

# Homework 5

## COSE312, Spring 2025

Hakjoo Oh

**Due: 5/28 23:59**

**Problem 1** 언어  $S$ 를 대상으로 런타임 오류를 모두 검출하거나 없으면 없다고 검증해 주는 정적 분석기를 구현해 봅시다. 입력으로 실행 흐름 그래프(Control-flow graph)로 표현된  $S$  프로그램을 받아서 런타임 오류가 발생하지 않음을 검증하는데 성공하면 `true`, 실패하면 `false`를 반환하는 함수 `analyze`를 작성하세요.

`analyze : Cfg.t -> bool`

`analyzer.ml`에 있는 위 함수를 작성하고 `analyzer.ml` 파일 하나만 제출합니다.

### 예제

- 아래 프로그램은 오류를 하나 가지고 있고 결과로 `false`를 반환해야 합니다.

```
{
  int i;
  int[1000] a;

  while (i < 1000) {
    i++;
  }
  a[i] = 1; /* bug */
}
```

위 프로그램을 아래와 같이 수정하면 오류가 없는 안전한 프로그램이고 인터벌 분석으로 이를 증명할 수 있습니다. `true`를 반환합니다.

```
{
  int i;
  int[1000] a;

  while (i < 1000) {
```

```

        i++;
    }
    a[i-1] = 1; /* safe */
}

```

2. 아래 프로그램도 올바르지 않으므로 `false`를 반환합니다.

```

{
    int i;
    int[10] a;

    while (0 <= i) {
        i = i - 1;
    }
    print (i);
    a[i] = 0; /* bug */
}

```

3. 아래 프로그램은 입력 `x`와 `y`의 값이 예를 들어 30, 15일 때 배열 넘침이 발생할 수 있으므로 오류가 있는 프로그램입니다. 결과로 `false`를 반환해야 합니다.

```

{
    int x;
    int y;
    int z;
    int[10] a;

    read(x);
    read(y);
    z = 2 * y;

    if (z == x) {
        if (x > y + 10) {
            a[x] = 0; /* bug (e.g., x = 30, y = 15) */
        }
    }
}

```

4. 아래 프로그램은 `x`에 어떤 입력이 주어져도 오류가 발생하지 않고 이를 인터벌 분석으로 증명할 수 있는 경우입니다. `true`를 반환합니다.

```

{

```

```

int x;
int y;
int[10] a;

read(x);
while (y < 100) { y++; }
if (x < 50) {
    if (x > y) {
        a[100] = 1; /* dead code: the bug cannot occur at runtime */
    }
}
}

```

5. 아래 프로그램과 같이 끝나지 않고 무한히 도는 프로그램에 대해서도 분석 기는 유한한 시간에 답(true)을 해야 합니다.

```

{
    int i;
    int[10] a;
    while (1==1) {
        int j;
        read(j);
        if (0 <= j) {
            if (j < 10) {
                a[j] = i;
            }
        }
        i++;
    }
}

```

6. 타입 오류의 예들:

```

{
    int[10] a;
    int i;
    while (i < 10)
    {
        i = i + 1;
    }
    i[a] = 0; /* type error */
}

```

```

{
    int i;
    int[10] a;
    i = 1;
    print(a + i); /* type error */
}

```

```

{
    int i;
    int[10] a;
    i = 1;
    if (a) { /* type error */
        i = 1;
    }
}

```

7. division-by-zero 오류 예:

```

{
    int i;
    int j;
    i = 10;
    j = 5;
    while (i > 0) {
        i = i - 1;
    }
    print(i);
    print(j / i);
}

```