추이적 행렬을 이용한 패트리 넷의 교착 상태 확인 분석

송유진⁰, 이종근
정보시스템연구실/컴퓨터공학과
창원대학교, 641-773, 창원, 경남
(syj⁰,jklee@sarim.changwon.ac.kr)

Analyze Method of Deadlock status in Petri nets Using the Transitive Matrix

Yujin Song, Jongkun Lee
LIS/Dept. of Computer Engineering
Changwon National Univ.

요약

본 연구에서는 패트리 넷에서의 교착 상태 확인을 추이적 행렬을 이용하여 분석하는 기법을 제안한다. 교착 상태란 패트리 넷에서 마킹이 더 이상 진행 되지 못하고 서로 점화 가능 상태를 기다리는 상태로 자원 공유의 문제에서 많이 발생 가능하다. 따라서, 모든 플레이스와 트랜지션과의 관계를 나타내는 추이적 행렬을 이용하여 간단하게 확인분석이 용이한 기법을 제안한다.

1. 서론

패트리 넷은 비동기적이며, 동시 발생적인 이벤트에 의해 시스템의 상태가 변화하는 이산사건 시스템(DES: Discrete Event System)을 모델링 하는데 아주 적절한 도구이며, 유연생산 시스템 (FMS: Flexible Manufacturing Systems) 환경에 고유한 라우팅, 자원공유, 공유된 자원간의 상호 배제와 같은 특징들을 모델링 하는데 적절한 도구가 된다. 일반적으로 시스템을 패트리 넷을 이용하여 모델링하면 패트리 넷의 도달가능그래프는 시스템의 전체상태를 표현하게 되며, 트랜지션의 점화순서는 그대로 시스템 운영에 있어서의 흐름이 된다 즉, 패트리 넷의 마킹은 모델링 된 시스템의 상태를 나타내며, 이러한 상태는 트랜션의 점화를 통해 변화하므로 트랜지션의 점화 순서를 제어하면 원하는 시스템의 상태 평가와 검증이 가능하다.
시스템에서 중요한 것은 이러한 마킹의 흐름이 원할하게 순환되어지는 것인데, 서로가 대기상태가 되어 마킹의 흐름이 정지 될 경우 이를 교착 상태라 한다. 시스템에서의 교착 상태 확인과 방지는 아주 중요한 과제중의 하나이다. 따라서, 많은 연구[*1,2,4,8*]가 이러한 교착 상태 확인과 방지를 위하여 제시되어 왔다. 특히 실시간 처리가 필요한 이산사건시스템의 응용에서 많은 연구가 이루어졌다. 제시되어진 연구들은 모두 상황 논리적으로 제시되었기 때문에 많은 시스템이나 프로세스가 포함된 모델에서의 이해 및 적용성 문제가 쉽지 않았다.

본 연구에서는 패트리 넷에서의 플레이스와 트랜지션간의 관계를 표현하는 추이적 행렬[*5,6*]을 이용하여 넷의 교착 상태를 확인 분석하는 방법을 제안한다. 넷의 전체적인 자료인 추이적 행렬을 통하여 교착 상태가 발생하는 위치를 확인함으로써 시스템의 성질 분석 및 검증이 용이한 특징이 있다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서 패트리 넷의 기본 성질을 설명하고, 3장에서 추이적 행렬을 설명한다. 4장에서는 추이적 행렬을 이용하여 교착 상태의 사례를 살펴보며, 마지막으로 결론과 향후 연구에 대하여 기술 한다.

2. 패트리 넷(*Petri Net*)

패트리 넷은 일반적인 시스템의 모델링에 쓰이며, 다음과 같이 정의할 수 있다[7].

$$PN = (P, T, I, O, M_0)$$

여기에서,

$P = \{p_1, p_2, \ldots\ldots, p_n\}$: 플레이스의 유한 집합 $(n \geq 0)$

$T = \{t_1, t_2, \ldots\ldots\ldots, t_m\}$: 트랜지션의 유한 집합 $(m \geq 0)$

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

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

$M_0 : P \rightarrow \{0,1,2,3,\ldots\ldots\}$: 초기 마킹

$P \cap T = \phi$

이 때 플레이스는 시스템의 상태 혹은 조건을 나타내며 그림에서 원으로 표시한다. 트랜지션은 시스템의 상태를 변화시키는 동작(event)을 나타내며 그림에서 선분으로 표시한다. 아크는 상태의 흐름으로써 화살표로 표시하고, 토큰은 플레이스의 조건의 진위 또는 시스템의 가용자원을 나타낸다. 토큰은 시스템의 동적이며 병행적인 동작의 특성을 나타내기 위해 사용되는데, 동작이 일어나도록 하는데 필요한 조건을 만족할 경우 플레이스에 토큰을 위치시킴으로써 표현한다. 트랜지션은 자신에게로 입력되는 모든 플레이스가 토큰을 보유하고 있어야 점화(fire)될 수 있다. 트랜지션이 점화되면 자신의 각 입력 플레이스로부터 토큰을 하나씩 제거하고 각 출력 플레이스에 토큰을 하나씩 첨가한다.

따라서 토큰의 수와 위치는 패트리 넷을 실행하는 동안 바뀌게 된다.

[정의 2.1][3] 교착 상태
패트리 넷에서 어떠한 트랜지션이 더 이상 점화 가능하지 않다면 마킹 M은 교착상태라 한다.

3. 추이적 행렬

[정의 3.1][5] 입력함수와 출력함수에 대한 행렬
PN 구조의 행렬 정의 C는 다음과 같다.

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

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

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

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

[정의 3.2][6] 인접 행렬(Adjacent matrix)과 추이적 행렬(Transitive matrix) 관계:

$$A=\begin{bmatrix} 0 & (B^+)^T \\ B^- & 0 \end{bmatrix}$$ 여기서 A 는 인접 행렬(Adjacent matrix)

$$S=\begin{bmatrix} (B^+)^T B^- & 0 \\ 0 & B^-(B^+)^T \end{bmatrix}$$ 여기서 S 는 추이적 행렬(Transitive matrix)

플레이스 추이적 행렬(Place transitive matrix) :

$$B_P=B^-(B^+)^T$$

트랜지션 추이적 행렬(Transition transitive matrix) :

$$B_T=(B^+)^T B^-$$

[정의 3.3][6] 표식화 플레이스 기반 추이적 행렬(Labeled P-invariant Transitive Matrix): L_{BP}

L_{BP} 는 표식화 플레이스 기반 추이적 행렬이라고 하며 다음과 같이 정의된다.

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

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

$|t_i|=1$ 이면 t_i 는 점화한다.

$|t_i|=0$ 이면 t_i 는 점화하지 않는다.

L_{BP} 요소는 하나 혹은 그 이상의 트랜지션들을 통해 한 플레이스로부터 다른 플레이스로의 이동을 나타낸다.
여기서 패트리넷의 기본적인 구성 요소와 표식화 플레이스 기반 추이적 행렬과의 기본적인 관계를 알 수 있다. 또한 L_{BP} 에 의해 패트리넷의 트랜지션 점화 조건을 알 수 있다.
[정의 3.4][6] 가중적 플레이스 기반 추이적 행렬(Weighted P-invariant Transitive Matrix) L_{BP}^*

L_{BP}^* 는 $m\times m$ 가중적 플레이스 기반 추이적 행렬이라고 정의한다. 만약 트랜지션 t_k 가 L_{BP} 의 같은 열에 s 번 나타난다면 L_{BP} 에 있는 t_k 를 L_{BP}^* 에서는 t_k/s 로 표시한다.
[정리 3.1] 추이적 행렬의 행 방향은 제어흐름을 표현한다.
따라서 $\sum \frac{t_{k_i}}{d_i}$ 의 값에 따라 제어흐름을 알 수 있다.

만약 $\#(p_i,Ot(t_k))<\#(p_i,Et(t_k))$ 이면, 해당 플레이스의 병행적인 흐름이 유지되거나 확장되지 않고, 축소됨을 뜻한다.
이를 유형별로 정리하면 다음과 같다.

각각의 $\sum \frac{t_{k_i}}{d_i}$ 에 대해,

(1) $\#(p_i,Ot(t_k))=\#(p_i,Et(t_k))$ 인 경우 $\Rightarrow \sum \frac{t_{k_i}}{d_i}=1$

$\Rightarrow$ 토큰 t_k 의 점화로 인한 제어 영역이 그대로 유지된다.

(2) $\#(p_i,Ot(t_k))>\#(p_i,Et(t_k))$ 인 경우 $\Rightarrow \sum \frac{t_{k_i}}{d_i}>1$

$\Rightarrow$ 토큰 t_k 의 점화로 인한 제어 영역이 넓어지게 된다.

(3) $\#(p_i,Ot(t_k))<\#(p_i,Et(t_k))$ 인 경우 $\Rightarrow \sum \frac{t_{k_i}}{d_i}<1$

$\Rightarrow$ 토큰 t_k 의 점화로 인한 제어 영역이 좁아지게 된다.

[정리 3.2] 추이적 행렬의 열 방향은 각 트랜지션의 점화 가능성을 표현한다. 또한, 열 방향의 자원 공유 플레이스에는 토큰이 존재한다.
그리고 자원 공유 플레이스의 열 방향에 표현된 트랜지션의 수는 그 플레이스로 입력되는 트랜지션의 수를 말하므로 $\sum \frac{t_{k_i}}{s_i}$ 의 값에 따라 해당 트랜지션의 점화 여부를 알 수 있는데, $\sum \frac{t_{k_i}}{s_i}$ 의 값이 1이면 점화가 가능하다.

이를 유형별로 정리하면 다음과 같다.

각각의 $\sum \frac{t_{k_i}}{s_i}$ 에 대해,

(1) $\#(p_i,Ot(t_k))=\#(p_i,Et(t_k))$ 인 경우 $\Rightarrow \sum \frac{t_{k_i}}{s_i}=1$

(2) $\#(p_i,Ot(t_k))>\#(p_i,Et(t_k))$ 인 경우 $\Rightarrow \sum \frac{t_{k_i}}{s_i}=1$

(3) $\#(p_i,Ot(t_k))<\#(p_i,Et(t_k))$ 인 경우 $\Rightarrow \sum \frac{t_{k_i}}{s_i}=1$

따라서, 추이적 행렬의 열 방향의 트랜지션의 합은 어떤 경우라도 1이 됨을 알 수 있으며, 만약 $\sum \frac{t_k}{s}$ 의 값이 1이 되지 않으면 패트리 넷 모델이 잘못 되었음을 알 수 있다.

4. 추이적 행렬을 이용한 교착 상태 확인

이 장에서는 간단한 사례를 통하여 제시한 교착 상태 기법을 검증하여 그 효율성을 살펴 본다.

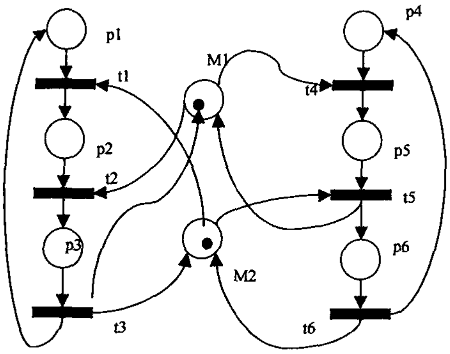

그림 1. 사례 모델

표 1. 추이적 행렬

$$
\begin{array}{cc}
 & \begin{array}{cccccccc} P1 & p2 & p3 & p4 & p5 & p6 & M1 & M2 \end{array} \\
L^{*}_{BP} = & \left[\begin{array}{cccccccc}
0 & t1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & t2/2 & 0 & 0 & 0 & 0 & 0 \\
t3 & 0 & 0 & 0 & 0 & 0 & t3 & t3 \\
0 & 0 & 0 & 0 & t4/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & t5/2 & t5/2 & 0 \\
0 & 0 & 0 & t6 & 0 & 0 & 0 & t6 \\
0 & 0 & t2/2 & 0 & t4/2 & 0 & 0 & 0 \\
0 & t1/2 & 0 & 0 & 0 & t5/2 & t5/2 & 0
\end{array}\right]
\begin{array}{c} P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ M1 \\ M2 \end{array}
\end{array}
$$

(1) 자원공유 플레이스는 M1과 M2이다.
(2) 행방향의 M1을 선택한다. (여기에 있는 트랜지션은 t2와 t4이다.)
(3) 트랜지션 t2를 선택한다.
(4) t2는 열방향으로 보았을 때, 일반 플레이스 p3에만 연결되어 있으므로 건너뛴다.
(5) t4는 열방향으로 보았을 때, 일반 플레이스 p5에만 연결되어 있으므로 건너뛴다.
(6)계속해서, p3을 따라서 진행하면 p1으로 가게 되는데, p1에서 열방향에서 M2의 t1이 존재하지 아니하므로 교착 상태가 발생하는 것을 알 수가 있다.

따라서, 행 방향의 자원공유 플레이스에 있는 트랜지션의 열 방향으로 일반 플레이스에 연결되어 있다면, 열 방향으로 자원공유 플레이스에도 연결되어 있는 트랜지션이 적어도 하나는 있어야 하므로 조건에 위배되어 이 모델은 교착상태에 걸린다.

5. 결론 및 앞으로의 연구 방향

이 연구에서, 우리는 추이적 행렬을 이용하여 넷의 교착 상태 여부를 분석하는 기법을 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 추이적 행렬에서의 자원공유를 중심으로 분석이 가능하므로 복잡도의 증가를 최소화 했을 뿐 아니라, 초기 모델을 그대로 활용하여 제시함으로써 이해용이성 부분을 고려한 분석 모델을 제시하였다.
그러나, 추이적 행렬은 아직 일반적인 페트리 넷에는 적용되지 못하는 단점이 있다. 즉 단일 아크와 단일 토큰만이 각 플레이스에 저장이 가능한 경우의 모델에서만 적용 가능한 한계가 있다. 따라서 앞으로의 연구에서는 일반 페트리 넷에도 적용 가능하도록 추이적 행렬의 확장과 응용을 통하여 이산상태 시스템에 적용토록 할 것이며, 또한 교착 상태 해소 기법에 대하여 연구하고자 한다.

참고문헌

[1]Corbett JC, "Evaluating Deadlock detection methods for concurrent software", IEEE tr. Sof. Eng. Vol.22(3),1996, pp.
[2]Damasceno BC. And Xie X. Petri nets and deadlock-free scheduling of multiple-resource operations, In IEEE SMC'99,1999,pp.878-883
[3]Desel J. And Esparza J.(1995). Free Choice Petri nets. In Cambridge univ. press
[4] Ezpleta J.,Colom JM,Martinez J., A Petri net based deadlock prevention policy for flexible manufacturing systems, IEEE tr. Robotics and Automat., Vol.11,no.2,1995,pp.173-184
[5] Lee J., Korbaa O., Gentina J-C(2001). Slice Analysis Method of Petri nets in FMS Using the Transitive Matrix", In: Proceeding INCOM01,2001
[6] LIU J., Itoh Y., Miyazawa I., Seikiguchi T., "A Research on Petri nets Properties using Transitive matrix", in proceeding IEEE SMC99,1999,pp.888-893
[7] Peterson JL, Petri Net Theory and the Modeling of Systems, Englewood Cliffs, NJ : Prentice-Hall, Inc., 1981.
[8] Xiong HH,Zhou MC., Deadlock free scheduling of an automated manufacturing system based on Petri nets",In IEEE ICRA'97, 1997,pp.945-950