추상 시간 기계를 사용한 실시간 시스템의 역명세 검증

박지연°, 노경주, 이문근
전북대학교 컴퓨터과학과
e-mail : {jypark, kjuno, mklee}@cs.chonbuk.ac.kr

Verification of Reverse specification for Real-Time System in Abstract Timed Machine

Ji-yon Park°, Kyoung-ju Noh, Moon-kun Lee
Dept. of Computer Science, Chonbuk National University

요 약

본 논문은 ATM(*Abstract Timed Machine*)으로 명세된 실시간 시스템을 검증하기 위한 방법을 기술한다. ATM은 임무 위급 시스템인 실시간 시스템을 명세, 분석, 검증하기 위한 정형기법이다. ATM은 모드와 전이, 포트로 구성된다. 다른 정형기법과 비교하여 ATM은 소프트웨어의 순환공학 과정에서 사용하기 위해 설계되었다. 역공학 과정에서 ATM은 계산 논리 뿐만 아니라 실시간 시스템의 실제 소스코드에 있는 설계나 환경정보를 표현할 수 있다. 이러한 목적을 위해 ATM은 다양한 모드를 사용한다. ATM을 사용한 실시간 시스템의 검증은 도달성 그래프를 생성함으로써 수행한다. 도달성 그래프는 상태와 시간을 추상화되고 압축된 형태로 표현할 수 있으며 그 결과 시간 속성을 지닌 상태 공간을 감소시킬 수 있다. 또한 시스템의 교착상태를 쉽게 발견할 수 있다. 본 논문은 ATM과 실행 모델, 도달성 그래프, 검증을 위한 속성 등을 기술하며 이들을 다른 정형 기법들과 예제를 통하여 비교한다.

1. 서 론

ATM은 복잡도가 높고 방대한 실시간 시스템의 명확한 명세와 역공학 과정에서 원시 시스템의 충분한 정보제공을 위해 개발된 정형기법이다. 본 논문에서는 ATM을 사용하여 명세한 시스템을 검증하는 방법을 기술한다. 시스템의 검증은 도달성 그래프를 생성함으로써 이루어진다. 생성되는 도달성 그래프는 시스템의 모든 동작을 표현하며 유한한 노드를 가진다. 도달성 그래프는 시스템 각각의 상태를 노드로 하여 모든 전이 과정에서 발생하는 시스템의 상태를 표현한다. 따라서 도달성 그래프의 노드와 에지를 분석함으로써 시스템의 모든 동작은 초기 상태로부터 추적가능하다.

도달성 그래프 생성 시 중요 쟁점은 시스템에서 발생하는 상태를 효율적으로 감소시키는 것이다. 이는 ATM의 모드가 압축된 상태를 표현하기 때문에 하나의 모드에 의해 여러 상태를 포함할 수 있기 때문이다. 또한 그래프 노드 생성 시 시간에 대한 조건을 줌으로써 시간에 따른 노드의 증가를 감소시킨다.

ATM의 검증방법은 ATM의 명세 특징으로 인해 전체 시스템이 최종 상태로 도달 가능한지 쉽게 파악할 수 있으며 이에 따라 교착상태(deadlock) 또는 무한실행(livelock) 상태 등을 발견할 수 있다.

2. 관련연구

정형기법들이 명세한 시스템을 검증하기 위한 여러 방법론이 고안되었다. 이의 대표적인 것으로 CSM(Communicating State Machines)과 이에 대한 상태 최소화(State Minimization) 방법[4], CRSM(Communicating Real-Time State Machines)[2]과 이에 대한 도달성 그래프(*Reachability Graph*) 방법[8]이 있다.

CSM에서의 상태 최소화 방법은 시스템의 전이 가능한 모든 상태를 명확하게 도달가능한 상태로 나누어가면서 전이되는 시스템의 상태를 중복되지 않게 표현하는 방법이다. 그러나 CSM이 시간 속성을 표현하지 못하기 때문에 상태 최소화 방법은 시간 개념을 가진 실시간 시스템의 검증에 어려움을 가진다.. 또한 상태 최소화 방법은 실제 전이와 관련 없는 여러 변수 값을 시스템의 검증을 위한 상태에 포함함으로써 실제 검증 과정에서 고려하지 않아도 되는 상태를 유발한다. 이러한 방법은 많은 상태의 증가를 피할 수 없으며 이에 따른 검증 과정의 오버헤드도 피할 수 없게 된다.

CRSM에서의 도달성 그래프를 통한 검증방법은 시간 속성을 지닌 실시간 시스템을 검증하기 위한 방법을 제시하였다. 이 도달성 그래프는 CSM에 대한 상태 최소화 방법과 같이 모든 변수에 대해 시스템의 상태를 증가시킴으로 검증 과정에서 검증이 불필요한 상태를 생성시켜 검증의 상태 공간을 증가시킨다. 또한 CRSM의 도달성 그래프에서의 노드는 시간의 이산적 변화를 민감하게 고려하여 상태를 증가시킴으로써 전이 가능한 시간 구간 안에서 조차도 이산적 시간 변화에 따른 많은 상태를 유발한다.

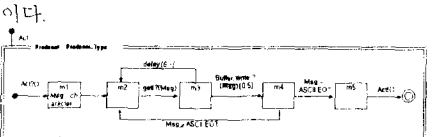
CSM과 CRSM의 검증 방법이 가진 이러한 단점을 극복하면서 실시간 시스템에 보다 적합한 검증 방법이 요구되며 이러한 요구사항을 만족시키기 위해 새로운 생성조건을 가진 도달성 그래프를 통한 검증방법을 고안하였다.

3. ATM

ATM은 계층성, 분산성, 실시간성, 우선권, 다수에 의한 동기화, 예외처리, 실행성 등 다양한 속성을 가진 실시간 시스템을 역공학 과정에서 명세하기 위해 개발된 LTS(Labeled Transition System)이다.

ATM은 모드(mode)의 집합, 가드(guard)된 전이의 집합, 포트(port), 실행 시작점과 실행의 종료점으로 구성된 머신이다.

ATM은 소스 코드로부터 프로그램 블록 단위에 따른 머신과 머신 내의 모드를 포함하는 타입 ATM을 생성하는 해석 (Interpretation) 단계와 시스템의 동적 행위에 따라 타입 ATM의 재구성하는 재구성 단계, 명세언어로의 표현 단계를 거쳐 원시 소프트웨어를 명세한다. <그림 1>은 ATM의 실제 명세 예제이다.



<그림 1> ATM 명세 예제

4. 도달성 그래프

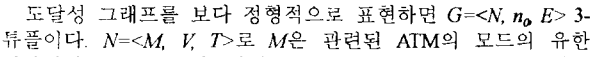
4.1 정의

도달성 그래프는 시스템의 실제 동작을 표현하는 실행 모델로서 노드와 에지로 구성된다.

본 연구는 한국과학재단 특정기초연구(과제번호 1999-2-003-3) 지원으로 수행되었음.

노드는 각 ATM의 전이에 따른 시스템의 모드, 시간, 전이에 관련된 변수의 값을 원소로 하는 튜플 형태로 표현된다. 검증 과정과 관련된 머신을 구분하기 위해 구분자 '{', '}'와 모드의 진입에 따른 시간과 값의 변화를 나타내는 기호 '[', ']'를 사용한다.

에지는 ATM의 발생가능한 전이에 따라 생성되는 노드간의 이동을 의미한다. 에지는 전이가 이루어질 때의 이벤트를 레이블로 갖는다. <그림 2>는 노드의 형식을 보여준다. 예제 노드는 두 개의 ATM이 관련되어 있으며 $m1$과 $n1$은 각 ATM의 현재 모드를 나타낸다. x_f x_l는 전이와 관련있는 변수로 x_f는 모드로 진입할 때의 x 값을, x_l은 모드를 벗어날 때의 값을 나타낸다. 마찬가지로, t_f은 모드로 진입한 시간을 나타내고, t_l은 모드에 머무른 시간을 의미한다. 시간을 나타내는 값은 숫자 또는 '-'이 된다. '-'은 전이가 발생하여 모드에 진입했을 경우 모드의 다음 전이가 시간과 관계 없는 경우에 표기되는 값이다. 만약 모드에 진입했을 경우 모드의 다음 전이가 시간 제약을 가졌다면 t_f은 '0'값으로 설정된다. 만약 전이가 t_l과 관련없이 진행되었고 '-'이 아니라면 x_l은 모드에 머무른 만큼의 시간 값을 가진다.

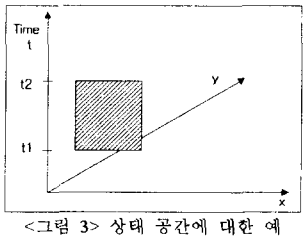


<그림 2> 도달성 그래프의 노드

도달성 그래프를 보다 정형적으로 표현하면 $G=<N, n_0, E>$ 3-튜플이다. $N=<M, V, T>$로 M은 관련된 ATM의 모드의 유한 집합이며 V는 전이와 관련된 변수 값의 집합, T는 전이와 관련된 시간 값의 집합이다. $n_0 \in N$는 그래프의 시작 노드를 의미한다. E는 유한한 에지의 집합으로 ATM의 발생가능한 전이에 따라 생성되며 전이에 따른 이벤트를 레이블로 갖는다.

4.2 도달성 그래프의 특징

ATM은 모드 특성으로 인하여 다른 정형 기법에 비해 많은 상태 공간을 감소시켜 명세 과정에서 보다 적은 상태 공간을 표현하였다. 따라서 도달성 그래프도 다른 정형 기법에서 보다 적은 노드로 시스템을 검증한다. 추가적으로 실시간 시스템의 검증을 위한 상태 공간의 감소를 위해 두 가지 생성조건을 가진다. 첫번째 생성조건은 도달성 그래프 노드의 한 구성 원소인 변수를 전이와 관련 있는 변수로 제한하는 것이다. CSM과 CRSM의 검증 방법은 모든 변수 값을 고려하기 때문에 실제 전이와 관련 없는 변수에 대한 노드가 생성되며 이에 따른 상태 공간의 증가는 막대하게 된다. ATM의 도달성 그래프는 노드를 이루는 변수를 전이와 관련된 것으로만 제한함으로써 상태 공간을 감소시킨다. 두 번째 조건은 시간에 관한 것이다. 한 노드가 생성되어 있고 다른 노드가 생성될 경우 시간을 제외한 다른 원소의 값이 같고 두 노드의 시간 값이 전이 가능한 시간 안에 존재하는 경우 두 노드를 같은 노드로 취급 함으로써 발생하는 노드의 수를 줄이는 것이다.



<그림 3> 상태 공간에 대한 예

<그림 3>은 이러한 예를 보여준다. 발생 가능한 상태 공간은 변수 x, y 값의 변화와 시간 t의 변화 공간이다. 전이가 시간 t1, t2와 y값에 의해 결정 되며 x값의 변화가 전이에 영향을 주지 않을 때 y값이 같고 노드의 시간 값이 t1과 t2 안에 있다면 이의 상태는 빗금친 부분이며 이 부분 내의 어떠한 값도 같은 상태로서 취급될 수 있는 상태 공간이다. 따라서 시스템의 상태 값이 사각형 내의 어느 점이라도 같은 상태 공간으로 보며 그 결과 검증을 위한 상태를 감소 시킬 수 있다.

시간에 관한 상태 감소 요인은 노드의 시간 표기에서 발생한다. 시간과 관련없는 전이에 대한 노드 생성시 '-'을 사용함으로써 시간과 관계없는 모드에서 시간의 이산적 변화에 의해 증가되는 상태 공간을 줄일 수 있다. 시간과 관련하여 또 다른 특징은 ATM의 명세 방법에 의해 발생한다. 일반적으로 시간 제약을 명세할 경우 전이상에 전이 가능한 제약만을 레이블로 한다. 그러나 ATM은 <그림 1>의 delay[6,-]이 보여 주듯이 시간 제약을 만족하지 못했을 경우의 전이의 진행을 명시적으로 표현함으로써 다른 정형 기법의 경우에는 추론해야 만이 검증 가능하였던 시스템의 진행 상태를 명시적으로 그래프 내에 표현할 수 있으며 그래프 생성 알고리즘도 간략화 시킬 수 있다.

도달성 그래프에서 ATM의 명세 방법에 따른 추가적 특징은 ATM의 명세는 분명한 종료점이 명시된다는 것이다. 즉, 관련된 머신이 최종적으로 종료 상태를 가지지 못했을 경우 시스템은 교착상태 혹은 무한실행 상태에 있으며 이의 발견이 쉽게 이루어진다.

4.3 도달성 그래프 생성 알고리즘

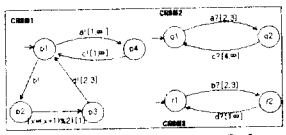
도달성 그래프는 검증할 ATM의 초기값을 입력 받아 이를 시작 노드로 하여 생성된다. 후속 노드는 현재 노드의 전이 중 최소시간(lower bound)에 전이 가능한 모든 이벤트에 대하여 생성된다. 전이가 시간 제약을 가졌을 경우에는 최소시간에 전이가 이루어진다고 가정하고 시간 제약이 없는 경우는 사용자가 입력한 전이 시간을 이벤트 전이에 소요되는 최소시간으로 가정한다. 이와 같은 가정은 시스템이 전이 가능하게 된 순간에 전이가 이루어 짐을 의미한다. 만약 현재 노드의 다음 전이가 RPC 이벤트이고 시간 제약을 가졌다면 송신은 최소 시간에 수신은 최대 시간에 발생하며 이 두 전이 사이에 다른 전이는 발생하지 않는 것으로 한다. <그림 4>는 도달성 그래프 생성 알고리즘에 대한 의사코드(pseudo-code)이다.

```
input : initial value of selected ATMs
output : rechability graph

rechability_graph()
{begein
no=initial value ;
make_graph(n0);
Eend

make_graph(node N)
begin
while(N's transition is Exist on lower time) //최소 전이시간에전이가능한전이가존재하는한
begin:
    T=choose_transiton(N); //탐색되지않은전이중하나선택
    CN = cumpute_node(N,T); /노드N으로부터전이T에의해생성되는후속노드계산
      if(CN = one of Existing Nodes)
      begin
      add_edge(N, Existing Node);
            /계산된노드가이미생성된노드와같은경우노드N으로부터연결되는노드로에지추가
      end
      else
      begin
      add_edge(N, CN); /노드N으로부터새롭운노드CN으로에지추가
      make_graph(CN); /노드CN에대한새롭운후속그래프생성
      end
end
return;
end
```

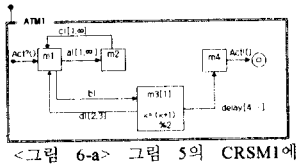
<그림 4> 도달성 그래프 생성 알고리즘

make_graph(node N)는 노드 N에 대하여 최소 시간에 대한 N의 모든 전이에 대하여 후속 노드를 생성하는 함수이다. choose_transition(N)은 노드 N의 전이 중 아직 탐색하지 않은 전이를 선택하는 함수이다. add_edge(N, CN)은 노드 N에서 노드 CN으로 에지를 생성시키는 함수이며 도달성 그래프는 초기 노드로부터 make_graph(node N)을 재귀적으로 호출하여 생성된다.

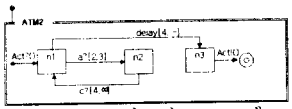
4.4 도달성 그래프 예제



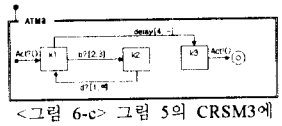
<그림 5> [7]의 CRSM 예제



<그림 6-a> 그림 5의 CRSM1에 대응하는 ATM



<그림 6-b> 그림 5의 CRSM2에 대응하는 ATM



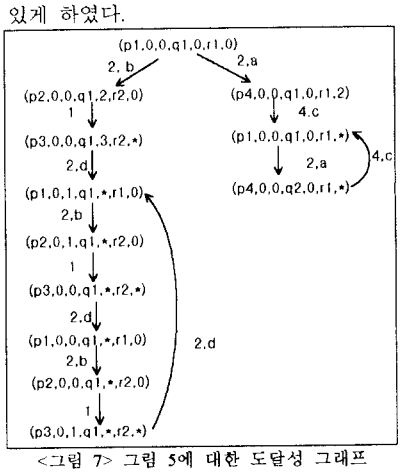
<그림 6-c> 그림 5의 CRSM3에 대응하는 ATM

<그림 5>는 [7]의 CRSM에서 도달성 그래프를 생성하기 위해 사용한 예제이다. <그림 6>는 이를 ATM 방법을 사용하여 재명세한 것으로 CRSM1이 ATM1, CRSM2가 ATM2, CRSM3가 ATM3을 각각 나타낸다.

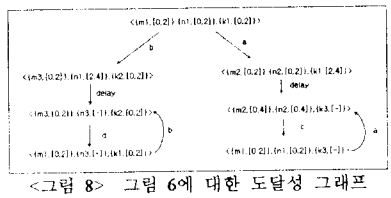
ATM 명세의 경우 delay[4,-]과 같이 시간 제약이 만족하지 못했을 경우 시스템의 제어 이동을 명시적으로 표현하였다.

<그림 7>은 CRSM의 도달성 그래프이고 <그림 8>은 ATM의 도달성 그래프이다. 비교의 명확성을 위해 Act 시그널과 관련된 이벤트는 생략하였다.

ATM의 경우 CRSM의 경우보다 적은 노드가 생성되었다. CRSM과 달리 전이와 관계없는 변수 x를 고려하지 않음으로써 노드의 감소 효과를 가져올 수 있었다. 또한 <그림 7>의 도달성 그래프가 2번의 과정을 거쳐 생성된 것에 비해 ATM의 것은 1번의 실행으로 생성되었다. ATM이 시간 제약에 대하여 delay 이벤트의 추가로 시간의 흐름에 대한 명확한 전이를 제공하였기 때문에 전의의 발생 수는 증가하였지만 명확한 전이를 제공함으로써 도달성 그래프를 보다 효율적으로 생성할 수 있게 하였다.

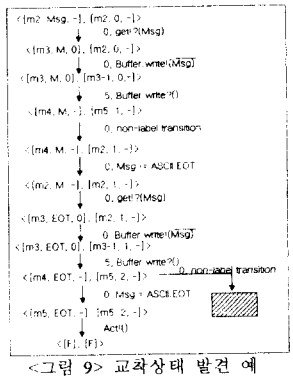


<그림 7> 그림 5에 대한 도달성 그래프



<그림 8> 그림 6에 대한 도달성 그래프

<그림 9>은 교착상태 발견에 대한 예제이다. 두개의 ATM에 대한 그래프의 <[F],[F]> 노드는 관련 ATM이 종료 가능함을 보여주는 노드이다. 만약 시스템이 이러한 노드를 포함하지 않는 다면 교착상태 혹은 무한 실행 상태에 있다고 간주할 수 있다.



<그림 9> 교착상태 발견 예

5. 결론 및 향후 연구

본 논문에서는 ATM으로 명세된 시스템을 도달성 그래프를 사용하여 검증하는 방법을 기술하였다. ATM이 가지는 시간 명세 방법과 압축된 상태를 표현하는 모드의 사용과 그래프 생성 과정에서 상태 축소를 위한 추가적 생성 조건을 줌으로써 효율적인 도달성 그래프를 생성하였다.

본 연구가 순환 공학을 위한 정형 기법을 대략 설계하고 프로토타입하는 단계에 있어 검증 방법도 추가로 정의되고 보완되어야할 사항이 존재한다. 예를 들어 레이블이 없는 경우 대해 어떠한 노드를 생성해야 하는 가 등의 주제는 상태 공간을 줄일 수 있는 방법이 될 수 있다. 전이에 따른 노드가 어떻게 생성되느냐에 따라 발생 가능한 노드의 수는 증가할 수도 있고 감소할 수도 있기 때문이다. 따라서 전이와 관련한 실시간 시스템의 상태 감소 방법이 계속 연구되어야 한다.

참고문헌

[1] C.A.R.Hoare, "*Communicating Sequential Processes*," Prentice-Hall MD.

[2] D. Harel, "Statecharts: A Visual Formalism for Complex System," Science of *Computer Programming*. Vol. 8. 1987, pp. 231-274.

[3] Feldman and Koffman, "Ada95," Addison-Wesley, 1996.

[4] I. Kang and I. Lee, "State Minimization for Concurrent System Analysis Based on State Space Exploration," *Proceedings of Conference on Computer Assurance*, Gaithersburg MD, June 1994.

[5] Moon Lee, *"An Environment for Understanding of Real-time Systems,"* Ph.D. Thesis The University of Pennsylvania, 1995.

[6] S. jahanian and A. Mok, "Modechart : *A Specification Language for Real Time Systems,"* IBM Technical Report: RC 15140, November 1989.

[7] Sitaram C. V. Raju, "*An Automatic Verification Technique for Communicating Real-Time State Machines*",