ECE 573
Problem Set 9: Dataflow analysis

In this problem set, your goal is to develop a dataflow analysis to find `null` values. A common error in programs is dereferencing `null` values, so a dataflow analysis which can detect such dereferences can be a valuable debugging tool (imagine a compiler that flags potential null dereferences for you, before you run the program). What you should build is a dataflow analysis that determines, at each step, whether a variable is *definitely not null*, *definitely null* or *may be null*.

A hint to keep in mind as you develop this analysis: think about how this analysis relates to the constant propagation analysis).

1. What is the lattice that you should use for this analysis?

2. Which direction should this analysis use?

3. What is the confluence operator?

4. Give the transfer functions for the following statements (for each transfer function, specify which variables will *change* their states, and under which conditions).

   - `x := null`
   - `if (x == null) then { ... } else { ... }`
     (for this statement, show what gets propagated along each branch)
   - `x := y`
   - `x := &z`

5. Argue that the transfer functions you developed in step (4) are monotonic.

6. How should this analysis be initialized?

7. Show the results of your dataflow analysis for the following piece of code:

```
    : x := 4;
    : y := null;
L1  : if (y == null) goto L2;
    :    x = y;
    :    if (x == null) goto L3;
    :       x = 5;
    :       goto L4;
L3  :       x = null;
```

```
L4 :    y = x;
   :    goto L1;
L2 : end;
```