

양상논리를 포함한 BML 변환을 위한 BIRS의 확장

The extension of BIRS to Translate the BML with Modal Logic

이상협, 김선태, 김제민, 박준석, 유원희
인하대학교

Lee sang-hyup, Kim seon-tae, Kim je-min,
Park joon-seok, Yoo weon-hee
Inha Univ.

요약

프로그램의 정적 검증을 위한 명세는 1차 술어 논리(First Order Logic)가 주로 사용된다. 하지만 1차 술어 논리가 모든 정보를 표현할 수가 없기에 이를 보완하기 위해 양상논리(Modal Logic)를 사용할 수가 있다. 정적 프로그램 검증을 위해 양상 논리를 이용하여 확장된 BML(Bytecode Modeling Language)은 BIRS로 변환 되어야 한다. 본 논문에서는 확장된 BML을 중간 표현 언어인 BIRS(Bytecode Intermediate Representation Specification)로 표현하기 위하여 BIRS 문법을 확장한다.

I. 서론

자바컴파일러는 컴파일시 대부분의 오류를 판단하지만 판단하지 못하는 오류 또한 존재한다. 그러한 오류는 프로그램 실행 시에 치명적인 결과를 초래한다. 그렇기에 프로그램 실행 전에 오류를 검출하는 정적 프로그램 검증이 필요하다.

프로그램 검증을 위한 연구는 다양한 방법으로 이루어졌다. 제안된 방법으로는 자바 바이트 코드에 명세를 추가한 BML(Bytecode Modeling Language)[1]이 있다.

BML은 JML2BML 프로그램을 통해서 만들어진 이진코드이며 명세 정보를 BML스펙에 부합되어 생성된 언어이다. 바이트 코드와의 차이점은 2차 상수풀이 존재해 명세 정보 추출이 가능하다.

검증엔 1차 술어 논리가 주로 사용되지만 표현의 한계가 존재해 양상논리를 추가해 보완할 수 있다. 양상논리는 술어 논리에 시간적 개념을 추가하여 시간을 논리적으로 표현을 가능하게 하였다. 따라서 BML에 양상논리를 추가하면 더 많은 논리를 표현할 수 있기에 확장된 BML이 필요하다[2].

프로그램 검증을 위해서 확장된 BML을 중간언어 형태인 BIRS(Bytecode Intermediate Representation Specification)[3]로 변환 후 SMT-SOLVER를 이용할 수 있다.

BIRS는 자바 바이트 코드를 SMT-SOLVER에 입력을 하기위해 제안된 중간언어이다.

본 논문에서는 확장된 BML을 BIRS로 변환 하기 위한 BIRS의 문법 확장에 대해 설명한다. 2장에서는 확장된 BML을 BIRS로 표현하기 위한 문법을 설명한다. 3장에서는 결과 및 향후 연구 과제를 설명한다.

II. 확장된 BML에 대응하는 BIRS문법 확장

1. 양상논리가 추가되어 확장된 BML 스펙

표1은 BML에 양상논리를 추가해 확장된 BML의 formula_info의 내용이다. Next, Strong until, Weak until, Always, Eventually 이 5가지의 연산자는 양상 논리에서 사용되는 연산자이다. Next의 의미는 다음 상태에서는 참이 된다. Always의 의미는 항상 참이다. Eventually는 프로그램이 종료되지 않는다는 가정하에 어떤 상태까지 언젠가는 도달한다. Strong Until과 Weak Until은 뒤의 연산자가 참인 동안에는 앞의 피연산자가 참이라는 의미이다.

표 1. Formula_info 내용

.....
Next
Strong until
Weak until
Always
Eventually
.....

2. 확장된 BML 변환을 위한 BIRS문법확장

그림1은 BIRS에서 양상논리를 표현하기위해서 BIRS의 command에 modal을 추가하였다. modal을 문법에 추가함으로써 프로그램 검증시 양상논리로 명세가 되어있음을 제시하여 적합한 절차로 검증이 가능하다.

```

Command ::= Index : instr
          | assert LogicalExpr
          | assume LogicalExpr
          | modal LogicalExpr

```

▶▶ 그림 1. 확장된 BIRS의 Command

그림2는 표1의 내용을 추가하기 위하여 BIRS의 LogicalExpr에 논터미널인 ModalExpr을 추가하였다. ModalExpr은 5개의 터미널인 next, always, eventually, strongUntil, weakUntil을 가지고 있다. BIRS의 논리식 표현에 비단말인 ModalExpr을 추가한 이유는 가독성의 향상을 도모하기 위해서다.

```

LogicalExpr ::= T | F
              | ArithExpr Relop ArithExpr
              ...
              | ModalExpr

ModalExpr ::= next LogicalExpr
             | always LogicalExpr
             | eventually LogicalExpr
             | LogicalExpr strongUntil
               LogicalExpr
             | LogicalExpr weakUntil
               LogicalExpr

```

▶▶ 그림 2. 양상논리가 추가된 BIRS의 LogicalExpr

next, always, eventually은 LogicalExpr을 피연산자로 하며 연산자 뒤에 위치한다. strongUntil과 weakUntil은 앞과 뒤에 피연산자인 LogicalExpr을 둔다.

next는 LogicalExpr이 다음 상태에는 참이 된다는 의미이다. always는 LogicalExpr이 항상 참을 나타내며, eventually은 LogicalExpr이 프로그램이 종료되지 않는다는 가정하에 언젠가 도달함을 의미한다. strongUntil과 weakUntil은 뒤의 LogicalExpr이 참인 한 앞의 LogicalExpr은 참을 표현한다.

위의 문법을 통해서 양상논리로 명세가 추가된 확장된 BML을 검증이 가능하다. modal LogicalExpr에서 양상논리로 명세가 되어있음을 알 수가 있으며 ModalExpr은 LogicalExpr에서 5가지의 키워드를 통해 양상논리 연산자를 알 수가 있다.

Ⅲ. 결론 및 향후연구과제

본 논문은 확장된 BML을 중간언어인 BIRS로 표현하기 위하여 BIRS의 문법을 확장하였다. 기존에 있던 LogicalExpr에 ModalExpr을 추가하여 양상논리를 표현 가능하게 하였다.

향후 연구 과제로는 위에서 정의된 문법을 통하여 BML

형태의 프로그램을 BIRS로 변환하고 변환된 BIRS를 이용해 프로그램 검증을 수행할 예정이다.

■ 논문 사사(acknowledgement) ■

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (No. 2011-0026495)

■ 참고 문헌 ■

- [1] Lilian Burdy, Marieke Huisman and Mariela Pavlova, "Preliminary Design of BML: A Behavioral Interface Specification Language for Java bytecode," In Fundamental Approaches to Software Engineering (FASE 2007), pp. 215-229, Volume 4422 of Lecture Notes in Computer Science, Springer-Verlag, 2007.
- [2] 김선태, 김제민, 박준석, 유원희, "양상논리를 포함한 BML 스펙 확장", 한국정보처리학회 춘계학술대회 논문집, 제19권, 1호, pp.265-268, 2012.
- [3] 김선태, 김제민, 박준석, 유원희 "바이트 코드 검증을 위한 스택리스 중간표현 설계", 한국정보처리학회 춘계학술대회 논문집, 제18권, 1호, pp265-268, 2011.
- [4] Amir Pnueli, "The Temporal Logic Of Programs" University of Pennsylvania and Tel-Aviv University, 1977.