지향성 퍼징: 선택과 집중의 미학

Tae Eun Kim

2021.10.21

Abstract

퍼징은 효과적으로 프로그램 전체를 탐색하는 테스팅 기법이지만 프로그램의 특정 부분만 테스트하는 용도로는 적절하지 못하다. 이에 제안된 것이 기존의 무지향성 퍼징과 차별화된 지향성 퍼징이다. 지향성 퍼징은 사용자가 원하는 지점까지 도달하는 입력을 생성하는 것이 목적을 가지며 다양한 상황에서 활용될 수 있다. 이 글에서는 Def-use Graph를 통해 기존의 지향성 퍼징 기술보다 더 탐색 범위를 좁히는 방법을 제안하고자 한다.

퍼징은 무작위한 입력을 생성하여 프로그램을 전반적으로 테스트하는 기술이다. 대상 프로그램에 대한 정보가 거의 없는 상황에서도 프로그램을 효과적으로 관통하는 입력을 만들어내는 등, 퍼징은 강력한 성능으로 인해 많은 사랑을 받아왔다. 그러나 은빛 총알(Silver Bullet)은 없다는 말[1]처럼 퍼징도 모든 상황에 적합한 만능 도구는 아니다. 별다른 정보 없이 무작위성에 기대어 프로그램의 모든 부분을 탐색하려다 보니, 검사하고 싶은 특정한 위치가 있는 경우에는 퍼징이 적절하지 않다고 볼 수 있다.

이러한 기존의 퍼징, 즉 무지향성 퍼징과 차별화된 기법이 바로 지향성 퍼징이다. 지향성 퍼징의 목적은 사용자가 원하는 지점까지 도달하는 입력을 빠르게 생성하는 것이다. 코드를 새로 추가하거나 수정했을 때, 혹은 결함위치추정 결과가 있을 때는 일부의 코드만 우선적으로 검사해야할 필요가 있다. 하지만 전체 코드에서 원하는 부분을 실제로 실행하는 입력을 만들어내기 어려울 수 있다. 지향성 퍼징은 이런 상황에서 효과적으로 필요한 입력을 생성하고 원하는 코드를 시험할 수 있도록 한다.

기존의 지향성 퍼징 기술들은 Control Flow Graph를 활용하는 등의 방식으로 목표하는 지점에 도달하였다. 하지만 Control Flow Graph는 프로그램의 문법적인 구조만 나타내기 때문에 목표하는 지점과 실제로는 상관이 없는 지점들을 구분할 수 없다. 따라서 정적분석을 수행하여 Def-use Graph를 활용한다면, 각 지점에 실제로 영향을 주는 프로그램의 위치들을 알 수 있고, 이를 통해 더욱 빠르고 효과적으로 목표 지점에 도달할 수 있을 것이다.

[1] Brooks, F., & Kugler, H. (1987). No silver bullet