CMSC 430
Introduction to Compilers
Fall 2016

Static Single Assignment Form
Motivation

• Data flow analysis needs to represent facts at every program point

• What if
  - There are a lot of facts and
  - There are a lot of program points?
  - $\Rightarrow$ potentially takes a lot of space/time

• Most likely, we’re keeping track of irrelevant facts
Example

\[
x := 3
\]
\[
y := a + b
\]
\[
z := 2 \times y
\]
\[
w := y + z
\]
\[
a > b
\]
\[
y := a - b
\]
\[
y := y \times 10
\]
\[
w := w + y
\]
\[
z := w + x
\]
Sparse Representation

• Instead, we’d like to use a sparse representation
  ▪ Only propagate facts about \( x \) where they’re needed

• Enter *static single assignment* form
  ▪ Each variable is defined (assigned to) exactly once
  ▪ But may be used multiple times
Example: SSA

- Add SSA edges from definitions to uses
  - No intervening statements use/define variable
  - Safe to propagate only along SSA edges
What About Joins?

- Add $\Phi$ functions/nodes to model joins
  - Intuitively, takes meet of arguments
  - At code generation time, need to eliminate $\Phi$ nodes