**Static Single Assignment Form**

**Last Time**
- Introduction to SSA
- Inserting $\phi$ functions

**Today**
- Csmith paper
- Renaming variables for SSA form
- SSA and DFA

**Next Time**
- Reuse optimizations

---

**LLVM Questions**

**Are you confused by LLVM?**
Transformation to SSA Form

Two steps
- Insert $\phi$-functions
- Rename variables

Variable Renaming

Basic idea
- When we see a variable on the LHS, create a new name for it
- When we see a variable on the RHS, use appropriate subscript

Easy for straightline code

Harder when there's control flow
- For each use of $x$, find the definition of $x$ that dominates it

Will such a definition exist?
How can we find it?
The dominance tree shows the dominance relation

- At any point in the dominance tree, look up the tree, searching for the first definition that you find
- Make a depth-first traversal of the Dominance Tree to assign version numbers

Variable Renaming (cont)

**Data Structures**
- Stacks[v] \( \forall v \)
  - Holds the subscript of most recent definition of variable \( v \), initially empty
- Counters[v] \( \forall v \)
  - Holds the current number of assignments to variable \( v \); initially 0

**Auxiliary Routine**

procedure GenName(variable v)
  i := Counters[v]
  push i onto Stacks[v]
  Counters[v] := i + 1

Use the Dominance Tree to remember the most recent definition of each variable
Variable Renaming Algorithm

```plaintext
procedure Rename(block b)
    if b previously visited return
    for each \( \phi \)-function \( p \) in b
        GenName(LHS(p)) and replace \( v \) with \( v_i \), where \( i = \text{Top}(\text{Stack}([v])) \)
    for each statement \( s \) in b (in order)
        for each variable \( v \in \text{RHS}(s) \)
            replace \( v \) by \( v_i \), where \( i = \text{Top}(\text{Stacks}([v])) \)
        for each variable \( v \in \text{LHS}(s) \)
            GenName(\( v \)) and replace \( v \) with \( v_i \), where \( i = \text{Top}(\text{Stack}([v])) \)
    for each \( s \in \text{succ}(b) \) (in CFG)
        \( j \leftarrow \) position in \( s \)'s \( \phi \)-function corresponding to block \( b \)
        for each \( \phi \)-function \( p \) in \( s \)
            replace the \( j \)th operand of RHS(\( p \)) by \( v_i \), where \( i = \text{Top}(\text{Stack}([v])) \)
    for each \( s \in \text{child}(b) \) (in DT)
        Rename(s)
        for each \( \phi \)-function or statement \( t \) in \( b \)
            for each \( v_i \in \text{LHS}(t) \)
                Pop(\( \text{Stack}([v]) \))
        Unwind stack when done with this node
    Recurse using Depth First Search
```

Call Rename(entry-node)