Homework 4
COSE312, Spring 2026

Hakjoo Oh

Due: 4/25 23:59

Problem 1 Implement a static analyzer that verifies the absence of runtime errors. It takes as input an `S` program (in control-flow graph) and returns `true` iff it succeeds to verify the program.

```
analyze : Cfg.t -> bool
```

Complete and submit `analyzer.ml`.

Examples

1. The following program contains a bug; `analyzer` should return `false`.

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

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

   Repairing the program as follows eliminates the bug; `analyzer` should return `true`.

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

  while (i < 1000) {
    i++;
```

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

2. `false` should be returned for the program:

```
{
  int i;
  int[10] a;

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

3. The bug occurs when `x` and `y` receive 30 and 15, respectively; `analyzer` should return `false`.

```
{
  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. The following program is safe for any input; the analyzer should return `true`.

```
{
  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. The following program does not terminate; however, the analyzer should return `true` in finite time.

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

6. Type-error examples:

```
{
  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 example:

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