

Petri Net을 이용한 시간논리 구조의 표현

김정철* 모영승 김진권 황형수
원광대학교 공과대학 제어계측공학과

Representation of Temporal Logic Framework Using Petri Net

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

Abstract - Temporal Logic Frameworks is convenient to represent temporal relation. It is useful to represent a dynamic properties of Discrete Event Dynamic Systems. Also, it is convenient to express a current and next state of event using logical representation. Because the teachability tree of the Temporal Logic Frameworks is very complexity it is difficult to understand. In this paper, we defined some rules to represent Temporal Logic Frameworks by Petri Net and proposed am method of the representation of them Petri Net for the Temporal Logic Frameworks.

An example are used to demonstrate the feasibility of this method.

1. 서 론

생산 시스템, 교통 시스템, 일괄처리, 통신 시스템, 전문가 시스템 등과 같은 대규모의 동적 시스템들은 이산사건 시스템 구조로 이루어져 있다. [4] 이러한 시스템들은 기존의 제어 및 시스템 이론으로 처리될 수 없고 이산적인 상태와 이 상태를 사이에서 사건의 천이관계를 나타냄으로써 표현할 수 있다. 이런 시스템들의 모델링과 제어에 시간논리 구조와 패트리 네트를 이용한 연구가 많이 진행되고 있다. 하지만 시간논리 구조나 패트리 네트의 이산사건 시스템을 모델링 하는 방법은 완벽하게 이산사건 시스템을 표현함에 있어 문제점을 가지고 있다. 시간논리구조는 시간관계를 논리적으로 표현함으로써 사건의 전후관계와 다음 상태, 최종상태 등을 표현하기에 편리하다. 하지만 시각적인 면에서 패트리 네트에 뒤지며 패트리 네트는 논리식 대신에 그래프 적인 표현으로 시각적인 효과는 아주 뛰어난 특성을 가지고 있다. 따라서 본 논문은 시간논리 구조의 복잡한 논리식과 시각적인 이해가 어렵다는 단점을 패트리 네트로 보완하는 방법을 제시한다.

2. 본 론

2.1 시간논리구조

시간논리(Temporal Logic)는 물리적 또는 프로그램에서의 상태가 시간에 따라 변하는 궤적을 수식적으로 표현하려는 방법으로 제안되었다. 시간논리구조는 시간 개념을 포함한 술어논리의 확장된 형태이며, 시간 순차 열에 따른 추론 지향적인 논리이다.

2.1.1. 시간논리구조의 기호와 표현

시간논리구조는 시간과 시간에 대한 추론을 할 수 있는 형식구조이다. 시간논리구조는 종래의 Boolean 연결자인 \neg (not), \wedge (and), \vee (or), \rightarrow (implies), \leftrightarrow (if and only if)를 포함하며 시간의 변화와 시간의 양을

표현하기 위한 시제 연산자 \square (henceforth), \diamond (eventually), \bigcirc (next), U (until), P (proceed) 와 같은 연결자와 연산자를 이용하여 시간과 시제관계에 관한 추론이 가능한 구조이다. 이들 연산자들의 정의는 다음과 같다. [1][2][3]

< 정의 1 > 부울 연산자와 시간 연산자

• 부울 논리 연산자

- (1) $(\neg\phi)$: ϕ 는 참이 아니다.
- (2) $(\phi\wedge\psi)$: ϕ 가 참이고 ψ 도 참이다.
- (3) $(\phi\vee\psi)$: ϕ 가 참이거나 ψ 가 참이다.
- (4) $(\phi\rightarrow\psi)$: ϕ 가 참이면 ψ 가 참이다.
- (5) $(\phi\leftrightarrow\psi)$: ϕ 가 참이면 ψ 가 참이고 ψ 가 참이면 ϕ 가 참이다.

• Temporal 연산자

- (1) $(\square\phi)$: 지금부터 ϕ 는 참이다.
- (2) $(\diamond\phi)$: 미래에 ϕ 가 참이 되는 시점이 있다.
- (3) $(\bigcirc\phi)$: 다음 번에 ϕ 는 참이 된다.
- (4) $(\phi U\psi)$: 현재 ϕ 가 참이고, 미래에 ψ 가 참이 되는 점이 존재한다면, 그때까지 ϕ 는 참일 것이다.
- (5) $(\phi P\psi)$: ψ 가 참이기 전에 ϕ 는 참이어야 한다.

• 연산자의 전형적인 축약형

- (1) 조건문 : $\phi \rightarrow \psi \equiv (\neg\phi \vee \psi)$
- (2) 동위 접속 연산자 : $\phi \wedge \psi \equiv (\neg(\neg\phi \vee \neg\psi))$
- (3) 쌍 조건문 : $\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$
- (4) eventually 연산자 : $(\diamond\phi) \equiv (\neg(\square(\neg\phi)))$
- (5) precedes 연산자 : $(\phi P\psi) \equiv (\neg((\neg\phi)U\psi))$

2.2 패트리 네트

패트리(Carl Adam Petri)가 1962년 이산사건 시스템의 해석으로 패트리 네트를 제시한 이래 많은 분야에서 응용되고 있다. 패트리 네트는 상호 작용하는 동시 발생 구성요소를 갖는 이산사건 시스템을 모델링하고 설계할 수 있는 그래프 이론적이며 시각적인 도구이다.

2.2.1 패트리 네트의 기호와 표현

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

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

여기서, $P = \{P_1, P_2, \dots, P_m\}$ 는 유한한 플레이스의 집합이며, $T = \{t_1, t_2, \dots, t_n\}$ 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)

2.2.2 패트리 넷의 실행규칙

- (1) 활성화 규칙 : 트랜지션의 모든 입력 플레이스에 arc수 보다 크거나 같은 토큰이 있을 때, 트랜지션은 점화가능 하다.
(2) 점화 규칙 : 활성화된 트랜지션이 점화하면, 입력 플레이스에서 arc수만큼의 토큰이 제거되고, 출력 플레이스에 출력 arc수만큼의 토큰이 더해진다.

2.3 패트리 넷을 이용한 시간논리구조의 표현방법

시간논리구조의 연산자는 단일 연산자와 이진 연산자가 있으며 다음과 같이 나타낼 수 있다.

단일 연산자 : 연산자 A

ex) $\neg A, \Box A$

이진 연산자 : A 연산자 B

ex) $A \wedge B, A \rightarrow B$

이와 같이 표현되는 시간논리구조는 다음 방법에 의해 패트리 넷으로 표현된다.

- (1) 피연산자 A와 B는 플레이스로 표현된다.
- (2) 플레이스는 토큰의 유무로 참, 거짓이 구분된다.
- (3) 트랜지션은 활성화가능성으로 참, 거짓이 구분된다.
- (4) 최종 결과는 하나의 트랜지션으로 표현한다.

〈정의 2〉 부울 논리 연산자는 다음과 같이 패트리 넷으로 표현한다.

- (1) ($\neg\phi$)

ϕ	$\neg\phi$
T	F
F	T

[표 1] ($\neg\phi$)의 진리표

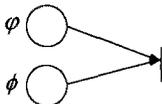


[그림 1] ($\neg\phi$)의 패트리 넷 표현

- (2) ($\phi \wedge \psi$)

ϕ	ψ	$\phi \wedge \psi$
T	T	T
T	F	F
F	T	F
F	F	F

[표 2] ($\phi \wedge \psi$)의 진리표

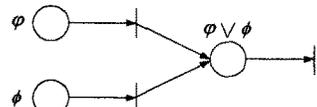


[그림 2] ($\phi \wedge \psi$)의 패트리 넷 표현

- (3) ($\phi \vee \psi$)

ϕ	ψ	$\phi \vee \psi$
T	T	T
T	F	T
F	T	T
F	F	F

[표 3] ($\phi \vee \psi$)의 진리표

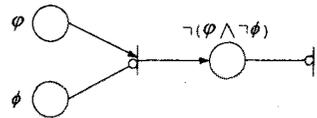


[그림 3] ($\phi \vee \psi$)의 패트리 넷 표현

- (4) ($\phi \rightarrow \psi$) \equiv ($\neg\phi \vee \psi$)

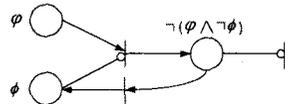
ϕ	ψ	$(\phi \rightarrow \psi) \equiv (\neg\phi \vee \psi)$ $\equiv \neg(\phi \wedge \neg\psi)$
T	T	T
T	F	F
F	T	T
F	F	T

[표 4] ($\phi \rightarrow \psi$)의 진리표



[그림 4] ($\phi \rightarrow \psi$)의 패트리 넷 표현

이 표현에는 고려해야 할 부분이 있다. 시스템을 시간논리구조로 표현할 때 $\Box[\phi \rightarrow \psi]$ 또는 $\Box[\phi \wedge \gamma \rightarrow \psi]$ 의 형태로 많은 부분이 표현된다. 이 표현에 대해 위와 같은 방법을 사용하여 패트리 넷으로 표현을 하면 $\neg(\phi \wedge \psi)$ 에 해당되는 플레이스에 토큰이 생성되는 사건이 발생하면 Deadlock이 발생한다. 이곳에 토큰이 생성되는 경우를 보면, 연산자의 입력측($\phi, \phi \wedge \gamma$)이 참일 때 출력측(ψ)이 거짓인 경우이다. 하지만, 시스템에서 입력측이 참일 경우 출력측은 다음 상태에서 참이 되므로 이 경우에 의해 생성된 토큰을 출력측의 입력으로 넣어줌으로써 Deadlock을 해결할 수 있다. 즉 $\phi \rightarrow \psi$ 는 다음과 같이 표현된다.

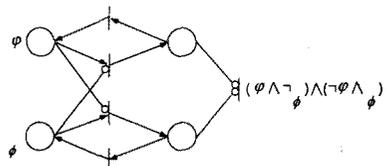


[그림 5] ($\phi \rightarrow \psi$)의 개선된 패트리 넷 표현

- (5) ($\phi \leftrightarrow \psi$) \equiv ($\phi \rightarrow \psi$) \wedge ($\psi \rightarrow \phi$)

ϕ	ψ	$(\phi \leftrightarrow \psi) \equiv (\phi \rightarrow \psi)(\psi \rightarrow \phi)$
T	T	T
T	F	F
F	T	F
F	F	T

[표 5] ($\phi \leftrightarrow \psi$)의 진리표



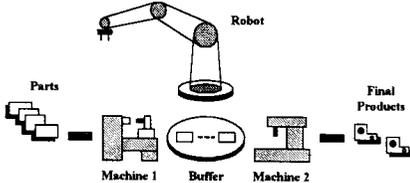
[그림 6] ($\phi \leftrightarrow \psi$)의 패트리 넷 표현

Temporal 연산자는 진리표로 구성을 할 수 없으며, 이를 패트리 넷으로 표현해야 할 경우 시스템에 대한 지식과 시간적 순서를 고려하여 플레이스와 트랜지션을 적절히 삽입함으로써 패트리 넷으로 표현할 수 있다.

2.4 적용 예 : 자동화 생산 시스템

(그림 7)은 공유자원을 가지고 있는 생산 시스템(manufacturing systems)이다. 두 개의 Machine이 직렬로 연결되어 있으며 그 사이에 중간 부품을 저장하기 위한 버퍼가 있고, 두 개의 Machine에 부품을 loading과 unloading하기 위한 Robot이 있다.

최종 생산품은 Machine1과 Machine2에 의한 부품처리에 의해 생성되며, Robot은 Machine들에 loading과 unloading할 때 다른 일을 할 수 없다. 또한 부품들은 Pallet에 실려서 이동한다.



(그림 7) 공유자원을 가진 생산시스템

생산 처리 과정을 시간논리구조로 표현하기 위해 다음과 같은 표기를 사용한다.

- PA : Pallet의 사용가능성을 나타낸다.
- M₁A : machine1의 사용가능성을 나타낸다.
- M₂A : machine2의 사용가능성을 나타낸다.
- RA : Robot의 사용가능성을 나타낸다.
- BA : Buffer의 사용가능성을 나타낸다.
- STM₁ : machine1에서 처리 시작되어 처리되고 있는 상태를 나타낸다.
- FNM₁ : machine1에서 처리 끝을 나타낸다.
- STM₂ : machine2에서 처리 시작되어 처리되고 있는 상태를 나타낸다.
- FNM₂ : machine2에서 처리 끝을 나타낸다.
- STOB : Buffer에 저장하고 있는 상태를 나타낸다.
- BS : Buffer에 저장된 상태를 나타낸다.

(그림 7)의 제조공정 시스템을 시간논리구조로 모델링을 하면 다음과 같은 구조를 갖는다.

- [PA(x) ∧ M₁A(x) → STM₁(○x)]
- [STM₁(x) U FNM₁(x)]
- [FNM₁(x) ∧ RA(x) ∧ BA(x) → STOB(○x)]
- [STOB(x) U BS(x)]
- [BS(x) ∧ RA(x) ∧ M₂A(x) → STM₂(○x)]
- [STM₂(x) U FNM₂(x)]
- [FNM₂(x) → PA(○x)]

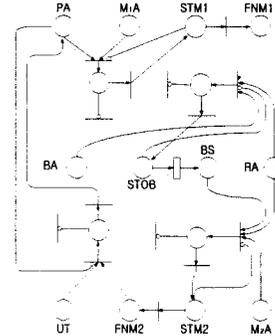
위와 같은 시간논리구조를 <정의2>를 사용하여 패트리 넷으로 표현하면 [그림 8]과 같다.

[그림 8]에서, 사용하지 않는 플레이스와 트랜지션을 제거하고 토큰 생성경로를 표현하면 [그림 9]와 같다.

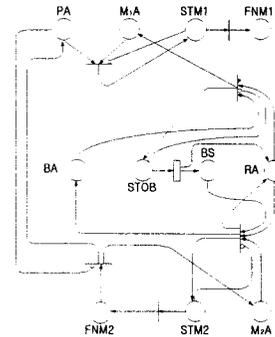
3. 결 론

시간논리구조는 이산사건 시스템의 동작을 나타내기 위해 용이하고 시간관계를 논리적으로 표현함으로써 사건의 전후관계와 다음 상태, 최종상태 등을 구현하기에 편리하다. 그러나, 시간논리구조의 도달성트리 구성은 매우 복잡하기 때문에 도달성트리를 이용하여 모델링된 시스템을 해석하는 것이 매우 어렵다. 본 논문에서는 시각적 표현이 우수한 패트리 넷을 이용하여 시간논리구조를

표현하기 위해 기본적인 규칙들을 정의하고 새로운 방법을 제안하였다. 이 방법은 기존의 도달성트리 구성보다 매우 우수한 특성을 보임을 한 예를 이용하여 확인하였다.



(그림 8) 시간논리구조의 패트리 넷표현



(그림 9) 토큰생성경로를 고려한 표현

(참 고 문 헌)

- [1] Dan Ionescu and Jing-Yue Lin, "Optimal Supervision of Discrete Event Systems in A Temporal Logic Framework", IEEE Trans. on Systems, Man, and Cybernetics, Vol. 25, No. 12, Dec. pp. 1595-1605, 1995.
- [2] A. Ganz, W. B. Gong, C. M. Krishna and W. Zhai, "Reconfiguration Algorithm for Ring Networks", IEEE Proceedings of the 31st Conference on Decision and Control Tucson, Arizons, December, pp. 3221-3226, 1992.
- [3] H. S. Hwang, S. C. Joo and D. Ionescu, "The Controller modeling using the temporal logic model in Discrete Event Dynamic Systems", Journal of the KISS, Vol. 21, No. 9, pp. 1665-1674, 1994
- [4] E. C. Yamalidou, E. P. Patsidou and J. C. Kantor, "Modeling Discrete- Event Dynamical Systems for Chemical Process Control - A Survey of Several New Techniques", Computers Chem. Engng, Vol. 14, No. 3, pp. 281-299, 1990.
- [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