

# 모델 체킹을 이용한 JUnit 기반 테스트 드라이버 자동 생성<sup>1)</sup>

이태훈<sup>0</sup>, 권기현  
 경기대학교 정보과학부  
 {taehoon, khkwon}@kyonggi.ac.kr

## Automatic generation of JUnit based test driver using model checking

Taehoon Lee<sup>0</sup>, Gihwon Kwon  
 Department of Computer Science, Kyonggi University

### 1. 서 론

소프트웨어 테스트는 소프트웨어의 오류를 찾기 위해 많이 사용되고 있다. 소프트웨어 테스트를 수행하기 위해서는 테스트 케이스를 잘 작성하는 것이 중요하다. 하지만 소프트웨어의 크기가 커질수록 테스트 케이스를 작성하는데 필요한 시간이 증가하고 많은 자원이 필요하다. 따라서 테스트 케이스를 자동으로 생성하는 연구가 계속 진행되어 왔다. 많은 연구에서는 테스트 입력 데이터에 대한 자동생성에 대해서 연구가 진행되지만 아직까지 테스트 케이스 자동생성과 테스트 케이스를 실행할 수 있는 테스트 드라이버의 자동생성에 대한 연구는 진행 되어 있지 않다. 본 연구에서는 자바 언어를 대상으로 기호실행을 이용해서 조건 커버리지를 만족하는 테스트 입력 데이터를 생성하고 테스트 입력 데이터로부터 실제 가능한 결과 값을 계산하고 대표적인 유닛 테스트 도구인 JUnit에서 동작하는 테스트 드라이버를 자동 생성 방법에 대해서 연구 한다.

### 2. 관련 연구

#### 2.1 기호 실행

기호실행[1]기본 아이디어는 실제 값 대신에 기호적으로 표현된 값을 사용하는 것이다. 결과적으로 프로그램에서 계산된 결과는 기호적으로 표현된 값으로 표현된다. 기호적으로 실행된 프로그램의 상태에는 프로그램의 변수값, 경로 진행 조건, 프로그램 카운터가 포함된다. 경로조건은 기호적으로 표현된 값을 이용해서 어떤 경로로 프로그램이 실행되어야 하는지를 나타내는 이진식이다. 프로그램 카운터는 다음에 실행될 프로그램의 구문을 나타낸다.

### 3. 테스트 드라이버 자동 생성

본 논문에서 제안하는 JUnit을 위한 테스트 드라이버 자동 생성 단계는 그림 1과 같다.

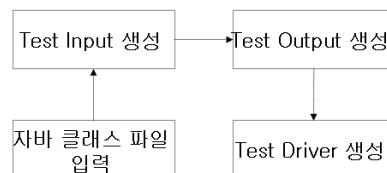


그림 1 자동 생성 진행 단계

1) 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음[2007-081-7, 효과적인 디지털 콘텐츠 서비스를 위한 상황 인식 정보 처리 기술

JUnit에서 동작하는 테스트 드라이버 자동 생성을 위해 우선 주어진 자바 프로그램에서 가능한 테스트 입력 데이터를 자동 생성해야한다. 테스트 입력 데이터 자동 생성을 위해 기호 실행을 이용하였다. 기호 실행을 이용하여 주어진 자바 프로그램에서 하나의 메소드에서 가능한 모든 입력 데이터를 생성한다.

기호실행을 이용한 자바 프로그램의 테스트 입력 데이터 생성 방법은 자바 프로그램의 클래스 파일을 받아들여서 클래스 파일의 각각의 구문(instruction)을 실행하면서 경로 조건을 기호적으로 표현한다. 만일 하나의 종료되는 경로를 만났을 경우 최근 방문했던 분기 구문으로 되돌아가서 새로운 경로진행 조건을 생성하고 다른 경로를 탐색한다. 만일 프로그램의 모든 분기 구문에 대해서 참이 되는 부분과 거짓이 되는 부분에 대한 경로를 탐색하고 각각의 경로에 대한 경로 진행 조건을 생성하였다면 각각의 경로로 이동하기 위한 경로진행 조건을 구체적인 값으로 변환한다.

#### 4. 사례 연구

자바 표준 라이브러리인 Vector 의 remove 메소드에 대해서 적용해보았다. Vector 는 길이를 자동으로 증가시켜주는 배열이다. 그중에 remove 메소는 중간에 있는 특정 인덱스의 값을 제거하고 그 뒤에 있는 값들을 변경해주고 지워진 값을 되돌려준다.

```
public synchronized E remove(int index) {
    modCount++;
    if (index >= elementCount)
        throw new ArrayIndexOutOfBoundsException(index);
    Object oldValue = elementData[index];
    int numMoved = elementCount - index - 1;
    if (numMoved > 0)
        System.arraycopy(elementData, index+1, elementData,
            index, numMoved);
    elementData[--elementCount] = null;
    return (E)oldValue;
}
```

그림 2. 벡터 클래스의 소스코드

Vector 클래스의 remove 메소드를 자동으로 테스트 드라이버를 생성할 경우 각각의 분기문을 통과하는 총 4개의 테스트 케이스를 생성하고 4개의 테스트 케이스를 실행 할 수 있는 테스트 드라이버를 자동 생성해 준다.

#### 5. 결론

본 논문에서는 유닛 테스트를 위한 자동화된 테스트 케이스 생성기법과 테스트 드라이버 생성 기법을 설명한다. 이를 통해서 실제 자바 라이브러리에 포함된 Vector 클래스의 메소드들에 대해서 테스트 드라이버를 생성할 수 있었다.

하지만 복잡한 자료구조의 경우 올바른 테스트 드라이버를 생성하기 못한다. 향후 연구로 복잡한 자료구조에 대해서도 테스트 케이스를 효율적으로 생성하는 기법이 필요하다. 또한 다른 자바 프로그램에 대해서도 적용해야한다. 현재 사용된 제약 해결기 이외에 다른 제약 해결기와 결정 프로시저를 이용하여 좀 더 좋은 성능을 얻을 수 있어야 한다.

#### 참고문헌

- [1] J.C. King, "Symbolic Execution and program testing" Communication of ACM, 19(7):385-394, 1976.