김일성종합대학학보
(자연과학)
주체106(2017)년 제63권 제5호

JOURNAL OF **KIM IL SUNG** UNIVERSITY
(NATURAL SCIENCE)
Vol. 63 No. 5 JUCHE106(2017).

# 모형검사를 위한 속성명세서작성을 지원하는 한가지 방법

**신춘옥, 공현우, 승남철**

경애하는 최고령도자 **김정은**동지께서는 다음과 같이 말씀하시였다.

《과학연구부문에서는 주체공업, 사회주의자립경제의 위력을 강화하고 인민생활을 향상시키는데서 나서는 과학기술적문제들을 우선적으로 해결하며 최첨단의 새로운 경지를 개척하기 위한 연구사업을 심화시켜야 합니다.》

속성명세서를 작성하기 위한 한가지 방법은 패턴과 그 적용범위를 리용하여 사용자가 선택할수 있는 LTL시제론리식들의 서고를 구축하는것이고[2, 3, 6] 다른 방법은 그라프적형식화를 리용하여 시간속성명세서를 작성하는것이다.[4, 5] 선행연구[4]에서 제안한 시간선편집기는 사건들의 사슬에 대한 직관적인 표기방법을 받아들였지만 부분구간에서의 순서관계를 .지적하지 못하였다.

우리는 그라프적인 대본표기에 의하여 속성명세서작성을 지원하는 한가지 방법을 연구하였다.

## 1. 시간선도명세서

모형검사기 SPIN에서 속성을 서술하는 LTL시제론리는 시제연산자 X(next)를 리용하지 않는다.[1] X(next)연산자에 무관계한 시제론리는 림시불변속성을 가진다.[6] 그러므로 사건들의 순서관계와 그것들에 대하여 성립하여야 할 제약만을 규정하여 속성명세서를 작성할수 있다.

론문에서는 비형식적으로 정확성속성을 정의하는 시간선도명세서언어를 도입하였다.

정의 1 사건식은 $Events = Event_{comp} \cup Event_{reg} \cup Event_{res} \cup Event_{fail}$ 우에서 정의된 *EventOp*에 의하여 다음과 같이 귀납적으로 형성되는 식이다.

$$< eventfml >::= e : ev \mid r : ev \mid f : ev \tag{1}$$

$$<eventfml>::=<eventfml>;\ <eventfml>;\ \cdots;\ <eventfml> \tag{2}$$

$$<eventfml>::=<eventfml>+<eventfml>+\cdots+<eventfml> \tag{3}$$

$$<eventfml>::=<eventfml>\|<eventfml>\|\cdots\|<eventfml> \tag{4}$$

$$<eventfml>::=@(<eventfml>,\ m) \tag{5}$$

여기서

정규사건 $e : ev \in Event_{\mathrm{reg}}$ 는 사용자들의 입력 혹은 외부적인 작용들과 같은 사건으로서 속성은 그것들이 발생하는가 안하는가에 관계된다. 여기서 *ev*는 사건의 이름을 지적한다.

요구사건 $r : ev \in Event_{\mathrm{res}}$ 가 발생되지 않으면 체계는 오유동작을 하는것으로 본다.

실패사건 $f : ev \in Event_{\mathrm{fail}}$ 는 체계에서 발생되지 말아야 할 사건으로서 실패사건이 검출

되면 오유로 된다.

*EventOp*={||, ;, +, @}는 복합사건들을 지적하기 위하여 *Events*우에 정의된 연산자들의 모임이다. 여기서 《||》는 동시사건, 《;》은 사건사슬, 《+》는 대리선택연산자이며 @는 사건렬의 반복연산자로서 식 @(*x*, *y*)에서 첫 인수 *x*는 반복되여야 할 사건 또는 사건렬을, *y*는 반복회수를 지적한다.

정의 2 시간선도명세서 *time lineSpec* =(*Events*, *Eventfmls*, *Constrs*, *Marks*)는 시간선과 그우에서의 표식들의 모임 *Marks*, 사건들의 모임 *Events*, 시간선상에서의 선후차관계와 사건발생에 대한 제약들의 모임을 정의하기 위한 그라프적인 표현이다.

$Marks = \{m_i = (i, ev_i)\}$ 는 시간선상에 수직선으로 표기된 표식들의 모임이다. 여기서 $i(0 \le i)$는 표식들의 번호를 지적하며 $ev_i \in Eventfmls$ 는 $i$번째 표식의 사건을 지적하는 사건식으로서 ↓를 접두사로 가진다.

제약 $c_i \in Constrs$ 는 제약식 $c_i\_f$ 와 시간선상에서 그것의 적용범위 $c_i\_scop$ 의 쌍 $c_i = (c_i\_f, c_i\_scop)$ 로 구성된다.

전화교환체계에 대한 정확성요구를 서술한 시각적인 시간선도명세서의 실례는 그림 1과 같다.

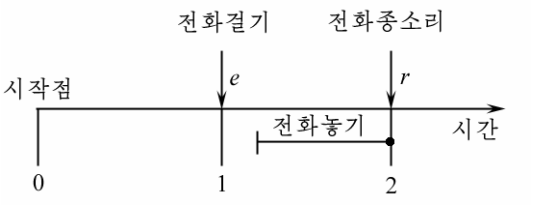


그림 1. 시간선도명세서의 실례

시간선도명세서에는 시작점을 원점으로 하는 주시간선이 있고 그 선우에 사건식들이 아래로 향한 화살표 ↓와 함께 표시된다. 또한 이 명세서상에서 제약 $c_i = (c_i\_f, c_i\_scop)$ 를 정의한다.

제약 $c_i$ 에서 성분 $c_i\_scop$ 는 주시간선도의 아래에 수평선을 그어 표기하며 시간선도표식에서 시작하고 끝난다. $c_i\_scop$ 는 시작표식 혹은 끝표식을 선택적으로 포함할수도 있고 포함하지 않을수도 있다.

제약 $c_i$ 의 조건식 $c_i\_f$ 는 $c_i\_scop$ 의 표시선우에 써넣는다.

표식들은 사건이름들이 붙은 얇은 수직막대기로 보여주며 매 표식은 하나의 사건표현식만을 나타낸다.

원점 시작점을 0번째 표식으로 하여 표식들에 순서대로 번호를 붙인다. $c_i\_scop$ 가 시작표식과 끝표식을 포함할수도, 포함하지 않을수도 있기때문에 표식을 포함하는 경우 그 표식을 검은색의 작은 원으로, 포함하지 않는 경우는 검은색의 작은 수직막대기로 표시한다.

$c_i\_scop$ 가 시작표식과 끝표식을 포함하는가 포함하지 않는가에 따라 다음과 같은 경우들로 나누어 론의한다.

① $c_i\_scop = [i, j]$: 제약이 표식 $i$로부터 표식 $j$까지 적용되는 경우

② $c_i\_scop = [i, j)$: 제약이 표식 $i$로부터 표식 $j$전까지 적용되는 경우

③ $c_i\_scop = (i, j]$: 제약이 표식 $i$후부터 표식 $j$까지 적용되는 경우

④ $c_i\_scop = (i, j)$: 제약이 표식 $i$후부터 표식 $j$전까지 적용되는 경우

시구간의 서술에서 괄호 [ 또는 ]는 해당한 표식을 포함한다는 의미이며 괄호 ( 또는 )는 해당한 표식을 포함하지 않는다는것을 의미한다.

## 2. 시간선도명세서로부터 뷰치자동체에로의 변환

론문에서는 시간선도명세서편집기를 리용하여 속성에 반영하여야 할 내용들을 검증자가 시간선도명세서로 작성하면 그것을 뷰치자동체의 본문적서술인 *never claim*으로 변환하여 프로멜라검증모형에 포함시켜 검증모형을 완성하는 방법을 적용하였다.

그리고 론리식에 대한 뷰치자동체를 생성할 때 제약을 가정하면서 하나의 표식에 지적된 사건에 해당한 부분자동체를 생성하고 최종상태를 통하여 부분자동체들을 합성하여 목적하는 자동체를 얻는다.

정의 3 뷰치자동체는 무한단어를 접수하는 유한자동체 $A=(\Sigma, S, S^0, \rho, F, l)$이다. 여기서 $\Sigma = Eventfmls \cup Constr$는 사건식과 제약들로 이루어진 자모, $S$는 유한이며 비지 않은 상태들의 모임, $S^0 \subset S$는 비지 않은 초기상태들의 모임, $F \subset S$는 접수상태들의 모임, $\rho: S \xrightarrow{\delta} 2^S$는 이행조건 $\delta$($\Sigma$우에서 정의된 론리식)가 표식붙은 상태이행함수이다.

우에서 정의한 시간선도명세서의 의미를 부정하는 뷰치자동체를 생성하는데서 매 표식을 단위로 부분자동체를 생성하고 부분자동체의 최종상태를 다음의 사건에 대한 부분자동체의 출발점으로 하여 뷰치자동체를 합성하였다.

1) 시간선도명세서의 단순사건들(식 (1))을 부분자동체로 넘기는 규칙

시간선도명세서에 표시된 단순사건형태에 따르는 부분자동체는 그림 2와 같다.

그림 2에서 상태 $s_0$은 사건발생전의 출발상태이고 색동그라미는 정규사건의 정확한 발생으로 인하여 이르게 되는 최종상태, 겹동그라미는 접수상태를 표시한다. 이행을 표시하는 화살표에는 이행조건 $\delta$가 표시되였으며 이행조건 true는 모든 가능한 사건을 표시한다. 최종상태에 이르면 체계의 다음동작이 계속되는가를 판정하여 귀납적인 합성을 계속한다. 전체 뷰치자동체는 이 부분자동체들의 귀납적인 합성으로 구축된다.

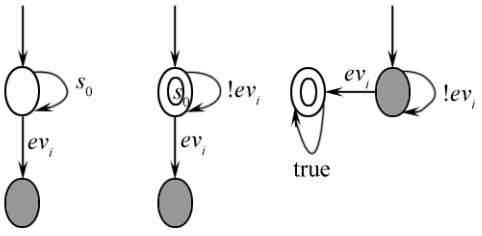


그림 2. 사건형에 따르는 부분자동체
ㄱ)－ㄷ)는 각각 하나의 정규사건,
요구사건, 실패사건인 경우

2) 사건식에 따르는 부분자동체들의 귀납적인 합성규칙

① 사건식 (2)의 경우

이 사건식은 사건렬의 사건들사이에 어떤 다른 사건이 끼워들수 없으며 사건렬이 순서대로 발생한다는것을 강조하기 위하여 쓰인다. 이 경우의 부분자동체들의 합성을 다음과 같이 진행한다. 매 $eventfml_i$에 해당한 부분자동체를 생성할 때 $eventfml_i=e$: $ev_i$인 경우 자동체에 표시된 이행조건식 true를 $!ev_i$로 바꾼다. 앞번호의 $eventfml_i$에 해당한 부분자동체의 최종상태를 뒤번호자동체의 초기상태로 하면서 순서대로 련결한다. 이때 이행들은 그대로 유지한다.

$f$: $eventfml_1$; $eventfml_2$; … ; $eventfml_n$인 경우는 마지막사건 $eventfml_n$까지 련이어 발생하여야 실패한다.

② 사건식 (3)의 경우

매 *eventfml*에 해당한 부분자동체를 생성하고 초기상태와 최종상태를 하나로 통합한다. 이때 모든 이행은 그대로 유지한다.

③ 사건식 (4)의 경우

모든 *eventfml*에 해당한 부분자동체를 생성하고 이 자동체들의 비동기적인 직적을 취한다.

④ 사건식 (5)의 경우

*eventfml*에 해당한 부분자동체를 생성하고 같은 자동체를 *m*번 련결한다. 앞부분자동체의 최종상태를 뒤부분자동체의 출발점으로 한다.

제약을 적용하는 단계는 다음과 같다.

① $c_i_scop=[i, j]$의 경우 표식 *i*에 대응하는 부분자동체의 최종상태에 들어오는 이행릉들과 표식 *j*에 대응하는 부분자동체의 최종상태에 들어오는 이행릉(최종상태의 자체이행은 제외)들의 이행조건들에 제약 *c*를 론리적한다.

② $c_i_scop=[i, j)$의 경우 표식 *i*에 대응하는 부분자동체의 최종상태에로 들어오는 이행릉들과 표식 *j*에 대응하는 부분자동체초기상태의 자체이행릉의 모든 이행조건에 제약 *c*를 론리적한다.

③ $c_i_scop=(i, j]$의 경우 표식 *i*에 대응하는 부분자동체의 최종상태의 자체이행릉(만일 있다면)과 표식 *j*에 대응하는 부분자동체의 최종상태에로 들어오는 이행릉들의 조건에 제약 *c*를 론리적한다.

④ $c_i_scop=(i, j)$의 경우 표식 *i*에 대응하는 부분자동체의 최종상태에 있는 자체이행릉(만일 있다면)과 표식 *j*에 대응하는 부분자동체초기상태의 자체이행(만일 있다면)의 이행조건들에 제약 *c*를 론리적한다.

## 맺 는 말

시간선도명세서로 모형검사를 위한 속성명세서를 작성하는 한가지 방법을 제안하였다. LTL시제론리의 림시불변속성을 리용하여 사건발생들의 선후차관계와 그의 제약조건만으로 체계동작의 정확성요구를 서술할수 있는 비형식적인 시간선도명세서를 정의하고 그로부터 형식적인 속성명세서에로 변환하는 방법을 제안하였다.

## 참 고 문 헌

[1] G. J. Holzman; The Model Checker, Addison Wesley, 127～147, 2004.

[2] H. A. Gabber; Modern Formal Methods and Application, Springer, 177～196, 2006.

[3] M. B. Dwyer et al.; In Proceedings of the 21st International Conference on Software Engineering, 411～420, 1999.

[4] M. H. Smith et al.; In 5th International Symposium on Requirements Engineering August, 30～38, 2001.

[5] F. K. Holger Giese; LNCS, 4422, 185, 2007.

[6] Ning Ge et al.; http://hal.archives-ouvertes.fr/hal-00675778, 2012.

주체106(2017)년 1월 5일 원고접수

## A Method to Support Specifying Temporal Property for Model Checking

*Sin Chun Ok, Kong Hyon U* and *Sung Nam Chol*

We propose the temporal property specification language called as TimeLineSpec, which is graphical notation that easily learn and use, and have enough expressiveness same as other property specification language. And we introduced a method transform TimeLineSpec to Buchi automata and insert it in promela model of spin.

Key words: model checking, property specification, temporal logic, visual language