Petri nets and deadlock-free scheduling of multiple-resource operations," *Proc. of International Conference of IEEE SMC'99*, pp. 878-883.
4. Desel J. And Esparza J.(1995), *Free Choice Petri nets*, Cambridge univ. Press.
5. Ezpleta J.,Colom JM, Martinez J.(1995), "A Petri net based

- deadlock prevention policy for flexible manufacturing systems,” *International Journal of IEEE Tr. Robotics and Automat.*, 11(2), pp. 173-184.
6. Giua A., et al. (2004), “Observer-Based state-feedback control of timed Petri nets with deadlock recovery,” *In: IEEE Tr. on Automatic control*, Vol. 49, No. 1, pp. 17-29.
  7. Han Z., Lee G. (2005), “Application of Petri nets for deadlock analysis and avoidance in flexible manufacturing systems,” *Int J. Adv. Manu. Tech.*, Vol. 25, pp. 735-742.
  8. Jeng ND, and DiCesare F. (1995), “Synthesis using resource Control Nets for Modeling Shaded-resource Systems,” *IEEE Tr. On Robotics and Automation*, Vol. 11, No. 3, pp. 317-327.
  9. Kim JW, Lee JK (2008), “Efficient deadlock detection in FMS based on the Transitive matrix of resource share places”, *In Proc. of ITC-CSCC2008*, Simonoski, Japan, pp. 277-280.
  10. Kim SH, Lee SH, Lee JK (2007), “Deadlock analysis of Petri nets based on the Resource Share Places Relationship”, *Studies in Informatics and control*, Vol. 16, No. 1, pp. 33-44.
  11. Li ZW, Zhao M. (2008), “On Controllability of Dependent Siphon for Deadlock prevention in Generalized Petri nets,” *IEEE Tr. SMC-A*, Vol. 38, No. 2, pp. 369-384.
  12. Li ZW, Hu HS, Wang AR (2007), “Design of Liveness-Enforcing Supervisors for Flexible manufacturing Systems Using Petri nets,” *IEEE Tr. SMC-C*, Vol. 37, No. 4, pp. 517-525.
  13. Li ZW, M.C. Zhou (2004), “Elementary siphons of Petri Nets and their application to deadlock prevention in flexible manufacturing system,” *IEEE Tr. On system, Man and Cybern*, part A, Vol. 34, pp. 38-51.
  14. Liu J., Itoh Y., Miyazawa I., Seikiguchi T. (1999), “A Research on Petri nets Properties using Transitive matrix,” *Proc. of 1999 International Conference on IEEE SMC99*, pp. 888-893.
  15. Lee J. (2004), “Deadlock find algorithm using the Transitive Matrix,” *In Proc. of CIE'04*.
  16. Lee J., Korbaa O. (2006), “Scheduling analysis of FMS: An unfolding time Petri nets approach,” *In: mathematics and Computer simulation*, 70, pp. 419-432.
  17. Melzer S. and Romer S. (1997), “Deadlock checking using net Unfoldings”, *In Proc. of the Conf. on Computer-Aided Verification (CAV'97)*.
  18. Murata T. (1989), “Petri Nets: Properties, Analysis and Applications,” *Proc. of the IEEE*, 77(4), pp. 541-580.
  19. Park J., Reveliotis SA (2001), “Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible manufacturing,” *IEEE Tr. Robotics and Automat.*, Vol. 46, pp. 1572-1583.
  20. Peterson JL (1987), *Petri Net Theory and the Modeling of Systems*, Englewood Cliffs, NJ : Prentice-Hall, Inc.
  21. Shatz SM., Tu S., Murata T. (1996), An application of Petri net reduction for Ada Tasking Deadlock Analysis, *In: IEEE Trans. on Parallel and Distributed Systems*, Vol. 7, No. 12, pp. 1307-1322.
  22. Uzam M. and Zhou MC (2007), “An Iterative Synthesis Approach to Petri net-based deadlock prevention Policy for Flexible manufacturing System,” *IEEE Tr. SMC-A*, Vol. 37, No. 3, pp. 362-371.
  23. Xiong HH, Zhou MC (1997), “Deadlock free scheduling of an automated manufacturing system based on Petri nets,” *In Proc. of the 1997 International Conference on IEEE ICRA'97*, pp. 945-950.
  24. Yoon H. and Lee D. (2000), “Deadlock-Free Scheduling for Automated Manufacturing cells,” *In Proc. of International Conference on Control, Automation Robotics and Vision 2000*.
  25. 송유진, 이종근 (2006), “DAPN과 인접행렬을 이용한 교차 상태 회피에 대한 연구”, *한국시뮬레이션학회 논문지*, 제 15권, 1호, pp. 1-10.





**김 종 옥** (jwkim@changwon.ac.kr)

1992 경상대학교전자계산학과 학사  
2002 창원대학교컴퓨터공학과 석사  
2006 창원대학교 컴퓨터공학과 박사 수료  
1992~현재 삼일 정보 기술(주) 대표이사

관심분야 : 패트리벨, 성능분석, 정보보호, 스케줄링 연구



**이 종 근** (jkleee@changwon.ac.kr)

1974 숭실대학교전자계산학과 학사  
1978 고려대학교경영학 경영학석사  
1987 숭실대학교 컴퓨터공학과 공학석사  
2002 LCGI / Ecole Centrale Paris 컴퓨터공학 공학박사  
1987~1990 LSI / Univ. de Montpellier II 연구원 역임  
1983~현재 창원대학교 컴퓨터공학과 교수

관심분야 : 패트리 넷, FMS 스케줄링 분석, 성능분석, 정보보호 관련 연구