김일성종합대학학보
정보과학

주체109(2020)년
제66권 제2호

원천프로그람의 교착상태검사에서 올리통합에 의한 검증모형구축과 증분적인 모형검사

변정룡, 한석민

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

《첨단돌파전을 힘있게 벌려야 나라의 과학기술전반을 빨리 발전시키고 지식경제의 토대를 구축해나갈수 있습니다.》

원천프로그람들의 시험[1]에서는 단위시험을 먼저 진행한 후 그 프로그람의 계층구조에 따라 체계의 통합시험을 올리통합 혹은 내리통합방식으로 진행하였다.

시험의 방법으로 발견하지 못하는 교착상태와 자료경주 등의 오유들을 모형검사방법[2]으로 검출할 때 반드시 모형구축문제와 상태폭발문제가 제기된다.

전형적인 병렬합성에 의하여 구축된 검증모형[3]은 체계를 이루는 매 부분품모형들의 비동기적인 직적으로 되며 이때 상태공간이 부분품모형들의 수의 증가에 따라 지수함수적으로 증가하므로 상태공간의 철저한 탐색에 기초하는 모형검사가 실패할수 있다.

론문에서는 원천프로그람에 교착상태가 있는가에 대한 모형검사에서 모형구축문제와 상태폭발문제를 동시에 해결하기 위하여 검증모형을 프로그람계층구조의 바닥준위로부터 올리통합하는 방법으로 구성하고 모형검사를 단위부분품모형으로부터 체계모형에 이르기까지 증분적으로 진행하는 한가지 방법을 제안하였다.

1. 부분품모형들의 구축방법

검증하려는 원천프로그람에 대하여 깊이가 I이고 폭의 크기가 J인 프로그람계층구조가 주어졌다고 가정하자. 이때 프로그람계층구조를 이루는 클라스들과 메쏘드들에 대한 부분품모형을 일반화추상화를 확장하여 다음과 같이 구축한다.

① 메쏘드/클라스들의 정의를 프로쎄스의 형선언 즉 proctype methodname($p_1, p_2, \cdots, p_n$) 선언으로 번역한다. 이때 국부변수들은 대응되는 proctype methodname($p_1, p_2, \cdots, p_n$)안에 들어가며 그것에 대한 구체적인 실례들에서 리용된다.

② 메쏘드/클라스호출을 다음의 절차로 모의한다.

· 호출하는 메쏘드/클라스에 해당한 proctype callingmethod($p_1, p_2, \cdots, p_n$)선언을 위한 모형형타본문 templatetext의 앞부분에서 호출(run)되는 proctype calledmethod($p_1, p_2, \cdots, p_n$)들의 실례들을 위한 통보문통로들을 미리 정의한다.

· 미리 정의된 통로를 인수로 넘겨 proctype calledmethod($p_1, p_2, \cdots, p_n$)을 정의한다.

· 미리 정의된 통로에서 프로쎄스 calledmethod의 완료를 대기한다.

③ 메쏘드/클라스의 값귀환을 모의하기 위하여 값전송통로를 통한 통신연산뒤에 goto end명령을 덧붙여 프로쎄스 calledmethod를 완료한다. 만일 void귀환이면 값 0을 보내여 그 통로가 동기화를 위해서만 리용되였다는것을 알려준다. 만일 실제값이 귀환된다면 그 값을 통로를 통해 실례프로쎄스 callingmethod($p_1, p_2, \cdots, p_n$)에 전송한다.

구축한 부분품모형의 연산적의미론을 정의하기 위하여 프로쎄스들의 정적모형을 다음과 같이 정의한다.

정의 1 부분품모형의 연산신호기 Σ는

$$\Sigma = \text{Procinforms} \cup \text{Chanels} \cup \text{Vars} \cup \text{Actions} \cup \text{Guards}$$

와 같이 정의된다.

모형을 이루는 프로쎄스정보 $\text{procinform} \in \text{Procinforms}$는

$$\text{procinform} = (\text{procname}, \text{pid}, \text{para}, \text{runningproc}, \text{callingproc}, \text{calledprocs})$$

이다. 여기서 procname은 프로쎄스이름, pid는 프로쎄스실례번호, $\text{para} = (p_1, p_2, \cdots, p_n)$은 프로쎄스의 파라메터렬, runningproc는 프로쎄스의 실행, callingproc는 현재 프로쎄스를 호출하는 프로쎄스, calledprocs는 현재 프로쎄스가 호출하는 프로쎄스들이다.

통보문통로 $\text{chanel} \in \text{Chanels}$는 $\text{chanel} = (\text{chanelname}, \text{mode}, \text{buffers})$이다. 여기서 chanel name은 통로이름, mode는 통신방식 즉 sync/async방식, buffers는 완충기용량이다.

Var는 대역 및 국부변수들의 모임이다.

Actions는 모형에서 수행되는 행위들의 모임(기본명령문들의 모임)이다.

Guards는 술어들의 모임이다.

연산신호기 Σ 우에서 모형서술언어로 작성된 부분품모형의 연산적의미론을 다음과 같이 정의한다.

정의 2 부분품모형 M은 5원쌍 $\langle S, q, \Delta, L_{\text{model}}, l_{\text{model}} \rangle$로 결정되는 비결정성유한상태자동체이다.

상태 $s \in S$는 상태벡토르 (gVars, Procstates, Chanstates, SysVars)에 의하여 결정된다. 여기서 gVars는 대역변수들의 값들의 유한렬, Procstates는 부분품모형을 이루는 능동인 프로쎄스들의 상태벡토르들의 유한렬, Chanstates는 통로들의 상태들의 유한렬, SysVars는 체계변수들의 값들의 유한렬이다.

$q \in S$는 M의 초기상태이다.

L_{model}은 연산신호기 Σ의 원소들과 결합된 모형서술언어이다.

$\Delta \subseteq S \times L_{\text{model}} \times S$는 L_{model}로 표식붙은 이행관계이다.

$l_{\text{model}} : S \times S \to L_{\text{model}}$은 상태이행릉에 대한 표식붙이기함수이다.

2. 올리통합에 의한 검증모형구축과 증분적인 모형검사절차

프로그람의 계층구조가 주어졌을 때 i준위 $(1 \le i \le I)$의 j번째 $(1 \le j \le J)$ 메쏘드/클라스에 대한 부분품모형 $M_{i,j}$들로부터 올리통합의 방법으로 검증모형을 구축하고 통합검사를 하기 위하여 검증보조모형들인 구동모형(driver)과 대용모형(stub)의 개념을 도입한다.

검사하려는 부분품모형 $M_{i,j}$를 호출(run)하는 callingproc$(p_1, p_2, \cdots, p_n)$의 통보문교환과 관련한 동작만을 추상화한 구동모형 $\text{driver}(M_{i,j})$는 외부사건들을 생성하고 호출하려는 부분품모형 $M_{i,j}$를 능동으로 만든 다음 부분품모형으로부터 결과를 받는 동작만을 수행하는 단일프로쎄스모형이다.

정의 3 부분품모형 $M_{i,j}$의 모형검사를 위한 구동모형 $\text{driver}(M_{i,j})$는

$$\text{driver}(M_{i,j}) = \langle S_d, q_d, \Delta_d, \Sigma, l_{\text{model}} \rangle$$

이다. 여기서

$$S_d = \{\text{active, waitcontrol}\}$$

$$q_d = \{\text{active}\}$$

$$\Delta_d = \{\text{active} \xrightarrow{\text{/eventgeneration activate } M_{i,j}} \text{waitcontrol},$$

$$\text{waitcontrol} \xrightarrow{\text{return control}} \text{active}\}$$

이다.

검사하려는 부분품모형 $M_{i,j}$로부터 호출되는 부분품모형들의 동작을 모의하는 모형검사보조모형으로서 대용모형의 개념을 도입한다.

정의 4 부분품모형 $M_{i,j}$의 모형검사를 위한 대용모형 stub$M(M_{i,j})$는

$$\text{stub}M(M_{i,j}) = \langle S_s, q_s, \Delta_s, \Sigma, l_{\text{model}} \rangle$$

인 단일프로쎄스모형이다. 여기서

$$S_s = \{\text{waitevent}\}$$

$$q_s = \{\text{waitevent}\}$$

$$\Delta_s = \{\text{waitevent} \xrightarrow{\text{synceven/delay, returnto } M_{i,j}} \text{waitevent},$$

$$\text{waitevent} \xrightarrow{\text{asyncevent/delay}} \text{waitevent}\}$$

이다.

정의 3과 4에서 $s \xrightarrow{\text{guard/actions}} s'$는 상태이행의 표시로서 $\text{guard} \in \text{Guards}$가 성립할 때 $\text{action} \in \text{Actions}$렬을 수행하면서 목표상태 s'에로 이행한다는것을 의미한다. 원천프로그람의 교착상태에 대한 모형검사에서는 단위부분품모형들에 대하여 검사를 먼저 진행한다.

단위부분품모형들에 대한 검증모형작성과 모형검사절차는 다음과 같다.

① 프로그람의 계층구조를 이루는 모든 메쏘드/클라스 즉 $\forall i, j (1 \le i \le I, 1 \le j \le J)$에 대하여 단위부분품모형 $M_{i,j}$를 작성한다.

② 단위부분품모형 $M_{i,j}$에 대하여 단위검증모형은 $M_{i,j}$와 그것을 호출하는 모형 callingproc를 대신하는 구동모형 driver$(M_{i,j})$와 $M_{i,j}$로부터 호출되는 calledproc들을 대신하는 대용모형 stub$M(M_{i,j})$들로 구성한다.

③ 구축된 매 단위검증모형에 대하여 모형검사를 진행하여 교착상태가 있는가를 검사한다.

매 단위검증모형에 대한 모형검사결과가 모두 교착상태를 가지지 않으면 원천프로그람에 대한 모형검사의 다음단계에서는 올리통합에 의하여 검증모형을 구축하면서 모형검사를 진행한다.

올리통합에 의한 검증모형구축과 모형검사절차는 다음과 같다.

① 프로그람계층구조에서 잎준위$(i = I)$의 부분품모형들에 대하여 모형검사를 진행한다. 이때 검증모형은 하나의 기능수행을 위하여 호상통신하는 부분품모형 $M_{i,j}$들로 형성된 부분품묶음과 그것에 대한 구동모형 driver$(M_{i,j})$로 구성된다. 그 단위검증모형에 대하여 교착상태가 없는가에 대한 모형검사를 진행한다. 계층구조의 정보를 리용하면서

같은 층의 모든 묶음에 대하여 차례로 모형검사를 진행한다.

② 같은 준위의 클라스묶음들에 대한 구축된 검증모형들에 대하여 모두 모형검사를 통과하면 $i=i-1$로 감소시켜 보다 웃준위까지 통합한 검증모형을 구축하고 모형검사를 진행한다. 이를 위해 driver$(M_{i,j})$를 실제의 부분품모형 callingproc로 교체하고 계층나무에서 $M_{i,j}$에 대하여 callingproc를 뿌리로 하는 부분나무에 해당한 부분품묶음과 callingproc를 호출하는 driver$(M_{i,j})$로 검증모형을 구축한다.

③ 검증모형에 대한 모형검사를 진행하고 모형검사결과에 교착상태가 있으면 즉시 모형검사를 끝낸다. 그리고 교착상태가 없으면 $i=1$일 때까지 ②, ③을 반복한다.

단위검증모형과 올리통합된 검증모형의 모형검사에서 생성되는 상태벡토르는 병렬합성에서의 상태벡토르길이에 비해 상당히 작아진다.

맺 는 말

원천프로그람에 교착상태가 있는가에 대한 모형검사에서 검증모형을 프로그람계층구조의 아래준위로부터 올리통합하는 방법으로 구성하고 단위부분품모형으로부터 체계모형에 이르기까지 증분적인 모형검사의 방법으로 모형구축문제와 상태폭발문제를 동시에 해결하였다.

참 고 문 헌

[1] Roger S. Pressman; A Practitioner's Approach, McGraw Hill, 485～498, 2003.

[2] James C. Corbett; ACM Transactions on Software Engineering and Methodology, 9, 1, 51, 2000.

[3] Hao Zheng et al.; IEEE Transactions on Computers, 64, 6, 1607, 2015.

주체109(2020)년 2월 5일 원고접수

A Research on Construction of Verification Model by Bottom-up Merging and Incremental Model Checking in Freedom-deadlock Checking on Source Program

Pyon Jong Ryong, Han Sok Min

In this paper, we constructed verification model by bottom-up merging method and resolved state explosion problem using the incremental model checking method.

Keywords: model checking, state explosion control, model construction