

자원공유를이용한추이적행렬의교착상태확인및회피

김상환*, 이상호*, 이종근**

*충북대학교 컴퓨터공학과

**창원대학교 컴퓨터정보통신공학부

e-mail:jkleee@changwon.ac.kr

Deadlock Detection and Avoidance in transitive matrix Using the Resource share

Sang-Hwan Kim*, Sang-Ho Lee*, Jong-Kun Lee**

*Dept of Computer Science, Chungbook National University

**Dept of Computer Engineering, Changwon National University

요 약

본 연구에서는 유연생산시스템에서의 교착상태 확인 및 회피 알고리즘을 추이적 행렬을 이용하여 제안한다. 추이적행렬은 플레이스와 플레이스간의 관계를 표현함으로 마킹의 흐름을 플레이스와 트랜지션간의 관계형에서 상태와 상태간의 관계를 표현함으로 상태의 변화 검증에 편리하다. 교착상태 확인 및 회피 알고리즘을 제시하고, 기존에 발표되어진 siphon과 DAPN알고리즘간의 비료 검토를 통하여 제안한 알고리즘의 유용성을 검증하였다.

1. 서론

본 연구에서는 추이적 행렬[6,8,13,14]의 자원공유 관계식을 이용하여 넷의 교착 상태를 확인 및 회피 알고리즘을 제안하고 기존의 다른 알고리즘과 비교분석함으로 그 효율성을 검증한다. 추이적 행렬이란, 패트리 넷의 플레이스와 트랜지션간의 관계를 모두 표현함으로 토큰의 흐름을 행렬을 통하여 분석이 가능하며, 시작플레이스와 최종 플레이스 확인이 가능하고, 또한 흐름의 모든 내용을 표를 통하여 분석 가능한 특징을 가지고 있다. 특히 교착 가능한 플레 이스는 자원공유 플레이스임을 착안하여 플레 이스의 종류를 정의하고 각 플레이스간의 관계를 분석하여 교착상태 확인 및 회피 알고리즘을 제안하고 기존에 [1,3,5,9,11,12,15] 제안된 siphon과[4] DAPN알고리즘을[7] 비교한다. siphon은 교착상태의 확인에 널리 활용되어지는 알고리즘이며 DAPN알고리즘은 확인 과 회피를 플레이스간의 관계형으로 정리하고 제안한 알고리즘이다. 본 논문의 구성은 다음과 같다. 먼저, 2장에서 추이적 행렬을 설명하고, 3장에서는 추이적 행렬을 이용하여 교착 상태의 확인 및 회피정책을, 4장에서는 예제를 통하여 siphon과 DAPN알고리즘을 비교하여 추이적행렬 알고리즘에 대한 검증을 살펴봄, 마지막으로 결론과 향후 연구에 대하여 기술 한다.

2. 패트리 넷

패트리 넷은 일반적인 시스템의 모델링에 쓰이며, 다음과 같이 정의할 수 있다[2,10].

[정의 2.1] 패트리 넷 PN은 6-튜플로 구성된다: $PN = (P, R, T, I, O, Mo)$ 여기에서,

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

$R = \{r_1, r_2, \dots, r_m\}$: 자원공유플레이스의 유한 집합

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

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

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

$Mo: (PUR) \rightarrow \{0, 1, 2, 3, \dots\}$: 초기 마킹

$(PUR) \cap T = \emptyset$

[정의 2.2] 입력함수와 출력함수에 대한 행렬

PN 구조의 행렬 정의 C는 다음과 같다.

$C = ((PUR), T, B^-, B^+)$

$B^-[i, j, k] = \#((p_i \cup r_j), I(t_k))$

$B^+[i, j, k] = \#((p_i \cup r_j), O(t_k))$

여기서 B^- 와 B^+ 는 각각 입력과 출력함수에 대한 행렬이며, $B = B^+ - B^-$ 는 추이적 행렬(incidence matrix)이라 한다[6].

[정의 2.3] 플레이스 추이적 행렬(Place transitive matrix): $B_{P \cup R} = B^-(B^+)^T$

[정의 2.4][6] 표식화 플레이스 기반 추이적 행렬 (Labeled P-invariant Transitive Matrix): M_{PR}

M_{PR} 는 표식화 플레이스 기반 추이적 행렬이라고 하며 다음과 같이 정의된다.

$L_{BP} = B^- \text{diag}(t_1, t_2, t_3, \dots, t_n)(B^+)^T$

여기서, $t_i (i = 1, 2, 3, \dots, n)$ 은 다음과 같다.

$|t_i| = 1$ 이면 t_i 는 점화한다.

$|t_i| = 0$ 이면 t_i 는 점화하지 않는다.

[정의 2.5][6] 가중적 플레이스 기반 추이적 행렬 (Weighted P-invariant Transitive Matrix)

M_{PR} 은 $m \times m$ 가중적 플레이스 기반 추이적 행렬 이라고 정의한다. 만약 트랜지션 t_k 가 M_{PR} 의 같은 열에 s 번 나타난다면 M_{PR} 에 있는 t_k 를 M_{PR} 에서 는 t_k/s 로 표시한다.

[정의 2.6] 도달마킹 $M_R(K+1)$ 은 초기마킹 $M(k)$ 으로부터 도달되어지는 마킹으로 정수형 m -벡터라 한다. 도달 마킹의 표현 형태는 다음과 같다:

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

교착 상태란 시스템의 흐름에서 서로가 대기상태가 되어 마킹의 흐름이 정지 될 경우 이를 교착상태라 한다. 또한 이러한 상태는 주로 자원공유 플레이스에서 자주 발생되어지는 성질이 있다[5].

[정의 2.7] 교착 노드

1) 마킹 M 으로부터 어떠한 트랜지션으로 점화가능 하지 못하다면, 그 트랜지션은 교착 트랜지션이라고 하고 한다.

2) 마킹 M 으로부터 어떠한 플레이스로 마킹이 전이 가 되지 못하다면 그러한 플레이스를 교착플레이스라 한다.

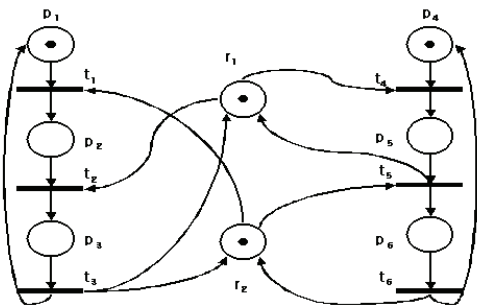
[정의 2.8] 교착 상태

패트리 넷에서 교착 노드가 하나 이상 나타나면 마킹 M 은 교착상태라 한다.

3. 추이적 행렬을 이용한 교착 상태 확인과 회피

3.1 교착상태 확인

교착상태는 자원공유 플레이스에서 발생하게 된다. 그림 2.1 과 같은 예에서처럼, 두 개의 트랜지션 t_2 과 t_5 는 플레이스 r_1 과 r_2 에서 교착 상태가 발생 된다 (그림 3.1).



(그림 3.1) 예제 패트리 넷

정의 2.6을 통하여, 예제의 도달 마킹관계식을 다음과 같이 정리 할 수가 있다:

$$M_0 = [1,0,0,1,0,0,1,1],$$

여기서 $[p_1, p_2, p_3, p_4, p_5, p_6, M_1, M_2]$.

도달마킹 M_1 은 :

$M_1 = M_0 M_{PR}$, 이면,

$$M_1 = [0, (p_1(t_1/2) + M_2(t_1/2)), M_1(t_2/2), 0, (p_5(t_4/2) + M_1(t_4/2)), M_2(t_5/2), M_2(t_5/2), 0].$$

토큰의 값은 0,1의 정수형이므로, 즉 1보다 작은 경우에는 0으로, 1보다 큰 경우에는 1로 제한하여 토큰의 값을 표현 할 수가 있다. 도달마킹 M_1 은 아래와 같이 나타 낼 수가 있다. :

$$M_1 = [0, 1, 0, 0, 1, 0, 0, 0].$$

다음 도달마킹은 :

$$\begin{aligned} M_R &= M_1 M_{PR} \\ &= [0,0, p_2(t_2/2), 0,0, p_5(t_5/2), 0,0] \\ &= [0, 0, 0, 0, 0, 0, 0, 0]. \end{aligned}$$

이제 여기서, 도달 마킹의 모든 토큰이 0이 되므로 어떠한 이후의 마킹이 점화가 불가능하게 됨으로, 트랜지션 t_2 와 t_5 가 플레이스 M_1 과 M_2 에 의하여 교착 상태가 발생하게 되어짐을 발견 할 수가 있다.

3.2 교착상태 회피

교착상태는 자원공유 플레이스에서 발생하므로, 이러한 교착 상태를 해소하기 위하여서는 교착 상태의 트랜지션에서 자원공유 플레이스에 점화후 새로운 토큰을 얻을 수 있도록 보완을 하게 되면, 도달 마킹에서 교착 상태를 해소 할 수가 있다.

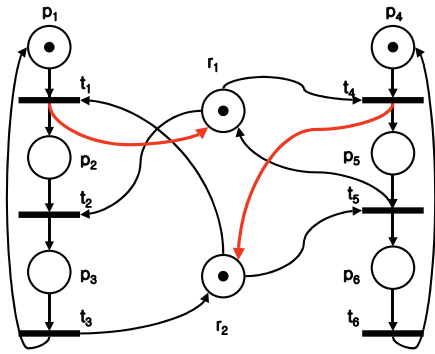
[정의 3.1] 교착상태 회피정책

플레이스 $\exists p_i, p_j \in P$ 가 트랜지션 t_i 와 t_j 에서 교착 상태라고하자. 여기서 $\exists t_i, t_j \in T, t_i \neq t_j, p_i \neq p_j, \cdot t_i = p_i, \cdot t_j = p_j$ 이고 플레이스 p_i 와 p_j 는 자원공유 플레이스라 한다. 그러면, 트랜지션 t_i 의 출력에 플레이스 p_j 를 추가하고, 트랜지션 t_j 에는, 출력 플레이스 p_i 를 추가한다.

$$t_i \cdot = O(p_i, t_i) \cup O(p_j, t_i)$$

$$t_j \cdot = O(p_j, t_j) \cup O(p_i, t_j)$$

이제 2 개의 기기 r_1 과 r_2 를 가지고, 그리고 2 개의 작업인 JOB_1 과 JOB_2 을 가진 유연생산 시스템의 예를 생각하여 본다(그림 3.2). 이 예제의 교착 자유상태 넷은 그림 3.2이 되며, 추이적 행렬은 표 3.1 이 된다.



(그림 3.2) 교착상태 회피 넷

<표 3.1> 그림 3.2의 추이적 행렬

$$M_{PR} = \begin{matrix} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & r_1 & r_2 \\ \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ r_1 \\ r_2 \end{matrix} & \begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{bmatrix} & \end{matrix}$$

이 추이적 행렬 M_{PR} 에서, 우리는 다음과 같은 도달 마킹 관련식을 얻을 수가 있다:

$$M_0 = [1,0,0,1,0,0,1,1], \quad \text{여기서}$$

$[p_1, p_2, p_3, p_4, p_5, p_6, r_1, r_2]$.

도달 마킹 관련식 M_R is :

$$M_R = M_0 \cdot M_{PR} \\ = [0, (p_1/2+r_2/2), (r_1/2), 0, (p_4/2+r_1/2), (r/2), (p_1/2+r_2/2), (p_4/2+r_1/2)].$$

$$= [0, 1, 0, 0, 1, 0, 1, 1]$$

여기서, 플레이스 p_2, p_5, r_1 와 r_2 가 토큰을 가진다. 따라서, 이 넷은 교착자유상태가 된다.

4. 비교분석

4.1 siphon & trap

siphon과 trap은 패트리 넷의 교착상태 특성으로부터 도입된 특별한 구조이다. Siphon과 Trap을 통하여 패트리 넷의 모든 도달 가능한 마킹을 조사하지 않고도 패트리 넷의 생동성(liveness)을 분석 할 수 있는 특징적 구조이다[4].

[정의 4.1]siphon과 trap

어느 플레이스 집합 S 에 대하여 $S \subseteq S'$ 를 만족하는 집합 S' 를 siphon이라고 한다. 또한 어느 플레이스 집합 Q 에 대하여 $Q' \subseteq Q$ 를 만족하는 집합 Q 를 trap 이라고 한다.

siphon과 trap을 통하여 교착상태에 대한 정의를 다음과 같이 정리한다[4].

[정의 4.2] 토큰이 모두 사라지는 siphon S 를 잠재적

인 교착(potential deadlock)이라한다.

[정의 4.3] minimal siphon

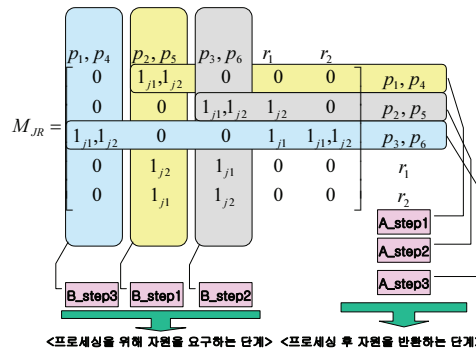
만일 하나의 siphon S_1 이 다른 siphon을 포함하고 있지 않다면 siphon s_1 은 minimal이라 한다.

[정의4.4] 어느 패트리 넷에 만일 minimal siphon이 없다면 이 패트리 넷은 교착상태가 아니다.

따라서, 예제의 모델을(그림 3-1) siphon을 이용하여 교착상태를 찾아본다면 먼저 모델에서 사이클 프로세스: $S_1=\{p_1, r_2, p_3\}$, $S_2=\{r_1, p_3\}$, $S_3=\{r_2, p_2, p_3, p_6\}$, $S_4=\{r_1, p_3, p_5\}$ 에서 s_1, s_2, s_3, s_4 모두 siphon이며, 그중에서 s_2 만 제외하고 모두 trap의 성질을 갖는다. 그러나, s_1 이 s_3 에 포함되는 minimal siphon이 되어 교착상태가 발생된다. siphon은 교착상태만을 확인하는 알고리즘이며, 사이클을 찾는 과정 그리고 각 사이클 프로세스가 siphon과 trap의 조건을 만족하는가 과정등이 매우 복잡하며 특히 복잡한 모델의 경우 분석 시간과 분석 알고리즘은 상당히 어려워진다.

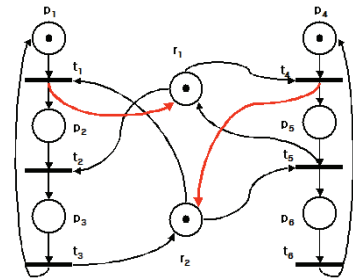
4.2 DAPN

DAPN은[7] 플레이스와 플레이스간의 관계를 그래프 이론을 사용하여 인접 행렬을 구성하고 각각의 열과 행의 처리 과정에서 시작 마킹을 찾아 도달성 트리와 같은 과정으로 마킹 처리 결과에서 음수가 되는 지점을 찾아 교착상태를 확인하며, 회피는 이러한 음수의 마킹 값을 양수화하면 교착 회피 상태를 극복하는 알고리즘이다.



(그림4.1) DAPN의 인접행렬

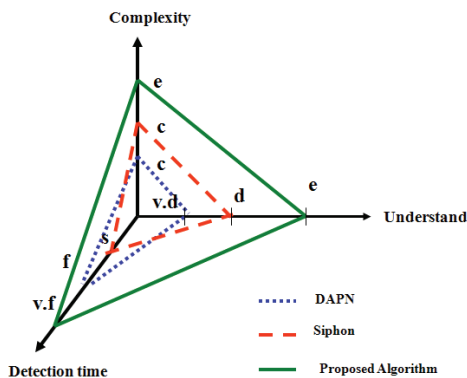
교착 회피알고리즘을 사용한 회피 모델은 다음과 같다. 이는 제안한 알고리즘과 같은 결과를 갖는다.



(그림4.2)DAPN의 회피결과

4.3 벤치마킹

기존의 siphon과 DAPN 그리고 제안 알고리즘을 비교하여 보면, 실제적으로 알고리즘을 프로그래밍하여 알고리즘의 복잡도와 정확도등을 여러 개의 모델을 통하여 비교분석을 하여야하나, 실질적으로 정량적으로 비교 분석함에는 무리가 있다고 생각된다. 특히, 사이클을 찾아서 siphon의 성질을 분석함에 알고리즘이 쉽지가 않아, 정성적으로 알고리즘을 비교하여 이해도와 복잡도 그리고 감지 시간을 비교 항목으로 그래프로 나타내 보면 다음과 같은데, siphon은 이해도는 DAPN보다 좋으나, 분석 시간은 오히려 DAPN보다 긴 시간을 요구하며, DAPN의 경우 복잡한 알고리즘을 가지나 분석 시간은 좋은 편이다. 결국 자원공유 플레이스를 기반으로 한 추이적 행렬알고리즘에 의한 교착상태 확인 및 회피 알고리즘이 이해도, 처리시간과 알고리즘 복잡도에서도 기존의 다른 알고리즘보다 효율성이 높다고 할 수가 있다.



(그림 4.3) 비교분석표

5. 결론 및 앞으로의 연구 방향

이 연구에서, 우리는 추이적 행렬을 이용하여 넷의 교착 상태를 분석하고 회피하는 알고리즘을 제시하였다. 자원의 공유 되어진 모델 분석은 조합 최적화 문제로써 NP-hard 문제이다. 즉 자원 공유의 수가 많거나 복잡하여 질 경우 복잡도가 지수적으로 증가하는 문제가 있다. 따라서, 이러한 모델의 교착 여부를 확인하는 데에는 기존의 siphon과 DAPN보다 자원공유의 추이적 행렬 분석이 복잡도의 증가를 최소화 했을 뿐 아니라, 초기 모델을 그대로 활용하여 제시함으로써 이해용이성 부분을 고려한 분석 모델을 제시 하였다. 따라서 앞으로의 연구에서는 일반 페트리 넷에도 적용 가능하도록 추이적 행렬의 확장 과 그 응용을 통하여 이산상태 시스템에 적용토록 할 것이며, 또한 교착 상태 해소 기법의 자동 패키지화에 대하여 연구하고자 한다.

참고문헌

- [1]Corbett JC,(1996), "Evaluating Deadlock detection methods for concurrent software", IEEE tr. Sof. Eng. Vol.22 (3), 1996, pp.
- [2]Berthomier B. and Diaz M.(1991).Modeling and Verification of Time Dependent Systems using Time Petri nets, *IEEE Tr. Soft Eng.*, **17**,pp. 259-273
- [3]Damasceno BC. And Xie X.(1999), Petri nets and deadlock-free scheduling of multiple-resource operations, In *IEEE SMC'99*,pp.878-883
- [4]F. Chu and X-L. Xie(1997), Deadlock Analysis of Petri nets using Siphon and Mathematical Programming, *IEEE Tr on Robotics and Automation*, vol.13,no.6,pp.793-804
- [5] Ezpleta J.,Colom JM,Martinez J.(1995), A Petri net based deadlock prevention policy for flexible manufacturing systems, *IEEE tr. Robotics and Automat.*, Vol.11,no.2,pp.173-184
- [6]Liu J.,Itoh Y., Miyazawa I, Seikiguchi T.,(1999). A Research on Petri nets Properties using Transitive matrix, In: *Proceeding IEEE SMC99*,pp.888-893
- [7]송유진,이종근(2006),"DAPN과 인접행렬을 이용한 교착 상태 회피에 대한 연구",한국시물레이션학회논문지,15집(1권),pp.1-10
- [8]Lee J., (2004) Modeling and Scheduling ratio-driven FMS using Unfolding TPN,In *Computer& Indertrial Engineering*, vol46(4),pp.639-653
- [9]Melzer S. and Romer S.(1997). Deadlock checking using net Unfoldings, In *Proc. of the Conf. on Computer-Aided Verification (CAV'97)*
- [10]Murata T.(1989). Petri Nets: Properties, Analysis an Applications, *Proceedings of the IEEE*, **77(4)**, IEEE,USA pp. 541-580.
- [11]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
- [12] Xiong HH,Zhou MC.(1997), Deadlock free scheduling of an automated manufacturing system based on Petri nets",In *IEEE ICRA'97*,pp.945-950
- [13] Kim S.H. Lee S.H. and Lee J.K.,(2007)"Deadlock analysis of Petri nets based on the resource share places relationship" In: *Studies in Informatics and Control*, vol. 16(1), the National Institute for Research & Development in Informatics
- [14] Kim S.H. Lee S.H. and Lee J.K.,(2005)"Deadlock avoidance policy algorithm in FMS using the Transitive matrix", In *proceeding IMACS 2005*, Pairs.
- [15]Giua A.,et etc.,(2004), Observer-Based state-feedback control of timed Petri nets with deadlock recovery, In: *IEEE Trans. on Automatic control*, Vol. 49, No. 1, pp. 17-29