시간적 오토마타에서 도달성 그래프의 도출기법
(A Method for Reachability Graph Derivation from Timed Automata)

박 찬 민 † 황 익 순 †† 김 태 형 †† 이 재 용 ††† 이 상 배 ††††
(Chan-Min Park)(Ik-Soon Hwang)(Tae-Hyong Kim)(Jai-Yong Lee)(Sang-Bae Lee)

요 약 프로토콜 검증에 있어서 reachability analysis를 사용하는 방법은 널리 알려지고 많이 사용되는 방법이다. Reachability analysis를 통한 검증 방법에 있어서는 프로토콜의 동작을 나타내는 reachability graph의 생성이 중요하다. 그러나 시간이 관련되는 경우 발생하는 상태 수 폭증(state explosion) 또는 시간의 실수적 성질의 문제가 있어서 reachability graph의 생성에 큰 장애가 되어왔다. 실시간 시스템이나 프로토콜에 있어서, 타이머는 대기 시간의 한계를 지정하거나, 일정한 시간간격을 시스템에 알릴 수 있도록 하는 등 시간의 상대적 기준점으로 사용되며 이외에도 여러 가지 용도로 타이머를 사용하게 되어 프로토콜을 모델링하는데 있어서 타이머는 불가결하다. 본 논문에서는 타이머를 모델링하지 못하며, 통신 프로토콜과 그 환경을 모델링하는데 적절치 못하던 기존의 방법과 달리, 타이머를 포함하며 통신 프로토콜과 그 환경을 적절히 표현할 수 있도록 시스템을 기술하는 방법과 함께 상대적 시간 개념으로 천이와 타이머를 다루는 방법을 통해서 시간 개념과 타이머가 들어있는 프로토콜의 reachability graph를 유도하는 방법을 제안하였다. 제안한 방법을 통해서 기존 방법으로는 찾아낼 수 없었던 시스템의 오류를 찾아낼 수 있었다. 이를 통해 보다 실제 프로토콜에 가깝게 모델링하고 검증할 수 있는 기반을 마련하고 있다.

Abstract One of the most famous and widely used methods in protocol verification is reachability analysis. In reachability analysis, it is important to derive reachability graph which describes the behavior of a protocol. However, in the case of concerning time, there have been many obstacles to derive reachability graph due to state explosion problem and the infinite precision and length of time. In real-time system and protocol, timer can specify the bound for waiting time, or indicate a certain time interval, so that timer can be seen as a relative base point for time measure. Moreover, timer is used for so many other purposes that it would be troublesome to model the protocols without timer. In this paper, we propose a method to derive a reachability graph from protocol with timed transitions and timers by handling transitions and timers with the relative time concept. The proposed method is capable of modeling timers which was impossible by the existing methods. And we also propose a system description technique which can describe protocols with timer and its circumstances more precisely than the existing methods. With the method we would find an error which was not detected by existing method. This method can give base for realistic modeling and powerful verification of real protocol.

· 본 연구는 한국과학재단 특정기초연구(1999-2-303-005-3)지원으로 수행되었음.
† 비 회 원 : 한국통신프리텔 기술연구소 연구원
pirotesa@persmail.com
†† 비 회 원 : 연세대학교 전자공학과
his@nasla.yonsei.ac.kr
thkim@nasla.yonsei.ac.kr
††† 종신회원 : 연세대학교 전자공학과 교수
jyl@nasla.yonsei.ac.kr
†††† 비 회 원 : 연세대학교 전기·전자공학과 자문교수
sblee@nasla.yonsei.ac.kr
논문접수 : 1999년 2월 12일
심사완료 : 2000년 10월 19일

1. 서 론

시스템이나 프로토콜의 개발 과정 가운데서 설계한 시스템이나 프로토콜이 올바른가를 시험하는 것이 검증(verification)이다[1]. 이러한 프로토콜 검증 방법에는 temporal logic을 사용하여 시스템이 가져야할 특성을 표현하고 시스템도 또한 temporal logic으로 표현한 후, 시스템이 특성을 만족하는가를 증명하는 방법과 시스템을 state machine으로 표현한 후 상태 영역(state space)를 탐색하여 시스템의 동작을 살피는 reachabi-

lity analysis를 사용한 방법이 있다.

그러나 reachability analysis를 이용한 기존의 방법들은 타이머를 표현하지 못하였으며, 통신 프로토콜과 그 환경을 모델링하기에 적절치 못하였다. 본 논문에선 기존의 방법들과는 달리, 타이머를 포함하며 통신 프로토콜을 적절히 표현할 수 있도록 시스템을 기술하는 방법과 함께 타이머를 시간의 상대적 기준으로 보고, 천이간 선후 관계를 상대적 시간으로 판단하는 방법으로 reachability graph를 유도하는 방법을 보이고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 연구의 배경과 기존의 연구에 대해서 기술하고, 3장에서는 제안하는 시스템 기술 방법과 reachability graph를 그리기 위한 알고리즘에 대해서 설명한다. 4장에서는 제안하는 방법을 적용해본 결과에 대해 보이고, 5장에서 결론을 맺는다.

2. 연구배경

위에서 구분한 바와 같이 시간 개념이 들어간 실시간 시스템 검증 방법은 대체로 두 가지 방법으로 나뉜다. Temporal logic에 시간 영역의 개념을 추가해 사용하여 주요 특성을 기술하고 이를 시스템이 만족하는가를 따지는 방법들[2][3][4]과 reachability analysis 방법을 사용하여 reachability tree나 reachability graph를 그리고 graph에서 이상한 행동이나 deadlock등을 밝혀내는 방법들[5][6][7]로 나뉜다.

이 중 reachability tree나 reachability graph를 유도하여 검증하는 방법의 경우 시간이 가지는 실수적 성질, 즉 무한의 길이와 무한의 정밀도를 가지는 것에 의해서 시스템이 가질 수 있는 행동의 가짓수 자체가 무한함에 따라 일어나는 상태수 폭증의 문제가 있고, tree가 아닌 graph화하는 문제는 동일한 상태를 묶는 것에 대한 문제를 가진다[7].

프로토콜 내에서 타이머는 기본적으로 재전송을 위해 쓰여 왔다. 현재의 프로토콜들에서는 타이머의 역할과 사용되는 타이머의 수가 과거의 프로토콜들에 비해서 점차 늘어나고 있고, 그 사용방법도 traffic control등이나 대역폭 효율 등을 위해서 교묘해지고 있다.[8] 따라서, 프로토콜의 기술함에 있어서 타이머를 제외할 수 없다.

기존 연구중 [5]에서는 process algebra에 시간 개념을 포함시킨 AORTA (Application Oriented Real Time Algebra)라는 표현 방식을 가지고 프로토콜을 기술하고, AORTA로 기술된 프로토콜에서 reachability graph를 도출하는 방법을 사용하고 있다. 기본적으로 LTS(labelled transition system)[9]에 시간 개념을 더

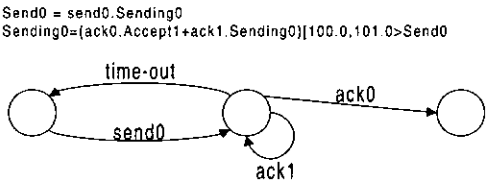

그림 1 기존 연구에서의 타이머 사용 예

한 것과 비슷한 형태의 천이(transition)를 기반으로 정의되는 machine을 기본 모델로 한다. 그리고 각각의 machine간 메시지를 주고받는 것은 LTS나 process algebra에서 쓰이는 방식인 입출력을 같게 하여 동기적으로 이루어지는 것으로 가정한다. Timer가 관련된 부분의 예제를 보면 그림 1과 같다.

[100.0,101.0>의 앞부분에 타이머가 작동하고 있는 동안에 행해질 수 있는 동작을 보이고 있다. 타이머가 작동 중의 가능한 모든 동작을 지정하지 않는 한 타이머가 동작하는 것과 동시에 다른 입출력이 이루어질 수 없는 모델링 방법이다. 상당히 타이머의 행동 자체가 제한적으로 정의되고, 사용되고 있다고 할 수 있다.

[6]에서는 프로토콜이 작동하는 모습을 통신하는 단말인 temporal communicating machine을 메시지 전달을 위한 선입선출(FIFO, First In First Out) queue와 함께 사용하여 temporal communicating system으로 모델링하고 있다. 함께 쓰이는 FIFO queue는 메시지 전달에 있어서의 최소 지연과 최대 지연을 정의하고 있어서 action을 하나의 machine으로부터 system 내의 다른 machine으로 정의된 시간을 지나 보내고 전달하게 된다.

[6]에서 문제점은 우선 타이머에 대한 개념이 없다는 점이다. 천이의 시간을 이용해서 타이머를 모델링 가능하지만, [5]의 경우와 마찬가지로 타이머가 여러개의 상태에 걸쳐서 지속되는 것을 모델링 할 수 없다. 또한 system을 구성하는 경우에 있어서 3개 이상의 machine으로 구성되는 경우에 대한 정의가 없다는 점도 문제가 된다고 할 수 있다. 그리고, 전송로가 단순히 한쪽에서 반대쪽으로 데이타를 보내기만 하는 것이 아니라, 중간에서 데이타를 잃어버리거나, 변형을 가할 수 있게 될 경우에 대해서 모델링이 불가능한 점이 있다.

3. 제안 방법

3.1 제안하는 시스템 기술 방법

우선 각각의 행동을 하며 주위에 연결된 machine들과 메시지를 주고받는 machine들이 모여서 하나의 시

스템을 구성하는 것으로 가정할 수 있다. 그리고, 시스템은 시스템 외부와 메시지 교환을 하지 않는 것으로 가정한다. 시스템을 구성하는 각각의 machine을 서로 통신하면서 그 내부에서 수행되는 천이가 시간적 요소를 지니며, 타이머를 가지는 CTFSM (Communicating Temporal FSM)으로 모델링하고 CTFSM을 다음과 같이 정의한다.

정의 1 CTFSM F는 다음과 같은 6-tuple F={S, M, A, T, Tr, s0}로 정의된다.

S : 상태들의 집합.

M : action을 주고 받는 인근 Communicating Temporal FSM. NULL ∈ M

T : 타이머들의 집합.

A : action들의 집합. T⊆A Time out 입력 action으로서 포함된다. NULL ∈ A

Tr : 천이들의 집합.

$Tr \subseteq S \times A \times M \times 2^{A' M} \times 2^{P} \times R \times R \times S$

P : 타이머에 대한 operation들.

SET(t), CLEAR(t), RESET(t). t∈T.

R : 시간을 나타내기 위한 실수 값.

s0 : 초기 상태.

CTFSM이 취할 수 있는 입력과 출력, action들 중에는 NULL이 있을 수 있다. 천이의 입력에 NULL이 올 경우, 해당 천이의 시작 상태에 들어서는 순간 선택되는 자발적인(spontaneous) 천이로 정의되는 경우이다. 그리고, 천이의 출력에 NULL이 올 경우는 단지 입력 action을 받은 후 아무런 action을 취하지 않고 천이만 일어나는 경우이다.

천이는 입력 상태와 출력 상태를 가진다. 입력 상태에 도달한 후 해당하는 입력 action이 정해진 인근 CTFSM으로부터 도달하면, 천이가 일어나게 된다. 천이에는 또한 천이에 필요한 시간의 최소값과 최대값이 정해져 있어, 해당하는 입력을 받은 후 최소한 최소값의 시간이 흐른 후, 최대값의 시간이 흐르기 전에 천이가 끝나고, 출력 상태로 천이를 하며, 정해진 1개 이상의 (NULL action도 하나의 action으로 본다.) 출력 action을 정해진 인근 CTFSM으로 전하거나 action이 전해질 CTFSM이 NULL로 정의되어있다면, 내부 action으로 마친다. 그리고, 정의가 되어있다면, 타이머들에 대한 operation, SET, RESET, CLEAR등도 행해진다. 다시 말하자면, 천이는 현재 상태와 입력 action과 그 action을 전해주는 인근 machine의 곱집합을 정의역으로 하여 출력 action과 그 출력 action을 전해주는 인근 machine, 타이머에 대한 operation, 천이가 수행되는 데 걸리는 최소시간과 최대시간, 그리고 다음 상태의 곱집합을 공역으로 갖는 함수로 정의된다. 정리하자면, 천이는 시작 상태, 입력 action, 입력 action를 준 인근 CTFSM, 출력 action(들)과 출력 action을 줄 인근 CTFSM(들), 타이머들에 대한 조작과 천이에 걸리는 시간의 최소, 최대 값, 그리고 천이 후 상태로 구성된다.

앞서 언급한 것과 같이 서로 연결된 CTFSM의 집합으로 이루어지는 시스템을 전체 시스템으로 가정하며, 이 시스템을 Temporal communicating system(TCS)으로 지칭하고, TCS를 다음과 같이 정의한다.

정의 2 TCS G는 다음과 같은 2-tuple G={M, C}로 정의된다.

M : 구성하고 있는 CTFSM의 집합.

C : CTFSM간의 연결 상태들의 집합.

$(M \times M) \in C$

이렇게 정의된 TCS의 작동하는 모습을 나타내는 것이 전체 시스템에 대한 reachability graph가 되겠다. Reachability graph의 node는 시스템의 상태, edge는 시스템 내에서의 천이로 이루어진다.

이러한 시간과 연결된 시스템을 기술하는 방법으로는 많이 쓰이는 Petri-net 기법이 있다. Petri-net 기법의 경우도 시간을 적용하는 경우에 대한 연구가 많이 되어, Firing Time, Enable Time, Holding Time등 많은 시간을 모델링하는 기법들이 있다.[11] Enable Time이나 Holding Time으로 시스템 시간을 모델링할 경우 시스템적으로는 본 제안과 비슷한 자유도의 기술이 가능하나, Petri-net으로 모델링시에는 시스템의 세부 모듈들이 서로 어떻게 액션을 주고 받는가를 기술자가 모두 계산해서 전체 시스템을 기술하여야 하는 단점이 있으나, 본 논문에서 제안하는 TCS의 경우 각 모듈의 FSM과 각 모듈간에 연결상태만을 정의함으로서 전체 시스템의 동작을 고려하지 않고, 세부 모듈만을 고려하여 시스템을 기술할 수 있다.

3.2 제안하는 reachability graph 도출 기법

Reachability graph는 초기 상태로부터 시스템이 도달할 수 있는 상태들과 천이들을 보인 그래프로 시스템의 행동을 보인다. 앞서 지적한 것과 마찬가지로 시간 개념이 개입함에 의해서 생기는 무한한 가능성이 생기므로 이를 해결하여야 한다. 천이가 일어날 수 있는 시간 범위를 정하여 그 사이에 일어나는 천이를 동일한 천이로 취급하여, 천이가 일어나는 시간에 따라 일어나는 상태 수 증가를 줄이려 하였다. 동일 상태의 기준으로는 각 상태에서의 이후 관찰 가능한 action들이 동일한가를 따지는 weak sense bisimulation 기준으로 하

였다. weak sense bisimulation은 다음과 같이 정의된다[10].

정의 3 어떤 두 상태 A, B의 관계는 시스템에서 관찰 가능한 모든 action a에 대해서 다음을 만족할 때 (weak) bisimulation 관계이다.

1) $A \xrightarrow{a} A'$면, 어떤 B'에 대해서 $B \stackrel{a}{\Rightarrow} B'$이고, A'와 B'도 (weak) bisimulation 관계이다.

2) $B \xrightarrow{a} B'$면, 어떤 A'에 대해서 $A \stackrel{a}{\Rightarrow} A'$이고, A'와 B'도 (weak) bisimulation 관계이다.

따라서, 두 상태에서 관찰 가능한 action들이 동일하려면, 두 상태에서 수행 가능한 천이와 그 가능한 순서가 동일하여야 한다. 따라서 다음과 같이 상태와 동일 상태를 정의하였다.

정의 4 TCS G={M, C}의 reachability graph의 상태 F는 다음과 같은 3-tuple={S, Tr, Ti}로 정의된다.

S : M내의 각 CTFSM의 상태.

Tr : 입력 action을 받아서 현재, 수행되어 출력 action을 내기를 기다리는 천이와 그 대기 시간 쌍들의 집합. {(tr, a, b), ... }

tr : TCS를 구성하는 CTFSM 중 하나에 속한 천이.

a, b : 천이가 수행 가능한 최소값과 최대값

Ti : Active한 타이머들과 그 시간 범위 쌍들의 집합. {(t, a, b), ... }

t : TCS를 구성하는 CTFSM 내에 선언된 타이머.

a, b : 타이머가 가질 수 있는 시간값의 최소값과 최대값

정의 5 Reachability graph 내의 두 상태 A={ S_A, Tr_A, Ti_A}, B={ S_B, Tr_B, Ti_B}가 다음의 조건들을 만족한다면, A와 B는 동일 상태이다.

1) $S_A = S_B$.

2) Tr_A와 Tr_B가 각각 {..., $(tr_{a_i}, a_{a_i}, b_{a_i})$, ...}과 {..., $(tr_{b_i}, a_{b_i}, b_{b_i})$, ...}와 같이 구성되고, 각각 a_{a_i}와 b_{a_i}를 기준으로 정렬되어 있다고 가정할 때, 모든 i에 대해서 $tr_{a_i} = tr_{b_i}$.

3) Tr_A와 Tr_B의 $a_{a_i}(a_{b_i})$와 $b_{a_i}(b_{b_i})$들을 모아서 순서대로 나열하였을 때, 그 순서가 일치한다.

4) Active한 타이머가 있을 경우 가장 천이 시간이 늦은 천이와 타이머의 time out 시간과의 관계가 동일하다. 이 때, 시간 관계는 time out 전에 이어질 수 있는 천이수로 계산한다.

TCS에서 천이가 일어날 경우 그 천이에 의해서 이어지는, 출력을 입력으로 받는 천이는 고정되어 있으므로, 그에 의한 변이는 없다. 따라서 시간상의 순서가 문제가 되고, 또한 시간상에 있어서 기준점으로 작용하는 것은 타이머이므로, 정의 5의 조건을 만족한다면, 두 상태에서 실행 가능한 천이의 순서들은 일치한다고 볼 수 있다. 따라서 두 상태를 같은 상태로 볼 수 있다. 이는 외부에서 관찰 가능한 action들의 순서와 경우의 수가 같은 것으로 볼 수 있으므로 weak bisimulation equivalent[9]하다고 볼 수 있다.

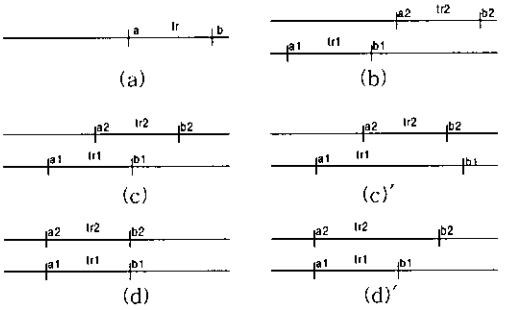

그림 2 천이 간 시간 관계

Reachability graph를 그리는데 있어서 필요한 것은 현재 상태에서 어떤 천이가 가능한가를 결정하는 것이다. TCS의 어떤 상태에 있어서 실행되려는 천이들이 있을 때, 그 천이들이 가질 수 있는 가지 수는 다음과 같고, 각각의 경우에 대한 시간 축에서 본 그림이 그림 2와 같다.

(a) 실행되려는 천이가 하나 뿐인 경우

(b) 실행되려는 천이가 둘 이상이며, 가장 빠른 두 개가 서로 겹치지 않는 경우

(c) 실행되려는 천이가 둘 이상이며, 가장 빠른 두 개가 서로 겹치는 경우

(d) 실행되려는 천이가 둘 이상이며, 가장 빠른 두 개의 최소값이 같은 경우

(a)의 경우에는 다른 경우가 없으므로 (a, b)에서 수행시켜주면 된다.

(b)의 경우에는 tr1은 (a1, b1)에서 수행되게 된다. 만약 tr2가 일어나기 전에 tr1에 의해서 유도된 천이 tr3가 수행될 수 있는 경우와 그렇지 않은 경우로 나누어 생각할 수 있다. 만약 tr3가 tr2보다 먼저 수행될 수 있는 경우에는 tr3가 tr2보다 먼저 일어날 수 있도록 /tr1이 수행되는 경우와 그렇지 않은 경우로 나뉘게 된다.

(c)의 경우도 (b)와 마찬가지로 생각할 수 있다. 다만 그 이전에 tr1이 (a1, a2)의 구간에서 수행될 경우와 tr1이 (a2, b1)에서 수행될 경우로 먼저 나누어야 한다.

(d)의 경우에는 수행 가능 최소 시간이 같은 천이가 더 있을 수 있다. 따라서 시간 간격을 (b)의 경우와 같이 좁힌 후, 좁혀진 시간 간격 내에서 같이 수행될 수 있는 천이들이 수행될 조합의 경우의 수만큼 reachability graph내에서 서로 다른 천이가 발생할 수 있다. 좁혀진 시간간격 내에 반드시 수행되어야하는 경우와 같은 입력 action을 공유하여 서로 상호 배타적인 천이들에 대해서 생각해 주어야 한다.

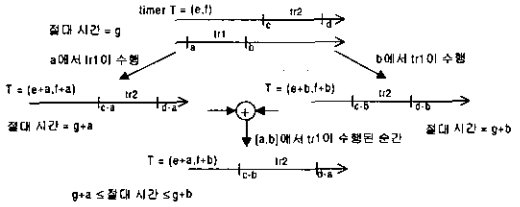

그림 6 천이 후 시간의 계산

위와 같은 방법에 의해 수행될 천이(들)와 천이가 일어날 시간간격이 정해지면, 수행되지 않을 천이들의 남은 시간간격을 조정해 주어야 한다. 이 때, 상대 시간이 아닌 절대적 시간으로 선후 관계를 따질 경우 초기 상태에서 시스템이 천이를 시작한 이후로 얼마만큼의 시간이 어떻게 흘렀는지, 현재까지의 경로를 항상 염두에 두어야 하게 되고, 이는 동일 상태를 구분하는데 있어서 현재까지의 경로를 비교하여야 하고, 결과적으로 시간에 의한 상태수 증가의 해결에 도움이 되지 못한다.

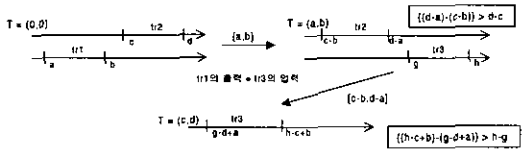

그림 7 천이 후 시간 계산과 타이머

상대적 시간으로 선후관계를 따질 경우는 다음과 같이 시간을 처리하게 된다. 위의 그림 3을 보자. (c, d)란 시간 간격은 (a, b)란 시간이 지난 후, (c-b, d-a)와 같은 형태를 띄게 된다. 여기서 문제는 원래의 시간간격의 길이, (d-c)보다 시간이 지난 후의 시간간격의 길이 {(d-a)-(c-b)} = (d-c+b-a)가 더 넓어져서, 최소값은 점점 앞으로 당겨지고, 최대값은 점점 미루어지는 것처럼 표현된다는 점이다. 이는 시간을 점으로 처리하지 않고 구간으로 처리하며, 감소를 표현하기 위해 감산을 사용하기 때문에 피할 수 없는 문제이다. 이것을 처리하기 위해서 타이머라는 시간의 기준점을 사용하였다. (a, b)란 시간 후에 (c, d)란 시간이 흐르면, 전체 흐른 시간은 (a+c, b+d)와 같이 된다. 시간이 더해지는 경우에는 감산에 의한 시간의 왜곡이 없다는 점에 착안하였다. 천이가 선택된 순간, 앞선 천이의 수행으로 출력 action이 일어나 그것을 입력 action으로 받아들인 순간의 타이머의 값과 선택된 천이가 일어나게 될 시간간격을 더한 값을 천이가 일어날 수 있는 시간간격으로 본다. 그리고, 이를 천이가 일어날 수 있는 시간값의 한계로 정하였다. 그 이후, 천이가 일어난 후 타이머의 값을 경과시킬 때, 앞서 계산해 둔 한계값을 벗어나는 경우 이는 감산에 의한 왜곡임으로 받아들여서 시간을 보정하도록 하였다. 이것을 앞의 그림 4의 예에서 보게 되면, tr3가 유도됐을 때, 그 상태에서 tr3가 수행된 후의 timer가 가져야 하는 값은 (a, b)이후에 (g, h)라는 시간이 흐른 것이므로 (a+g, b+h)가 되어야 하므로 이를 tr3와 함께 묶어두어 tr3가 수행된 후의 시간은 (c+g-d+a, d+h-c+b)가 아닌 보정된 값 (a+g, b+h)으로 결정하여야 하는 것이다. 또한, 계속적으로 천이가 미루어져, 타이머 값이 받아들일 수 없을 정도로 왜곡되기 전에 천이가 수행되도록 하여야 한다.

알고리즘의 커다란 윤곽은 다음과 같다.

```
algorithm build_reachability_graph
{
  초기 상태를 설정한다.
  while( 선택할 수 있는 천이가 있다 )
  {
    if((a)의 경우)
    then  tr의 실행;
    else if( (b),(c)의 경우)
      then 시간간격의 조절과 수행될 천이의 선택과 실행;
      else
        (d)의 경우에서의 시간간격 조절과 수행될 천이의 선택과 실행;
  }
}
function 천이의수행
{
  수행되지 않은 천이들의 시간 감소;
  타이머의 처리;
  새로 선택된 천이의 한계값 계산;
  중복 state의 정리;
  time out 천이에 대한 고려;
}
```

4. ABP에 대한 적용 결과

[5]에서도 예제로 사용했던 alternating bit protocol (ABP)이라는 간단한 프로토콜에 대해서 적용해 보았다.

전송시 데이타를 보낸 후 타이머를 동작시켜서 acknowledgement가 오지 않으면, packet이 도달하지 못한 것으로 간주하여 재전송을 하는 특성을 가지고 있다. Packet이 사라지는 통신선로에서 작동할 때, acknowledgement가 손실되는 경우 livelock에 들어가는 성질을 가졌다. 기존 방안에서는 모델링할 수 없었던 송수신부 모두 갖춘 ABP의 reachability graph를 얻을 수 있었다. 본 논문에서는 지면 관계상 reachability graph 중 livelock이 일어났음을 보이는 그림 5만 보이도록 하겠다.

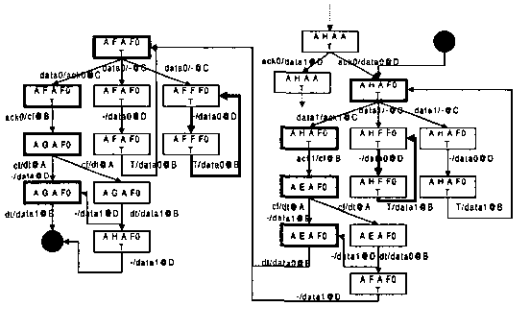

그림 8 ABP에서 일어나는 livelock

Reachability graph에서 벗어날 수 없는 cycle이나 subgraph는 livelock을 의미하고, 나가는 천이가 없는 상태는 deadlock을 의미한다[5]. 결과 reachability graph에서 livelock 상태가 걸리는 것을 보이는 그림을 추출하면 위의 그림 5와 같다. 기본적으로 그림 5에서 굵은 선으로 표시한 2개의 천이로 이루어지는 두 cycle은 송수신 양단에 걸쳐서 livelock이 일어났음을 보이고 있다. 또한 그림 전체적으로 볼 때, 굵은 선으로 표시한 상태들로 이루어지는 cycle과 그 cycle내의 subgraph는 0 bit 상태에서 machine B의 송신단과 machine C가 livelock 상태에 들어갔음을 보이고 있다.

5. 결 론

본 논문에서는 프로토콜을 검증하는데 있어서 필수적인 reachability graph를 타이머를 포함하고, 시간 개념을 지닌 복수의 temporal machine으로 구성된 시스템으로부터 유도해내는 방법을 제안하였다.

기존의 방법에서 시간 개념을 포함하기는 했으나 타이머에 대한 모델링과 처리가 불가능했던 면을 타이머를 포함하여 프로토콜을 적절히 모델링하고 reachability graph를 작성해낼 수 있음을 보임을 통해 해결하였다.

Machine 사이에서 메시지를 전달하는 전송선로를 또 다른 machine으로서 모델링할 수 있도록 하여 더욱 실제 시스템에 가깝게 모델링이 가능하게 하고, 또 전송선로에 따라서 문제가 있을 수 있는 경우에 대해서도 문제를 발견해낼 수 있었다.

보통의 state machine으로부터 reachability graph를 유도하는 경우에 비해 시간 개념이 들어갈 경우에 생기는 상태 수 증가의 문제를 시간 간격을 통해 묶어나가는 방법으로 해결하였으며, 동일한 상태를 구분 짓는 문제도 상대적 시간 개념을 통해서 과거에 상관없이 현재의 상태, 수행될 천이, 천이들의 선후관계, 타이머와의 시간관계를 비교함으로써 동일 상태를 구분 짓는 방법을 제시하여 해결하였다.

또한 천이들의 선후관계 비교를 상대적 시간 개념을 통해 해결해 나가려는 과정에서 발생하는 시간의 왜곡 문제를 타이머를 상대적 시간의 기준점으로 사용하여 시간의 한계를 정하는 방법으로 해결하였다.

추후 연구 과제로는 FSM에서 확장하여 EFSM에서의 reachability graph를 도출해내는 방법에 대한 연구가 필요하겠다.

참 고 문 헌

[1] Gerald J. Holzmann, "Design and Validation of Computer Protocols," Prentice Hall Software Series, 1991.

[2] Dieter Barnard, and Simon Crosby, "The Specification and Verification of an Experimental ATM Signalling Protocol," Protocol Specification, Testing and Verification XV, pp. 155-170, Chapman & Hall, 1996.

[3] Stefan Leue, "Specifying Real-Time Requirements for SDL Specifications - A Temporal Logic-Based Approach," Protocol Specification, Testing and Verification XV, pp. 19-34, Chapman & Hall, 1996.

[4] Kshirasagar Naik and Behcet Sarikaya, "Protocol Conformance Test Case Verification Using Timed-Transitions," Protocol Specification, Testing and Verification XIV, pp. 103-118, Chapman & Hall, 1995.

[5] S. Bradley, W.D. Henderson, D. Kendall and A.P. Robson, "Validation, Verification and Implementation of Timed Protocols using AORTA," Protocol Specification, Testing and Verification XV, pp. 205-220, Chapman & Hall, 1996.

[6] L. Cacciari and O. Rafiq, "A Temporal reachability analysis," Protocol Specification, Testing and Verification XV, pp. 35-49, Chapman & Hall, 1996.

[7] Inhye Kang and Insup Lee, "An Efficient State

Space Generation for Analysis of Real-time Systems," International Symposium on Software Testing and Analysis '96, pp 4-13, 1996.
[8] Andrew S. Tanenbaum, "Computer Networks 3rd edition," Prentice Hall, 1996.
[9] Gerrit Jan Tretmans, "A Formal Approach to Conformance Testing," 박사학위논문, Faculteit der Informatica Universiteit Twente, Netherlands, 1992.
[10] Robin Milner, "Communication and Concurrency," Prentice Hall, 1989.
[11] Fred D.J. Bowden, "Modelling time in Petri nets," The Second Australia-Japan Workshop on Stochastic Models in Engineering, Technology and Management, Gold Coast, July 17-19 1996, pp. 66-81, 1996.

박 찬 민

1997년 연세대학교 전자공학과 학사. 1999년 연세대학교 전자공학과 석사. 1999년 ~ 현재 한국통신프리텔 기술연구소 근무. 주요 관심분야는 프로토콜 엔지니어링, 무선 화상통신 프로토콜, 무선 VoIP, 초고속 무선 인터넷 서비스, 블루투스 관련 기술 및 서비스.

황 익 순

1995년 연세대학교 전자공학과 학사. 1997년 연세대학교 전자공학과 석사. 1997년 ~ 현재 연세대학교 전자공학과 박사과정. 관심분야는 프로토콜 공학, 소프트웨어 공학, 컴퓨터 통신.

김 태 형

1992년 연세대학교 전자공학과 학사. 1995년 연세대학교 전자공학과 석사. 1996년 ~ 현재 연세대학교 전자공학과 박사과정. 관심분야는 프로토콜 공학, 소프트웨어 공학.

이 재 용

1997년 3월 연세대학교 전자공학과 공학사. 1984년 5월 Iowa State Univ. 컴퓨터 공학과 공학석사. 1987년 5월 Iowa State Univ., 컴퓨터공학과 공학박사. 1977년 2월 ~ 1982년 6월 국방과학연구소 연구원. 1983년 1월 ~ 1986년 12월 Iowa State Univ., 연구조원. 1987년 6월 ~ 1984년 8월 포항공과대학교 전자계산학과 부교수. 1994년 9월 ~ 현재 연세대학교 전자공학과 교수. 관심분야는 프로토콜 공학, 고속/멀티미디어 통신 프로토콜, 망 관리.

이 상 배

1958년 공군사관학교 통신공학과 공학사. 1961년 서울대학교 전자공학과 공학사. 1964년 Stanford Univ. 전기공학과 공학석사. 1975년 영국 Newcastle Univ. 전자공학과 공학박사. 1969년 ~ 1979년 서울대학교 공과대학 조교수. 1979년 3월 ~ 2000년 2월 연세대학교 전자공학과 교수. 2000년 3월 ~ 현재 연세대학교 전기·전자공학과 자문교수. 1982년 9월 ~ 1983년 9월 영국 Newcastle Univ. 방문교수. 1990년 1월 ~ 1990년 12월 대한전자공학회 회장. 관심분야는 회로 및 시스템, 컴퓨터 네트워크.