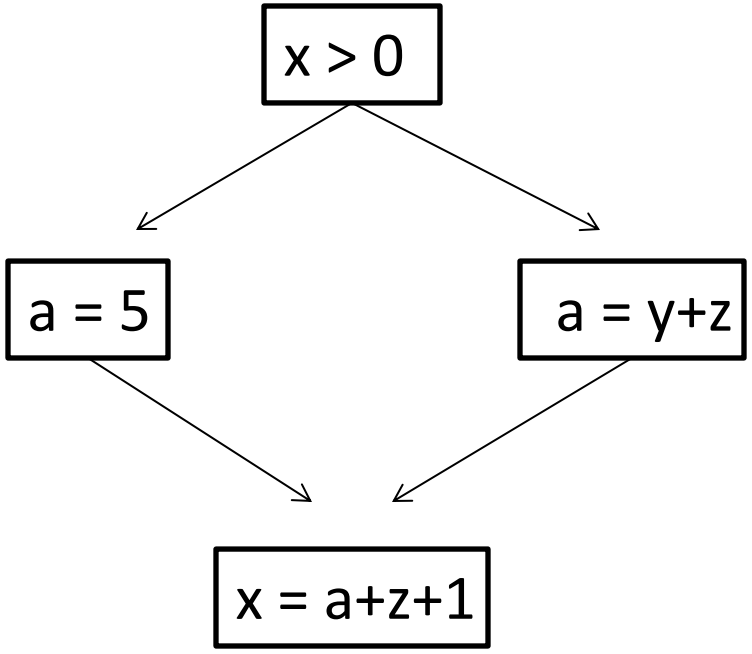
Dataflow Analysis

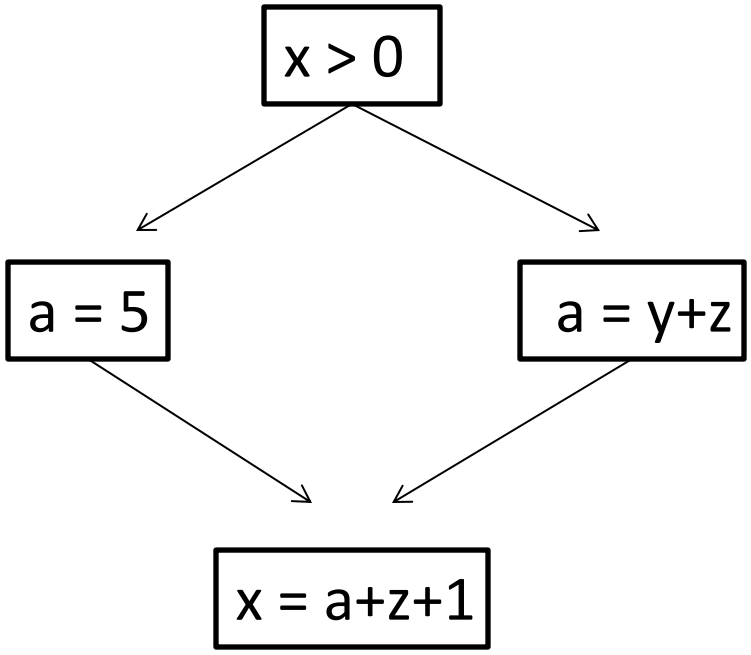
Dataflow analysis is a collection of techniques for analyzing the runtime behavior of a program. Many optimization techniques are based on dataflow analysis. For example, one kind of dataflow analysis called *live variable analysis* determines which variables at a given point have data in them that will be used again. We can use this to determine which variables should be saved in registers, and we can also use it to find un-initialized variables. Most compilers do some sort of dataflow analysis before generating code.

For example, consider the following block of code:

```
if (x > 0)
        a = 5
else
        a = y+z
x = (a+z)+1
```



The flowgraph for this block is shown at the right:



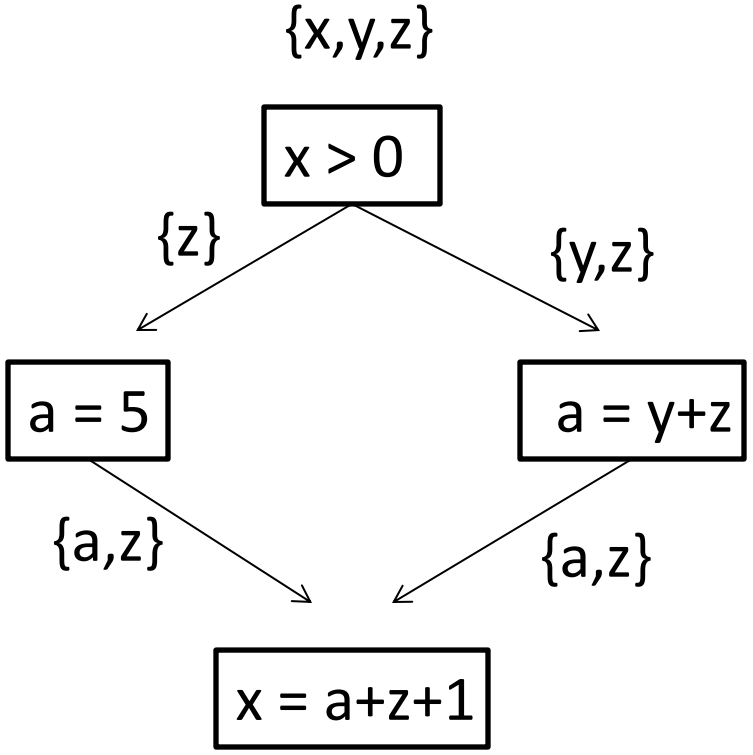
Dataflow analysis constructs the flow graph for selected portions of a program and then tries to draw conclusions about runtime behavior from the flow graph.

As an illustration of the dataflow techniques we will do some *live variable analysis*.

Terminology:

- The statement x=y+z is said to *define* x and to *use* y and z.
- The variable x is *live* at point p in the program if:
  A. There is a backwards path starting at p that goes to a definition of x
  B. There is a forward path starting at p that reaches a use of x without passing through another definition of x.

For the example assume that {x,y,z} are all live coming into the block. Here are the live variables at every point:



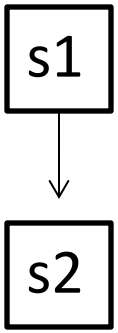
To compute live variables we will use the following sets:

In(s)={ live variables before statement s is executed}
Out(s) = {live variables after statement s is executed}
Use(s)={variables used in statement s}
Def(s)={variables defined in statement s}

In the following equations + represents set union and - represents set difference.

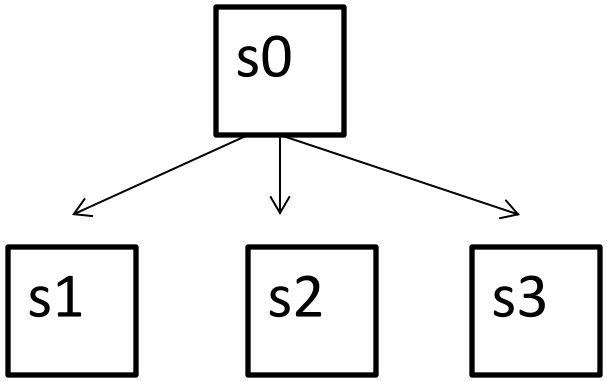
Here are equations relating these sets:

A. In(s)=Use(s) + {Out(s) - Def(s)}
B. If s2 immediately follows s1:



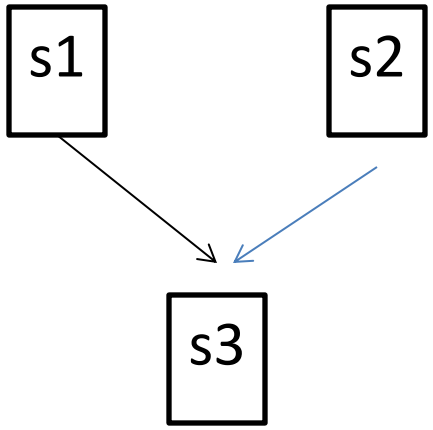
then In(s2) = Out(s1)

C. The Fan-out rule:



Out(s0) = In(s1)+In(s2)+In(s3)

D. The Fan-in rule:



Out(s1) = In(s3)

Out(s2) = In(s3)

E. Nothing is live after the final statement