

Project

SNU 4541.664A

Kwangkeun Yi

- 이 프로젝트의 목적은 강의에서 다루는 정적분석 이론들(요약해석 abstract interpretation)을 익힐때 구체적인 예시가 되는 것이다.
- 이론 강의중에 실제 예시로서 이 프로젝트의 내용을 다루게 된다.
- 숙제들은 이 프로젝트를 실현해가는 것으로 진행될 것이다.

대상언어 D

D라고 부르기로한 아래의 언어를 생각하자. 정수입력을 받고, 변수의 주소도 값으로 다룰 수 있도록 했고, 타겟이 실행중에 계산되는 점프명령이 있다.

$$\begin{aligned} C &\rightarrow x := E \mid *x := E \\ &\mid C ; C \\ &\mid \text{if } E C C \\ &\mid \text{repeat } C E \\ &\mid \text{goto } E \\ E &\rightarrow n \quad (n \in \mathbb{Z}) \\ &\mid E + E \\ &\mid E * E \\ &\mid - E \\ &\mid E < E \\ &\mid x \mid *x \mid \&x \\ &\mid \text{readInt} \quad (\text{정수입력}) \end{aligned}$$

점프명령어 “goto E ”에서 식 E 를 계산하면 점프할 명령어(C)의 번호가 된다. 그 명령어로 점프한다. 프로그램의 모든 명령어마다 고유의 자연수가 명령어 번호로

붙는다고 가정한다. 고유 번호 l 이 붙은 명령어 $l : C$ 는 다음과 같이 만들어진다:

$$\begin{aligned} C &\rightarrow l : C' \\ C' &\rightarrow x := E \mid *x := E \\ &\mid C ; C \\ &\mid \text{if } E C C \\ &\mid \text{repeat } C E \\ &\mid \text{goto } E \end{aligned}$$

이 프로젝트에서는 D 프로그램을 정적분석하는 “Waker”(watermark checker, 와커)라는 분석기를 디자인하고 구현하게 된다.

분석기 와커(Waker)의 배경과 목표

Natural이라는 회사는 AI 기술을 사용해서 D 프로그램 자동생성 서비스를 제공한다. 이 서비스에서 생성하는 D 프로그램에는 Natural AI가 자동 생성한 코드임을 알 수 있는 고유의 식별 코드 조각이 포함되어있다.

그 식별 코드가 실행중에 하는 일은, 어느 시점 이후에는 어느 특정 변수에 늘 특정한 성질의 정수를 저장하는 것이다. 그 성질이란 231010 (Natural사 설립일)으로 나누어 나머지가 항상 1129 (Natural사 CEO의 생일)라는 것이다.

주어진 D 프로그램이 Natural AI의 작업인지 아닌지를 확인하려면 간단치가 않다. 프로그램의 모든 입력의 경우마다 일일이 돌려보고 그런 성질의 변수가 있는지를 실행중에 출현하는지 찾아내야 하는데, 이 과정이 비용이 많이 들거나 심지어는 아예 불가능하게 된다. 입력의 경우가 무한하거나 너무 많은 경우도 그렇고, 프로그램이 사용하는 많은 변수들의 값들을 실행을 통해서 매순간 일일이 추적해야 하는 어려움도 그렇고.

좋은 방법은 정적분석 기술을 이용하는 것이다. 정적분석기술을 이용하면 주어진 소프트웨어가 어느 지점에서 위와 같은 변수를 가지고 있는지를 자동으로 안전하게 판단할 수 있게 된다. 정수 변수들이 실행중에 가지는 값들이 231010으로 나누어 어떤 나머지 값들을 가지는지를 추적하면 된다.

예를들어, 분석기 와커(Waker)는 아래의 D 프로그램을 분석해서 Natural AI가 만든 코드인지를 확인해 줄 수 있어야 한다. 변수 xp 는 프로그램의 세 번째 줄 이후부터는 항상 231010으로 나누어 나머지가 1129이기 때문이다.

```
port = 196; len = 43; pos = 139;
index = 14; fd = 15; xp := 1100; count = 0;
i = &xp; *i = xp + index + fd
repeat
    len = len + port;
    pos = pos * (*i);
    index = index * port;
    xp = *i + 1010;
    fd = fd * port;
    *i = *i + 230000;
    if (sock + 1 < pos) index = index + xp
        index = index + (-1)
(pos * 10 < port)
```

□