2001년도 대한전기학회/대한전자공학회 시스템및제어분야 합동 추계학술대회 논문집 (2001.11.24)

Siphon을 이용한 이산 사건 시스템의 Deadlock 해석

김 정 철, 김 진 권, 황 형 수
원광대학교 제어계측과

Deadlock Analysis of Discrete Event Systems Using Siphon

Jung-Chul Kim, Jin-Kwon Kim, Hyung-Soo Hwang
Dept. of Control & Instrumentation Engineering, Wonkwang Univ

Abstract - Siphons and traps are special structures of Petri nets with closely related to liveness for the deadlock analysis of Discrete Event Dynamic Systems. It can analysis the liveness of Petri nets using the siphon and trap without to check reachability.[1] Deadlock analysis approach we proposed is based on the notion of potential deadlock which are siphons that eventually become empty. And in this paper, we proposed a initial marking method by siphon to avoid and to prevent deadlock. It is shown that our approach is more efficient than classical state enumeration approach.

1. 서 론

최근에 설계되는 생산 시스템은 제품 제조의 경비 절감과, 소비자의 다양한 욕구를 만족시키는 다품종 소량 생산 체제로의 전환과 제품 생산의 고속화를 추구하고 있다. 이것은 생선업계의 FA(Factory Automation)를 확산시켜 나가려는 움직임을 유발시켰으며, 컴퓨터 제어 응용으로 인하여 더욱 가속화되었다.

FA의 한 분야인 유연성 생산 시스템(Flexibility Manufacturing System : FMS's)은 다양한 종류의 로봇, 컨베이어, 무인 반송차(AGV)등을 이용하여, 자동화된 조립 과정을 통해 인력, 경비와 작업 위험을 줄이고 생산 능률을 극대화하는 것이 목표이다.

FMS's는 이산 사건 시스템(Discrete Event System)으로 분류되어 진다. 이러한 FMS's는 동시에 여러 가지 제품들을 처리하는 특징을 가지고 있다. 각 제품들은 각각의 제품 형태의 작업과정에 따라 정해진 순서를 가지고 machine들을 통해서 생산되며, 제품들은 작업과정을 거치면서 유한 자원을 점유하기 위하여 경합을 하게 된다. 이러한 유한 자원의 경합은 deadlock 발생의 원인이 된다.[4] 유한 자원과 관련된 deadlock은 자원의 수를 조절함으로서 deadlock 발생을 막을 수 있다. 따라서 본 논문에서는 페트리 네트의 특별한 구조인 Siphon과 Trap을 이용하여 deadlock이 발생하지 않도록 초기 마킹을 설정하는 방법을 제시한다. 즉, 초기마킹에서 유한 자원의 수를 조절함으로서 deadlock 발생을 사전에 방지 할 수 있다.

2. 본 론

2.1 페트리 네트

페트리(Carl Adam Petri)가 이산사건 시스템의 해석에서 페트리 네트를 제시한 이래 많은 분야에서 응용되고 있다. 페트리 네트는 동시적(concurrent), 비동기적(asynchronous), 분산적(distributed), 병렬적(parallel), 비확정적(nondeterministic), 그리고 확정적(stochastic)인 현상을 갖는 정보처리, 작업공정 시스템 등을 묘사하고 분석, 연구하는 데에 유익한 도구이다.

그래픽 도구(graphic tool)로서, 페트리 네트는 플로우 챠트(flow chat)나 블록다이어그램(block diagram)또는 네트워크(network)와 유사한 시각적인 정보매체로 사용될 수 있다. 또한 정보의 흐름을 나타내는 토큰(token)은 시스템의 동적이고 동시 발생적인 상황과 활동을 모의실험 하는 데에 사용될 수 있다. 페트리 네트는 이산사건 시스템의 명확한 규정(formal specification), 명확한 확인(formal verification), 그리고 성능평가(performance evaluation)에 대한 통일된 환경을 제공하며, 시스템의 관계를 시각적으로 보여주어 시스템의 모델링 도구이다.

페트리 네트는 그래프 표현이므로 FMS(flexible manufacturing systems)등과 같은 실시간 제어(real-time control)의 구현에 사용될 수 있으며 토큰 정보로부터 시스템의 각 부분의 상태를 실시간 감시(monitoring)를 할 수 있고 또한 PLC(programmable logic controller)와 같이 여러 요소시스템(subsystem)을 순서대로 동작시키고 조정한다. 시스템에 대한 페트리 네트 모델로부터 시뮬레이션(simulation)을 할 수 있고, 이로부터 성능 평가를 할 수 있다. 페트리 네트는 상호 작용하는 동시발생 구성요소를 갖는 이산사건 시스템을 모델링하고 설계할 수 있는 이론적이며 시각적인 도구이다.

2.1.1 페트리 네트의 기호와 표현

일반적으로 페트리 네트는 5개의 원소로 구성되어 있다.[5][6][7]

$N = \langle P, T, I, O, M_0 \rangle$

여기서, $P = \{P_1, P_2, \cdots\cdots, P_m\}$ 는 유한한 플레이스의 집합이며, $T = \{t_1, t_2, \cdots\cdots, t_n\}$ 는 유한한 트랜지션의 집합이다.

(1) P (Place) : 조건, 자원의 사용 가능성 또는 공정상태, 원(Circle)으로 표기한다.
(2) T (Transition) : 사건(event) 또는 공정의 시작과 끝, 막대(Bar)로 표기한다.
(3) I (Input Function) : Place로부터 Transition으로의 방향성 화살표(Arc/Arrow).
(4) O (Output Function) : Transition으로부터 Place로의 방향성 화살표.
(5) M (Marking) : 각 플레이스에 있는 토큰의 개수 (M_0 : 초기마킹)

2.1.2 페트리 네트의 실행규칙

페트리 네트는 트랜지션의 모든 입력 플레이스에 토큰이 있을 때, 트랜지션은 활성화(Enabled)된다는 활성화 규칙과 활성화된 트랜지션이 점화하면, 모든 입력 플

레이스에서 하나의 토큰이 제거되고, 모든 출력 플레이스에 하나의 토큰이 더해지는 점화규칙을 따른다.

〈 정의 2.1.1 〉 **활성화 규칙** : 주어진 페트리 네트 N=⟨P, T, I, O, M_0⟩에서, marking M을 가지는 t_j가 있다. $\forall p_i \in P$ 일 때 $M(p_i) \geq mult(p_i, I(t_j))$ 이면, t_j는 활성화(Enable)된다. 여기서, mult(x, Y)는 bag Y내의 item x의 중복도(multiplicity)이다.

〈 정의 2.1.2 〉 **점화 규칙** : 주어진 페트리 네트 N=⟨P, T, I, O, M_0⟩에서, marking *M*을 가지는 t_j가 있다. 이때, t_j가 enabled가 되면 발화(Fire)라고 한다. 발화된 후 새로운 marking M′에 의한 결과는 다음 식과 같다.

$$M' = M_0 + CY$$

여기서, C는 사건발생 행렬(incidence matrix)이며, Y는 점화 순열(Firing vector)이다. 일반적으로 이 방정식을 상태 방정식이라 부른다.

2.2 Siphon과 Trap

Siphon과 Trap은 페트리 네트의 특별한 구조이다.[2][3] 이것들은 페트리 네트의 liveness의 성질과 매우 밀접한 관계에 있으며, 이를 통해 페트리 네트의 모든 도달 가능한 마킹을 조사함이 없이 페트리 네트의 liveness 성질을 분석할 수 있다.

〈 정의 2.2.1 〉 어느 플레이스 집합 S에 대하여 ·S⊆S· 을 만족하는 집합 S를 **Siphon**이라고 한다.

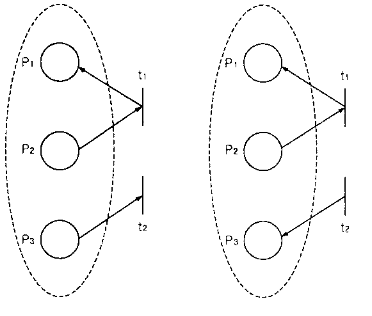
여기서, ·S는 플레이스 집합 S의 입력 트랜지션들의 집합이며, S· 는 플레이스 집합 S의 출력 트랜지션들의 집합이다.

정의에 따라서, S에 속하는 플레이스의 입력 트랜지션이 그 플레이스의 출력 트랜지션에 반드시 속하게 된다.

〈 정의 2.2.2 〉 어느 플레이스 집합 Q에 대하여, Q·⊆·Q를 만족하는 집합 Q를 **Trap**이라고 한다.

정의에 따라서, Q에 속하는 플레이스의 출력 트랜지션이 그 플레이스의 입력 트랜지션에 반드시 속하게 된다.

아래의 〈 그림 1〉은 Siphon과 Trap의 예이다.



〈 그림 1 〉 Siphon과 Trap

Siphon에 존재하는 마킹은 Siphon으로 되돌아오거나, 외부로 나간다. 따라서, Siphon에 초기에 토큰이 존재하지 않는 경우에 영원토록 토큰이 들어오지 않게 된다. 이와는 반대로 Trap에 존재하는 마킹은 Trap안으로 반드시 돌아오고, 외부로부터 들어오는 토큰은 계속 Trap안에 머문다. 따라서, Trap안에 토큰이 초기에 있으면, 영원토록 Trap안에 토큰이 존재하게 된다.

일반적으로 서로 다른 두 개의 Siphon(Trap)의 합집합도 Siphon(Trap)이다. Basic Siphon(Trap)이란 더 이상 서로 다른 두 개의 Siphon(Trap)의 합으로 나타낼 수 없는 Siphon(Trap)을 말한다. Minimal Siphon(Trap)이라는 것은 또다른 Siphon(Trap)을 포함하고 있지 않은 Siphon(Trap)을 의미한다. 따라서 Minimal Siphon(Trap)은 항상 Basic Siphon(Trap)이 되지만, Basic Siphon(Trap)이 항상 Minimal Siphon(Trap)이 되는 것은 아니다. 즉 Basic Siphon(Trap)은 쪼갤 수는 없지만 더 작은 Siphon(Trap)을 포함할 수 있다.

아래의 정리는 여러 종류의 페트리 네트에서 liveness와 Siphon, Trap의 관계이다.

〈 정리 2.2.3 〉 Free-choice net에서 모든 siphon이 marked trap을 포함하면 live하다.

〈 정리 2.2.4 〉 Live한 free-choice net에서 네트가 strongly-connected SM(State Machine) component에 의해 covered 되고 그 SM component가 유일하게 하나의 토큰만을 가지면 그 네트는 safe하다.

〈 정리 2.2.5 〉 An asymmectric choice net (N, m0)는 모든 siphon이 marked trap을 포함하면 live하다.

2.3 초기 마킹에서의 Deadlock-Free

이번 장에서는 Siphon과 Trap의 성질을 이용하여 상태방정식을 대수적인 조작을 통해 deadlock을 회피하는 초기 마킹 설정 방법을 제시한다.

Deadlock 해석에서 필요한 Siphon과 Trap의 중요한 특성과 정의는 다음과 같다.

〈 특성 2.3.1〉

1) 어떤 마킹에서 토큰이 없는 Siphon S는 토큰이 없는 상태(token-free)를 유지한다.
2) 어떤 마킹에서 토큰이 생성되어 마크된(marked) Trap은 마크된 상태를 유지한다.
3) 활성화된 트랜지션이 없는 어떤 마킹에서, empty 플레이스의 집합은 Siphon을 형성한다.

〈 정의 2.3.1 〉 활성화된 트랜지션이 없는 마킹 M은 **Dead Marking**이라고 한다.

〈 정의 2.3.2〉 결국에는 empty되는 Siphon S를 **Potential Deadlock**이라 한다.

〈 특성 2.3.2〉 만약 결국에는 empty되는 minimal siphon이 없다면 페트리 네트는 deadlock-free 이다.

〈 특성 2.3.3〉 마크된(marked) Trap을 포함한 어떤 Siphon은 potential deadlock이 아니다.

페트리 네트가 만약 potential deadlock인 siphon이 없다면 deadlock-free가 명백하며, 이와 관련된 함수 F(S)를 다음으로 정의한다.

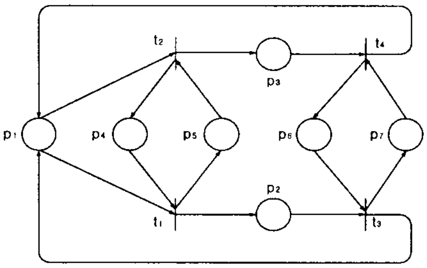
$$F(S) = \min\{M(S) | M = M_0 + CY,\ M \geq 0,\ Y \geq 0\}$$

여기서, C는 사건발생행렬이며, Y는 점화 순열이다. 따라서, 어떤 siphon S에서 F(S)>0을 만족하면 그 siphon은 potential deadlock이 아니다.

F(S)>0인 상태를 얻기 위해 siphon의 특성을 이용하여 상태 방정식을 풀면 potential deadlock이 없는 초기 마킹을 얻을 수 있다. 상태 방정식의 풀이 순서는 다음과 같다.

1) 상태 방정식을 작성한다.
2) Siphon에 포함된 플레이스는 토큰의 수를 zero로 한다.
3) 2)에서 zero로 표현된 부분의 상태 방정식을 전개하여 간략화한다.
4) 간략화한 방정식을 다른 방정식에 대입한다.
5) Deadlock-free 조건을 만족하도록 관계된 방정식을 푼다.

2.4 예제



〈 그림 2 〉 A Petri Net

그림 3의 PN의 minimal siphon은 $S_1=\{p_4, p_5\}$, $S_2=\{p_6, p_7\}$, $S_3=\{p_2, p_2, p_3\}$, $S_4=\{p_1, p_3, p_6\}$, $S_5=\{p_1, p_2, p_7\}$이다. S_1, S_2과 S_3는 또한 trap이다. S_4와 S_5는 trap을 포함하고 있지 않고 invariant-controlled이다. 간결하게 하기 위해, $m_i=M_0(pi)$라 하자.

$F(S_4)=0$은 상태 방정식을 만족하는 어떤 M에서 $M(p)=0\ \forall p\in S_4$과 $M(p)\geq 0\ \forall p\notin S_4$을 의미한다. 모든 $p\in S_4$에서 상태 방정식 $M(p)=0$을 만족하는 해는 다음으로 주어진다.

$y_1=m_1+m_6+2m_3+y_2,$

$y_3=m_6+m_3+y_2,$

$y_4=m_3+y_2.$

이들 결과를 남은 상태 방정식에 대입하면 다음이 된다.

$M(p2)=m_2+m_1+m_3,$

$M(p4)=m_4-m_1-m_6-2m_3,$

$M(p5)=m_5+m_1+m_6+2m_3,$

$M(p7)=m_7+m_6.$

따라서, $F(S_4)=0$ **iff** $m_4-m_1-m_6-2m_3\geq 0$ 또는 등가적으로 $F(S_4)>0$ **iff** $m_4-m_1-m_6-2m_3<0$이다.

유사하게, $F(S_5)>0$ iff $m_5-m_1-m_7-2m_2<0$를 얻을 수 있다. S_1, S_2와 S_3의 제약들과의 관계를 결합하면, 만약 $m_4+m_5>0$, $m_6+m_7>0$, $m_1+m_2+m_3>0$, $m_1+m_6+2m_3>m_4$, $m_1+m_7+2m_2>m_5$이면 PN은 deadlock-free이다.

2.4.1 프로그램 결과 비교

〈 그림 2 〉의 페트리 네트 예에 대하여, 여러 초기 마킹에서 Matlab을 이용하여 프로그램을 실행한 결과는 다음과 같다.

① $M_0=[1, 0, 0, 2, 1, 1, 1]^T$

Deadlock-freeness 조건을 만족하지 않는 초기 마킹이며, 이를 이용하여 공정을 시뮬레이션 한 결과 t_1 t_3 t_1의 순서로 점화할 경우 deadlock이 발생하였다.

② $M_0=[1, 0, 0, 0, 2, 1, 1]^T$

Deadlock-freeness 조건을 만족하지 않는 초기 마킹이며, 이를 이용하여 공정을 시뮬레이션 한 결과 t_2 t_4 t_2의 순서로 점화할 경우 deadlock이 발생하였다.

③ $M_0=[1, 0, 0, 0, 1, 1, 1]^T$

Deadlock-free 조건을 만족하는 초기 마킹으로서, 이를 이용하여 공정을 시뮬레이션 한 결과 deadlock이 발생하지 않고 30000번의 점화가 발생하였다.

3. 결 론

FMS's는 동시에 여러 가지 제품들을 처리하는 특징을 가진 시스템이며, 각 제품들은 각각의 제품 형태의 작업과정에 따라 정해진 순서를 가지고 machine들을 통해서 생산된다. 제품들은 작업 과정을 거치면서 이동을 하는 동안 machine, robot, 도구, 부품 등의 유한 자원을 경합하게 되며, 이러한 유한 자원의 경합은 deadlock 발생의 원인이 된다. 유한 자원의 수가 원인이 되는 deadlock 발생은 자원의 수를 변경함으로써 막을 수 있다. 따라서 본 논문에서는 페트리 네트의 특별한 구조인 siphon과 trap을 이용하여 초기에 자원의 수를 조절하여 deadlock 발생을 막는 방법을 제시하였으며, 또한 제시된 방법을 이용하여 초기 마킹을 하였을 경우와 그렇지 않은 경우를 Matlab 프로그램을 이용하여 검증하였다. 프로그램을 이용하여 검증한 결과 제시한 deadlock-free 조건에 만족하는 초기 마킹은 deadlock이 발생하지 않았으며, deadlock-free 조건에 만족하지 않은 초기 마킹의 경우 deadlock이 발생함을 확인하였다.

[참 고 문 헌]

[1] Feng Chu and Xiao-Lan Xie, "Deadlock Analysis of Petri Nets Using Siphons and Mathematical Programming", IEEE Tran. vol. 13, no. 6 pp793-804, 1997
[2] G. W. Brams, *Reseaux de Petri : Theorie et Pratique*, Masson, France, 1983
[3] T. Murata, "Petri nets : Properties, analysis and applications," in Proceedings of IEEE, 1989, vol. 77, pp. 541-580.
[4] Naiqi Wu "Necessary and Sufficient Conditions for deadlock-free Operation in Flexible Manufacturing System Using a Colored Petri Net Model" IEEE Trans. Systems, Man, And Cybernetics.. Vol. 29, No. 2 May 1999
[5] E. C. Yamalidou and J. C. Kantor, "Modeling and Optimal Control of Discrete-Event Chemical Processes using Petri Nets", Computers chem. Engng, Vol. 15, No. 7, pp. 503-519, 1991.
[6] J. L. Peterson, "Petri Net Theory and the Modelin g of Systems", Prentice Hall, 1981
[7] T. Murata, "Petri Nets : Properties, analysis and application", Proc. IEEE, vol.77, no.4, pp.541-579, 1989