

EMFG의 추적트리에 관한 연구

김정수*, 이태훈*, 여정모**

*부경대학교 전자계산학과

**부경대학교 전자컴퓨터정보통신공학부

e-mail:jayes21@mail.emfg.pe.kr

A Study on the Trace Tree of an EMFG

Jung-Soo Kim*, Tai-Hoon Lee*, Jeong-Mo Yeo**

*Dept of Computer Science, PuKyong National University

**Division of Electronics, Computer and Telecommunication Engineering, PuKyong National University

요약

이산 시스템을 모델링한 EMFG의 동작, 즉 마크 흐름을 분석한다는 것은 시스템의 동작을 파악하는 것이다. 본 연구에서는 EMFG의 과거동작을 파악할 수 있으며 시스템의 과거 이력을 판단할 수 있는 추적트리를 도입하고, 추적트리를 구할 수 있는 추적트리 알고리즘을 제안한다. 추적트리 알고리즘은 EMFG로 모델링된 이산 시스템의 과거동작을 파악할 수 있게하여 이산시스템의 고장진단이나 Deadlock과 같은 문제를 해결하는데 도움을 줄 것으로 기대된다.

1. 서론

EMFG(Extended Mark Flow Graph)[2]는 Petri Net[1]에서 파생된 그래프 이론으로서 이산 시스템의 모델링이나 분석에 유용한 도구로 잘 알려져 있다. EMFG로 모델링된 시스템은 마크의 흐름을 통해 시스템의 동작상태를 직관적으로 파악할 수 있고 이러한 마크의 변화는 수학적 해석이 가능하다. 그러나 마크의 변화가 없거나 특정상태의 마크벡터에 도달하지 않을 때는 시스템의 동작을 제대로 파악하기가 어렵다. 시스템의 올바른 설계를 위해 모델링된 EMFG에서 마크의 흐름, 즉 시스템 동작상태의 검증은 매우 중요한 부분이다.

본 논문에서는 EMFG로 모델링된 시스템에서 임의의 한 상태에 대한 마크벡터를 이용해 이전마크벡터를 알아내고 마크가 변화되어온 과정을 파악할 수 있는 추적트리를 도입하고, 추적트리를 구할 수 있는 추적트리 알고리즘을 제안한다. 본 논문의 구성은 먼저 EMFG의 기본 성질과 점화 동작에 대해 알아보고 추적트리에 대한 정의를 바탕으로 알고리즘에 필요한 행렬과 벡터들을 정의하여 추적트리 알고리즘을 서술한다. 마지막으로 본 논문에서 제안한 추적트리 알고리즘을 적용한 시스템의 예를 보이고 결론을 짓는다.

2. 본론

2.1 EMFG의 성질

EMFG는 m 개의 박스(box)와 n 개의 트랜지션(transition), 아크(arc)들로 구성되는 마크(mark)를 갖는 선도로 정의된다. 박스는 개념적인 상태를 나타내며 박스 안에 마크를 두어 상태의 만족 여부를 나타낸다. 트랜지션은 박스(들)의 상태가 변화하거나 다른 박스(들)의 상태를 변화시키는 곳으로 변화가 일어나는 과정을 트랜지션이 점화(fire)한다고 한다. 아크는 일반아크, 역아크, 조건아크가 있으며 박스와 트랜지션 혹은 트랜지션과 박스사이에서 트랜지션의 점화 조건을 결정하고, 트랜지션이 점화하면 박스의 마크 상태가 결정된다. 또한 트랜지션을 기준으로 트랜지션으로 들어오는 아크를 입력아크, 트랜지션에서 나가는 아크를 출력아크라하고 이들은 각각 입력행렬, 출력행렬로 나타낼 수 있다.[1~5]

2.2 EMFG의 점화 동작

EMFG에서 마크의 이동은 트랜지션(들)의 점화에 의해 이루어진다. 트랜지션이 점화하면 트랜지션에서 역아크로 연결된 박스의 마크는 소멸하고, 일반아크로 연결된 박스에는 마크가 생성된다. 이와 같

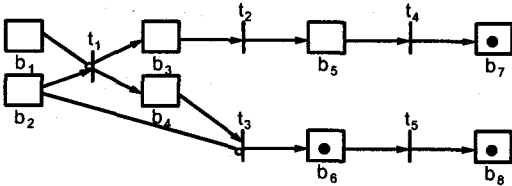
은 성질을 이용하면 마크의 흐름을 역으로 찾아갈 수 있다. 우선 마크가 존재하는 박스에 어떠한 트랜지션이 점화하여 마크가 생성되었는지를 알아낸 후, 이러한 트랜지션(들)이 점화하기 위한 박스의 상태를 구해서 이전상태의 마크벡터를 구하는 것이다.

정의 1. EMFG에서 트랜지션 t_i 의 출력일반아크(들)로 연결된 모든 박스(들)에는 마크가 존재하고 출력역아크(들)로 연결된 모든 박스(들)에는 마크가 존재하지 않을 경우 ' t_i 는 전점화가능하다'라하고, 이 때 t_i 를 '전점화가능 트랜지션'이라하며 이외의 트랜지션은 '전점화불가 트랜지션'이라 한다. ■

정의 2. EMFG에서 트랜지션(들)의 전점화가능 상태를 나타낸 벡터를 '전점화가능 트랜지션벡터 T_p '라 한다. ■

전점화가능 트랜지션벡터 T_p 의 원소가 1이면 해당 트랜지션이 전점화가능하고, 0이면 해당 트랜지션이 전점화불가하다. 예를 들어 (그림 1)에서 전점화가능 트랜지션벡터 T_p 는 (식 1)과 같이 표현되고, 트랜지션 t_3, t_4, t_5 의 전점화에 의해 마크가 박스 b_6, b_7, b_8 등에 대응되어 생성됨을 알 수 있다.

$$T_p = [00111] \quad (\text{식 1})$$



(그림 1) EMFG의 예

전점화가능 트랜지션은 현재마크벡터 M_k 와 트랜지션의 출력행렬 [6] T_o 에서 구한다. 출력행렬의 열은 각각의 트랜지션이 점화할 경우 마크가 생성되는 박스(들)을 나타낸다. 그러므로 전점화가능 트랜지션이 전점화해서 마크가 생성되는 박스는 현재 마크벡터이거나 현재 마크벡터에 포함되어야 한다.

EMFG에서 트랜지션이 점화하기 위해서는 점화조건이 만족되어야 한다. 점화조건은 트랜지션의 입력아크로 연결된 박스의 상태를 알아야 한다. 입력아크가 일반아크이면 박스에 마크가 존재해야하고, 역아크이면 박스에 마크가 없어야 트랜지션의 점화조건이 만족된다.

정의 3. 전점화가능 트랜지션의 점화가능조건과 전점화불가 트랜지션의 점화불가조건을 더한 결과를 전점화조건 C_i 라 한다. ■

전점화가능 트랜지션의 전점화조건은 점화조건행렬 [6]에서 구할 수 있다. 점화조건행렬의 열은 트랜지션의 점화조건을 만족시키는 박스의 상태를 의미한다. 그러므로 그림 1에서 전점화가능 트랜지션(들)에 대한 전점화가능조건을 다음 (표 1)과 같이 나타낸다. 트랜지션 t_3 이 전점화하기 위해서는 b_2 에는 마크가 없고, b_4 에는 마크가 있어야 하며 트랜지션 t_4 와 t_5 가 전점화하기 위해서는 b_5 와 b_6 에는 마크가 존재해야 한다.

(표 1) 전점화가능 트랜지션의 전점화가능조건

t_3	t_4	t_5
$b_2' \cdot b_4$	b_5	b_6

전점화불가 트랜지션의 전점화불가조건은 전점화불가 트랜지션의 전점화가능조건을 구한 후, 여기에 부정을 취하여 전점화불가조건을 구할 수 있고 (표 2)와 같이 나타낸다.

(표 2) 전점화불가 트랜지션의 전점화불가조건

t_1	t_2	t_1	t_2
$b_1' \cdot b_2$	b_3	$b_1' \cdot b_2'$	b_3'
		$b_1 \cdot b_2'$	
		$b_1 \cdot b_2$	

① 전점화가능조건

② 전점화불가조건

트랜지션 t_1 이 전점화하지 않기 위해서는 b_1 에 마크가 없고 b_2 에도 마크가 없는 경우와 b_1 에는 마크가 있고 b_2 에는 마크가 없는 경우, 마지막으로 b_1, b_2 모두 마크가 있는 경우이다. 트랜지션 t_2 는 b_3 에 마크가 없으면 전점화할 수 없다. 그리고 하나의 박스가 전점화가능 트랜지션의 전점화가능조건과 전점화불가트랜지션의 전점화불가조건에서 서로 다른 상태로 동시에 나타나는 경우 그 박스의 상태는 전점화가능 트랜지션의 전점화조건을 따른다. 이들 (표 1)과 (표 2)의 결과를 더하면 (표 3)과 같은 전점화조건을 구할 수 있다.

(표 3) 전점화조건

t_1	t_2	t_3	t_4	t_5
$b_1' \cdot b_2'$ $b_1 \cdot b_2'$	b_3'	$b_2' \cdot b_4$	b_5	b_6

전점화조건은 트랜지션이 전점화하기 위한 박스의

상태와 전정화할 수 없는 박스의 상태를 의미한다. 마크벡터 역시 박스의 상태를 나타내므로 전정화조건은 마크벡터로 변환이 가능하다. 전정화조건에서 구할 수 있는 모든 경우를 마크벡터로 변환하여 이를 마킹 상태 천이 방정식[7]을 이용하여 검증한다.

2.3 EMFG의 추적트리 알고리즘

EMFG의 현재 마크벡터에서 이전상태의 마크벡터를 찾고, 이러한 과정을 반복해서 구해지는 마크벡터들을 트리로 표현한다.

정의 4. EMFG에서 현재의 마크벡터 M_k 를 루트로 하고, 이를 이용해 구하여진 이전 상태가 될 수 있는 마크벡터 M_{k-1} (들)을 하위노드로 추가하여 구성된 트리를 추적트리(Trace Tree)라고 한다. ■

추적트리를 구하는 알고리즘은 다음과 같다.

단계 1 : 전정화가능 트랜지션벡터 T_p 를 구한다.

단계 2 : 전정화가능 트랜지션벡터 T_p 와 정화조건행렬 C 를 이용하여 전정화조건 C_t 를 구한다.

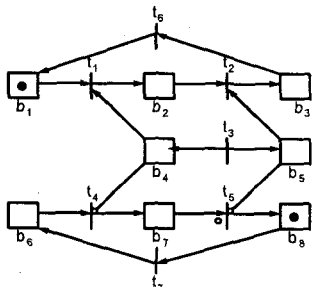
- ① 전정화가능 트랜지션의 전정화조건(들)을 구함
- ② 전정화불가 트랜지션의 전정화불가조건(들)을 구함
- ③ ①, ②의 결과를 더해서 전정화조건 C_t 를 구함

단계 3 : 단계 2의 결과 구해진 전정화조건 C_t 를 모두 마크벡터 M_{k-1} 로 변환한다.

단계 4 : 각각의 마크벡터 M_{k-1} 을 마킹상태 천이 방정식을 이용해서 현재의 마크 벡터 M_k 와 일치하는지 검사하고, 일치하면 추적트리의 노드로 추가하고 일치하지 않으면 제거한다.

단계 5 : 단계 1 ~ 단계 4의 과정을 전정화가능 트랜지션이 나타나지 않을 때까지 반복한다.

2.4 시스템 적용 예



(그림 2) EMFG로 표현된 시스템

위의 (그림 2)는 두 개의 공정 P1, P2에서 P1($b_1, b_2,$

b_3)은 자원(b_4, b_5)이 있어야 공정을 진행하고, P2(b_6, b_7, b_8)는 자원이 없어야 공정을 진행하는 시스템을 EMFG로 모델링한 것이다.

(그림 2)에 대한 추적트리는 다음과 같이 구할 수 있다.

단계 1 : 마크벡터 M_k 와 출력행렬 T_o 에서 전정화가능 트랜지션벡터 T_p 를 구한다.

$$M_k = [10000001] \quad T_o = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 7 & 0 & 1 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \end{bmatrix}$$

위의 마크벡터 M_k 와 출력행렬 T_o 에서 트랜지션 t_5 와 t_6 가 각각 전정화했을 때 마크가 생성되는 박스(b_1, b_6)가 현재 마크벡터에 포함되므로 전정화가능 트랜지션벡터 T_p 를 다음과 같이 구한다.

$$T_p = [0000110]$$

단계 2 : 전정화가능 트랜지션벡터 T_p 에서 전정화조건 C_t 를 구한다.

- ① (표 4)와 같이 전정화가능 트랜지션의 전정화조건을 구함

(표 4) 전정화가능 트랜지션의 전정화가능조건

t_5	t_6
$b_5' \cdot b_7$	b_3

- ② (표 5)와 같이 전정화불가 트랜지션의 전정화불가조건을 구함

(표 5) 전정화불가 트랜지션의 전정화불가조건

t_1	t_2	t_3	t_4	t_5
$b_1' \cdot b_4$	$b_2' \cdot b_5$	b_8	$b_4' \cdot b_6$	b_8

t_1	t_2	t_3	t_4	t_5
$b_1' \cdot b_4'$	$b_2' \cdot b_5'$	b_8'	$b_4' \cdot b_6'$	b_8'
$b_1' \cdot b_4$	$b_2' \cdot b_5$		$b_4' \cdot b_6$	
$b_1' \cdot b_4'$			$b_4' \cdot b_6'$	

- ③ (표 4)와 (표 5)에 의해서 전정화조건 C_t 를 다음 (표 6)과 같이 구함

(표 6) 전정화조건

t_1	t_2	t_3	t_4	t_5
$b_1' \cdot b_4'$	$b_2' \cdot b_5'$	b_8'	$b_4' \cdot b_6'$	b_8'
$b_1' \cdot b_4$	$b_2' \cdot b_5$		$b_4' \cdot b_6$	
$b_1' \cdot b_4'$			$b_4' \cdot b_6'$	

단계 3 : (표 6)에서 생성 가능한 전정화조건의 모든 경우를 다음과 같이 마크벡터 M_{k-1} 로 변환한다.

[00100010], [10100010], [01100010],
[11100010], [00110010], [01110010],
[00110010], [01110110]

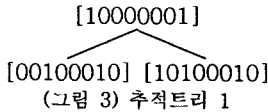
단계 4 : 마킹상태 천이 방정식을 이용하여 마크벡터를 검증하고 추적트리를 작성한다.

① 마킹상태 천이 방정식에 적용

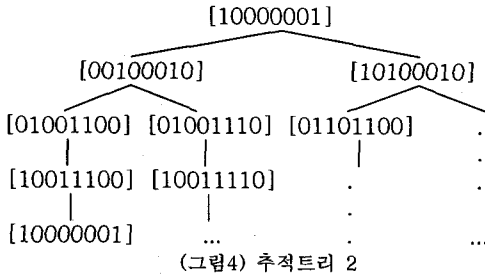
(표 7) 마킹상태 천이 방정식을 이용한 검증

M_{k-1}	M_k	만족 여부
[00100010]	[10000001]	추가
[10100010]	[10000001]	추가
[01100010]	[11000001]	제거
[11100010]	[11000001]	제거
[00110010]	[10010001]	제거
[01110010]	[11010001]	제거
[00110010]	[10010001]	제거
[01110110]	[11010101]	제거

② 추적트리의 작성



단계 5 : 위의 단계 1 ~ 단계 4의 과정을 반복해서 추적트리를 (그림 4)와 같이 나타낸다.



추적트리 알고리즘을 사용하여 추적트리를 구한 결과 현재 마크벡터에서 5스텝 이전상태를 구했을 때 현재의 마크벡터와 동일한 마크벡터가 트리에 나타나고 시스템이 순환 구조를 가지고 있는 것을 알 수 있다.

3. 결론

본 논문에서는 EMFG의 동작해석 알고리즘을 바탕으로 EMFG로 표현된 시스템의 이전상태를 찾아낼

수 있는 추적트리 알고리즘을 제안한다. 추적트리 알고리즘을 이용해 시스템이 어떤 상태에서 변화되어 왔는지를 추적트리로 표현하고, 마크의 변화를 파악할 수 있도록 한다. 또한 시스템의 동작을 역으로 찾고, 시스템의 현재상태가 되기 위한 이전 조건들을 파악할 수 있다. 이 알고리즘을 이용하면 시스템의 고장진단이나 Deadlock의 경우에 그 원인을 찾고 해결하는데 도움이 될 수 있다. 그리고 시스템의 특정한 상태를 정해 놓고 그 결과를 얻기 위한 시스템의 이전조건들을 찾을 수 있기 때문에 시스템의 성능을 향상시키는 면에서 도움을 줄 것으로 기대된다. 향후 시간 트랜지션과 조건아크를 포함하는 EMFG에서 적용 가능하도록 연구를 진행할 것이다.

참고문헌

[1] Tadao Murata, "Petri Nets : Properties, Analysis and Applications", Proceedings of the IEEE, Vol. 77, No. 4, p541-580, 1989
 [2] 여정모, "EMFG 회로의 간략화에 관한 연구", 부산개방대학 연구보고 Vol. 29, p741-760 1987. 12 .
 [3] 여정모, "이산제어시스템 설계를 위한 확장된 마크흐름선도의 동작해석", 정보처리논문지 Vol. 5 No. 7, p. 1896-1907, 1998. 7.
 [4] 여정모, "이산 시스템의 설계와 해석을 위한 확장된 마크흐름선도의 재정의를 회로변환", 멀티미디어학회 논문지 Vol. 1 No. 2, p224 - 238, 1998. 12.
 [5] 김희정, 백형구, 김종민, 여정모, "이산시스템 설계를 위한 확장된 마크흐름선도의 수학적 해석", 한국해양정보통신학회 춘계종합학술대회. Vol. 5. No. 1, p p.692-695, 2001. 5.
 [6] 김희정 외 3인 "접속 행렬을 이용한 EMFG의 수학적 해석", 한국멀티미디어학회, 추계학술발표논문집 Vol. 4. No. 2, pp815-820, 2001. 11
 [7] 김희정, 서경룡, 여정모 "EMFG의 개선된 동작해석 알고리즘", 한국정보처리학회 논문지A Vol. 9. No. 3, pp371-378, 2002. 9