**Source and Target CFG Vertices**

Translation of an AST construct in general takes:

* initial vertex
* target vertex (or two target vertices for conditional statements)

**Sequence of Statements**

Sequence of statements:

s1;s2

v0 ---------> v1

becomes

s1 s2

v0 -----> v -----> v1

That is:

translateStmt(v0, s1; s2, v1) =

val v = newVertex

translateStmt(v0,s1,v)

translateStmt(v,s2,v1)

For simple statements:

translate(v0, Assign(L,R), v1) =

cfg += (v0, Assign(L,R), v1)

**Conditional Expressions**

translateCond(v0, p1 && p2, vFalse, vTrue) =

vp1True = newVertex

translateCond(v0, p1, vFalse, vp1True)

translateCond(vp1True, p2, vFalse, vTrue)

**Using 'Current Vertex'**

Use a global variable for the originating vertex

var pc : Vertex

emit(simpleStmt) =

val v = newVertex

cfg.edges += (pc, simpleStmt, v)

pc = v

translateStmt(s1; s2; ... sn, vAfter) =

emit(s1)

emit(s2)

...

emit(sn)

pc = vAfter