Concolic 테스팅 도구 CREST 의 사용자 친화성 향상 연구:

Windows OS 로의 포팅과 개선된 CREST UI 을 통한
CREST 활용 및 분기 커버리지 분석 작업의 효율 증가

김현우 ○ 김윤호 김문주

한국과학기술원

hyunoo@kaist.ac.kr yunho.kim03@gmail.com moonzoo@cs.kaist.ac.kr

Improving Applicability and User Interface of CREST :

Porting CREST to Windows Platform and Improving Work Efficiency
for Coverage Analysis

Hyunwoo Kim○ Yunho Kim Moonzoo Kim

Korea Advanced Institute of Science and Technology

요 약

본 연구에서는 Concolic 테스팅 기법 도구인 CREST 를 포팅하여 기존에는 불가능했던 Windows OS 에서도 동작이 가능하도록 확장했다. 추가로, 기존에는 CREST 가 달성한 분기 정보들을 사용자가 확인하려면 추가로 해석하는 작업이 필요했으나, CREST 의 기존 출력을 변경하고 분기 정보 확인/분석에 유용한 정보들을 추가로 제공하도록 개선하였다. 이로 인해 CREST 를 사용한 테스팅 과정에서 사용자의 작업 부담을 줄이고, 작업 효율을 늘리고자 하였다.

1. 서 론

소프트웨어의 늘어나는 필요성과 요구사항으로 인해 그 복잡성이 높아짐에 따라 자동화 테스팅에 대한 중요성 역시 높아지고 있다. 테스팅의 목적은 소프트웨어 내부에 존재하는 에러를 검출하는 것으로, 이를 달성하기 위해서 소프트웨어의 가능한 많은 path 탐색을 필요로 한다.

테스팅 기법 중 하나인 Concolic 기법[1]은 소프트웨어를 실행한 뒤, 실행한 path 을 기반으로 새로운 path 을 탐색하는 테스트를 생성하고, 이 테스트로 타겟 소프트웨어를 다시 실행하는 일련의 과정을 반복한다. 이론적으로 Concolic 기법을 사용하면 타겟 소프트웨어의 모든 path 을 탐색하는 것이 가능하다.

CREST[2]는 C 프로그램용 Concolic 기법 도구이다. CREST 는 현재 Linux 환경만 지원하고 있기 때문에 Linux 환경에서 동작하지 않는 소스 코드를 대상으로 CREST 를 실행하기 위해서는 Linux 환경에서 동작하는 소스 코드로 변환하는 추가 작업이 필요한 상태이다. 또한 CREST 실행 결과의 분석에 필요한 정보들은 사용자가 곧바로 알아볼 수 없는 형태이기 때문에 추가로 해석하는 작업이 필요하다.

본 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(NRF-2016R1A2B4008113), 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대 정보 컴퓨팅기술개발사업의 지원(NRF-2017M3C4A7068177), 정부(교육부)의 재원으로 한국연구재단의 지원(NRF-2017R1D1A1B03035851)을 받아 수행한 연구임.

본 논문에서는 CREST 을 사용하고, 결과를 분석하는데 필요한 노력과 시간을 줄여서 효율적인 CREST 사용이 가능하고 생산성을 높이고자 하였다. 본 논문에서 수행한 작업은 구체적으로 다음과 같다. 첫 번째로, 가장 많이 사용되는 개발 환경 OS 중 하나인 Windows OS 에서 CREST 사용이 가능하도록 CREST 포팅을 수행하였다. 두 번째로, CREST 결과 분석에 필요한 정보들을 사용자가 쉽게 이해할 수 있는 형태로 추가 제공하여 결과 분석에 필요한 추가 비용을 크게 줄이고자 했다.

2. CREST 소개

CREST 는 Concolic 기법(Concrete + Symbolic execution)을 사용하는 C 프로그램 용 테스팅 도구이다. CREST 는 먼저 Concrete execution 을 통해 Symbolic execution 을 유도한다. Concrete execution 은 입력 변수에 값을 넣고, 타겟 프로그램의 한 path 를 실행시킨다. Symbolic execution 은 입력 변수를 심볼로 가정하고, Concrete execution 에서 실행한 path 에 도달하기 위해 입력 변수가 만족해야하는 심볼릭 경로 수식(symbolic path formula)을 수집한다. 수집한 심볼릭 경로 수식을 수정하여 아직 실행하지 못한 path 에 도달하기 위한 새로운 심볼릭 경로 수식(next formula)을 만들어낸다. 심볼릭 경로 수식을 수정하는 방법은 여러가지가 존재하며 기본적인 “DFS” 방법의 경우 마지막 수식에 negation 을 취해서 새로운 심볼릭 경로 수식을 획득한다. Z3 solver[3]에 새롭게 구한

심볼릭 경로 수식을 전달하면 심볼(입력 변수)의 해(값)를 구할 수 있다. 새롭게 구한 입력 변수 값으로 다시 타겟 프로그램을 실행시키고, 또 다른 입력 변수의 값을 추출하는 일련의 과정을 반복하면서 프로그램의 path 들을 탐색하게 된다.

CREST 는 Instrumentation 작업이 선행 되야 하는데 이 작업은 심볼릭 경로 수식 기록에 필요한 probe 들을 코드에 삽입해주는 작업이다. 특히, 분기마다 __CrestBranch 함수 호출 코드가 삽입되며, 인자 중 하나로 분기 식별이 가능한 분기 id 를 가진다. CREST 를 실행하면 달성한 분기 id 가 "coverage"파일에 기록되는데, __CrestBranch 의 인자인 분기 id 와 "coverage"파일에 기록 된 분기 id 를 매칭시켜서 분기 달성 여부, 위치 등의 파악이 가능하다.

3. 사용자 친화성 향상 기법

3.1 Windows 포팅

CREST 가 개발 환경에서 널리 사용되는 Windows OS 를 지원하지 못하여 초래되는 문제와 비용 비효율성을 없애고자 Windows OS 에서도 CREST 의 사용이 가능하도록 포팅하는 작업을 수행했다. 아래 표 1 은 포팅 과정에서 변경 된 주요 구문들이다.

표 1 포팅 시 변경 된 주요 구문

구성 요소	주요 변경 구문
Front-end	1 CIL 에 의해 파싱이 불가능한 구문을 제거하는 코드 추가. 2 Visual studio 컴파일러가 지원하지 않는 GCC 확장 구문 생성 방지.
Middle-end	1 Visual studio 컴파일러가 지원하지 않는 CREST 코드 안의 GCC 확장 구문 제거. 2 Visual Studio 가 지원하지 않는 POSIX 라이브러리 대체.
Back-end	1 Visual Studio 가 지원하지 않는 POSIX 라이브러리 대체. 2 Visual Studio 가 지원하지 않는 출력 형식 대체.

3.2 CREST 달성/미달성 분기정보 UI 향상

CREST 도구에 이미 탑재 되어있는 print_execution 명령어의 결과를 사용자 친화적으로 변경하였고, 커버리지 결과 분석에 필요한 추가 정보가 담긴 파일 "new_coverage"를 생성하는 새로운 명령어 gen_new_cov 를 추가하였다.

3.2.1 print_execution

print_execution 은 CREST 에 내재된 명령어로, CREST 로 프로그램을 실행시킨 후 사용 가능한 명령어이다. 본 명령어 사용 시 CREST 가 타겟 프로그램을 실행했을 때 사용 된 심볼릭 변수가 갖는 값, 심볼릭 경로 수식, 실행 된 분기 id 가 출력된다. 이러한 정보들을 바탕으로, 심볼릭 변수가 갖는 값과 그에 대응되는 심볼릭 경로 수식을 확인할 수 있다. 개선 된 사항은 다음 1 - 4 항목과 같다.

1. 가명 심볼릭 변수명(x0, x1, x2, …)이 아닌 실제 변수명 출력.
2. 심볼릭 경로 수식 출력을 기존 전위(pre-order)방식에서 사용자에게 친숙한 중위(in-order) 방식으로 변경.
3. 형 변환 표기 시, 실제 형 변환 된 자료형으로 출력.
4. 각 정보 출력 시, 위치 정보(line, file)를 함께 표기하는 방식으로 변경.

3.2.2 gen_new_cov

gen_new_cov 는 새로 추가 된 스크립트로서, 실행 시 CREST 가 달성한 상세 분기 정보가 담긴 파일 "new_coverage"을 생성한다.

분기 id 를 소스 코드와 매칭시켜볼 필요 없이 new_coverage 파일에는 분기에 대한 상세 정보(분기 id, 분기식, 달성 여부, 분기 위치 정보)와 분기의 동향(파일 별, 함수 별 분기 커버리지)을 파악할 수 있는 분기 요약 정보를 추가했다. 따라서, 기존 coverage 파일의 분기에 대한 정보 부족으로 인한 사용자의 추가 수행 작업을 줄일 수 있었다.

4. 결론 및 향후 연구

본 논문에서는 Linux 환경에서만 동작이 가능했던 CREST 를 Windows OS 에서도 사용이 가능하도록 확장하는 작업을 수행했다. 지금까지는 Windows OS 에서만 실행 가능한 C 소스 코드에 대해 CREST 적용을 포기하거나, Linux 환경에서 동작가능하도록 변환하는 작업을 거치고 난 뒤 OS 간의 동작 차이를 감안하고 CREST 를 적용해야 했다. 하지만 본 연구에서 수행한 포팅 작업으로 인해 Windows OS 에서 동작하는 소스 코드에 대해서도 쉽게 CREST 적용이 가능해졌다.

두번째로 기존 CREST 에서 제공했던 print_execution 의 출력 메시지를 개선하고, 커버리지 분석에 필요한 추가 정보가 담긴 new_coverage 를 사용자에게 제공하여 테스팅에 필요한 시간을 크게 줄이고자 하였다.

향후 연구로써 현재 포팅 된 CREST 는 Windows OS 의 이점인 GUI 특성을 살리도록, 현재 프롬프트 창에서 명령어로 동작하는 CREST 에 GUI 인터페이스를 추가할 계획이다.

참 고 문 헌

[1] K. Sen, D. Marinov, G. Agha, "CUTE: a concolic unit testing engine for C," in *ACM SIGSOFT Software Engineering Notes*, 2005.

[2] J. Burnim and K. Sen, "Heuristics for scalable dynamic test generation," in *ASE*, 2008.

[3] L. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in *TACAS*, 2008.