QnA*

아이락5 Airac5 愛樂5

RopasWork Inc.
Programming Research Laboratory
Seoul National University

Q: Airac5는 무엇인가?

A: "Array Index Range Analyzer for C"의 약자로 프로그램 분석 기술을 사용해서 메모리 접근 오류(buffer overrun)가 일어날 수 있는 C 소스의 위치를 미리 모두 자동으로 찾아주는 소프트웨어 시스템이다.

C 프로그램에서 메모리 접근은 항상 할당된 메모리의 내부에 국한되야 한다. Airac5는 주어진 C 프로그램의 모든 실행 상황을 분석해서, 할당된 메모리를 벗어나서 접근하는 경우(buffer overrun)들을 미리 모두 찾아준다.

Q: 테스트와 어떻게 다른가?

A: 테스트의 문제점들을 보완해 준다. 테스팅은 프로그램을 실행시켜야 하고, 몇개의 입력에 대해서만 제대로 작동된다는 것을 확인할 수 있을 뿐이다. 가능한 입력이 무수히 많다면 테스트에서 제외된 입력이 있을 수 있고, 이 경우에 오류를 발생시킬 수 있는 가능성을 체크할 수 없다. 또, 테스트는 프로그램을 돌릴 수 있는 환경이 갖추어질 때 까지 기다려야 한다.

Airac5는 대상 프로그램을 실행시키지 않으면서 찾고자 하는 오류들은 모두 찾아준다. 프로그램의 소스만 있으면 된다.

특히, 그 확인 과정이 또 다른 소프트웨어(프로그램 분석기)를 통해서 완전히 자동으로 이루어진다. 분석기 소프트웨어에 입력으로 들어가는 것은, 검증할 소프트웨어 소스이다.

Q: 핵심 기술이 무엇인가?

A: 정적 프로그램 분석 (static program analysis) 기술이다. 이 기술은 주어진 프로그램의 모든 실행상황을 실행하기 전에 미리 엄밀하게 확인하는 기술이다.

정적 프로그램 분석(static program analysis) 기술은 다양한 이름으로 다양한 수준에서 다양한 필요에 맞추어 불리워지는 기술들을 모두 포섭한다: "static analysis", "abstract interpretation", "type system", "software model checking", "data-

*For expert C programmers.

flow analysis", "program logics and proof system" 등.

이 기술은 지난 30년 동안 연구가 무르익어 이제 비로서 실제 오류검증에 적용되기 시작하였다. 우리는 지난 10년 이상 이 분야 연구를 수행해왔고, 지난 해 부터 상용을 목표로 개발해 온 시스템이 AIRAC5이다. 요약해석(abstract interpretation) 기술을 기반으로 한다.

Q: Valgrind나 Rational PurifyPlus와는 어떻게 다른가?

A: Valgrind나 PurifyPlus는 실행시키면서 오류를 찾아주는 도구들이다. 테스트를 통해 오류를 찾는 것과 같다. "runtime program analysis"라고 한다. 테스트의 문제점들을 고스란히 가진다.

AIRAC5는 실행시키지 않고 모든 오류들을 소스를 분석해서 찾아준다. "static program analysis"기술의 특징이다.

Q: 그 기술에 기반한 경쟁 제품들과 비교하면?

A: 정적 프로그램 분석기술에 기반해서 오류 자동검증기에 특화된 회사들은 대표적으로 두개가 있다. 프랑스 École Polytechnique 출신의 `polyspace.com`과 미국 스탠포드대 출신의 `coverity.com`이다.

Coverity의 제품은 목표한 오류(buffer overrun)를 모두 찾아주지도 못하고 허위경보도 있다. Polyspace의 제품은 분석비용이 크고 허위경보가 많다. AIRAC5는 Coverity가 찾아내는 오류를 포함해서 Coverity가 놓치는 것까지 모두 찾아주고, Polyspace보다는 분석비용과 허위경보가 적다.

`ropas.snu.ac.kr/airac5`를 방문하면 AIRAC5 자체의 성능뿐 아니라 Coverity와의 비교 성능을 살펴볼 수 있다.

Q: 제한점이나 숨은 비용은?

A: 대상 오류들이 있으면 모두 찾아주지만, 실제 오류가 아닌데도 오류라고 판단하는 경우(false alarm)가 있다. 예를들어, 실제 오류가 10군데라면 그 장소들을 포함해서 15군데를 찾아주는 격이다.

이러한 허위경보를 0개로 줄이는 것은 불가능하다. 오류라고 판정된 지점마다 허위경보인지 아닌지를 프로그래머가 확인하는 과정은 필요하다.

Q: C++ 표준 라이브러리나 Java에서 실행중 예외를 발생시키는 방식과 어떻게 다른가?

A: 그 방식은 에러가 발생하는 것을 미연에 방지하는 것이 아니라 실행 중에 에러가 발생했을 때 예외를 발생시키고 프로그래머는 그러한 예외상황을 잘 처리할 수 있도록 프로그램해야 한다.

또한, 그런 방식은 안전성을 보장하기 위해 실행 중에 메모리 접근 범위를 매번 검사하기 때문에 프로그램의 전체 실행 속도가 느려진다.

Q: malloc으로 할당된 메모리 공간에 대해서도 분석이 이루어지는가?

A: 물론이다. 배열로 선언되지 않은 할당된 메모리 공간들에 대해서도 그 범위를 벗

어나는 오류(buffer overrun)를 찾아주며, 메모리 공간에 접근하는 다양한 방법에 대해서도 모두 처리한다.

Q: ANSI C로 짜여진 모든 프로그램을 다 지원하는가?

A: 모든 ANSI C 프로그램을 지원한다.

Q: 데몬(demon) 프로그램처럼 실행이 끝나지 않는 프로그램에 대해서도 사용할 수 있는가?

A: Airac5는 실행이 끝나지 않는 프로그램에 대해서도 유한한 시간 내에 항상 분석을 마친다.

Q: 인터럽트 핸들러(interrupt handler) 를 사용하는 경우도 분석 가능한가?

A: 그렇다. 해당 C 소스들이 모두 있기만 하면 된다.

Q: 병렬 쓰레드(thread)들을 가지는 C 프로그램도 분석 가능한가?

A: 제약이 있다. 병령 쓰레드들이 서로 교차(dependence/interference)하면서 생길 수 있는 메모리 접근 오류는 찾지 못할 수 있다. 그러나, 개별 쓰레드가 독립적으로 가질 수 있는 오류는 모두 찾는다.

Q: 입력값이나 외부 자료에 의해 좌우되는 메모리 접근은 어떻게 처리되나?

A: 입력값이 정의되지 않았다면 가능한 모든 경우를 고려한 분석결과를 내놓는다. 만약 올바르지 않은 입력값이 들어왔을 때 그 값을 사용하지 않게 하는 루틴이 포함되어 있다면 올바른 입력값으로 판정되어 실제 수행되는 경우만을 고려하여 분석한다.

Q: 분석 속도는 얼마나 되는가?

A: 대게 컴파일러 속도의 50배 이내고, 더 많은 시간이 걸리는 경우도 있다. 분석 속도는 컴파일 속도와 달리 프로그램 소스의 크기에 비례하지 않고 논리적 복잡도에 비례한다. 따라서, 작은 소스이지만 분석이 오래걸릴 수 있고 크지만 빨리 끝날 수 있다.

`ropas.snu.ac.kr/airac5`를 방문하면 구체적인 성능 데이타를 살펴볼 수 있다. 그곳에서 온라인 데모를 해 볼 수도 있고, 실제로 내려받아 사용해 볼 수도 있다.

Q: 전체 프로그램의 크기가 크다면 사용하기 힘들지 않나?

A: 컴파일러가 모듈화를 지원하기 위해 개별 파일에 대해 컴파일할 수 있듯이 Airac5도 개별 파일 단위로 검사할 수 있다.

Q: 얼마나 큰 파일까지 분석할 수 있는가?

A: 분석 과정에서 메모리의 제약이 있으나 인텔 펜티엄4 3.2GHz 4GB RAM 환경에서 약 1만 라인의 소스파일을 한 번에 분석 가능하다. 대형 프로그램의 경우 각 개별 파일, 또는 소스의 부분 별로 분석을 수행할 수 있기 때문에 사실상 큰 제약은 없다.

Q: 그렇다면, 여러 파일로 구성된 프로그램 소스를 그대로 사용할 수 있다는 말인가?

A: 일반적으로 그렇다. Makefile이 있다면 AIRAC5이 알아서 필요한 파일을 찾아 main함수부터 호출을 시작하게 된다. 만약 main함수가 없는 경우에는 간단한 호출 시나리오를 main함수로 구현할 필요가 있다. `Makefile`이 없다면 `*.c`파일들을 전처리 한 후의 소스코드를 입력으로 주면 된다.

Q: Airac의 분석결과 오류로 지적된 부분이 실제로는 오류가 아니었다. 어떠한 현상인가?

A: 모든 상황을 유한하게 포섭하다보면, 오류인지를 확실히 알 수 없는 경우가 있다. 이러한 경우에는 안전한 결론을 위해서 오류가 있는 것으로 판별한다.

Q: AIRAC5가 잡아내는 오류 경보중에서 그러한 허위 경보(false alarm)를 어떻게 구별할 수 있는가?

A: 완벽하게 구별하는 것은 불가능하다. 대신에, 오류 경보들이 진짜일 확률을 계산해서 가장 높은 확률을 가진 오류 경보들부터 사용자가 살펴보도록 한다.

공개버젼에서는 찾아진 오류들 중에서 진짜 오류일 확률이 가장 높은 상위 5개의 오류만 보여준다. 풀 버젼은 모든 오류들을 진짜일 확률의 순서대로 보여준다.

Q: 시험적으로 사용 가능한가?

A: `ropas.snu.ac.kr/airac5` 에서 실행화일을 내려받아 사용해 볼 수 있다. 온라인 데모도 그곳에서 제공된다. 공개된 버젼은 실제 버젼보다 제한점이 있다. 분석할 수 있는 소스의 크기가 프로시져 100개 이내여야 하고, 분석결과 찾아진 오류들중에서 5개만 알려준다.

Q: 완전한 버젼을 사용하려면?

A: 다양한 라이센스 계약을 통해 구매할 수 있다. AIRAC5 자체뿐 아니라 장기적인 "whole product service"를 구매할 수 있다.

□