

# Lec 11

## SSA

### # SSA

- Every var / temp assigned at most once
- Like functional prog. & math
- Makes optimisation easier

```

ex i ← 0
   ⋮
   ] if SSA, i not
   ] changed here
   if (i > 0) then cont
   else done → goto done
    
```

### # Transforming basic block

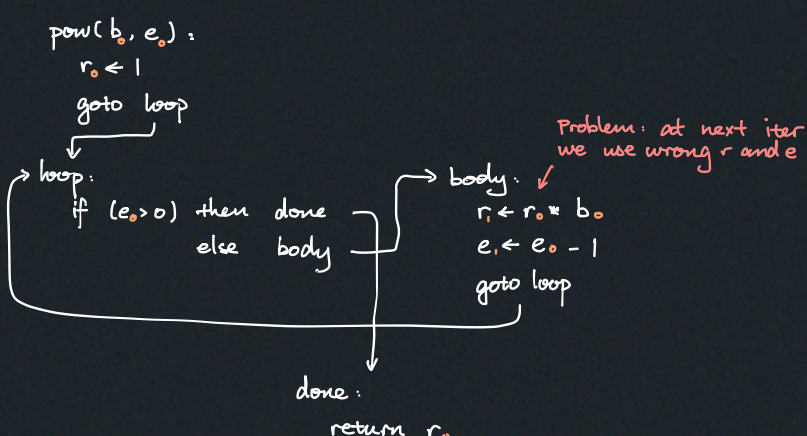
Recall basic block : - starts with label  
 - ends with jump  
 - no jump / label in middle

Use generation counter.

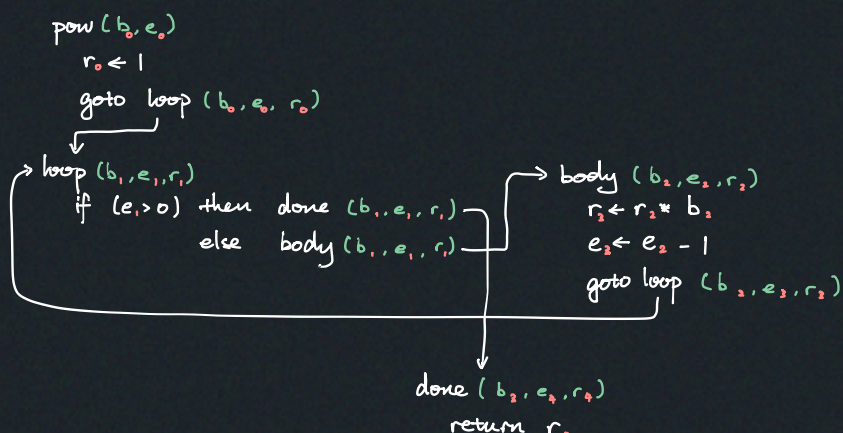
```

Consider dist(x, y):
  x1 ← x0 * x0
  y1 ← y0 * y0
  t0 ← x1 + y1
  t1 ← isqrt(t0)
  return t1
    
```

← Works fine



Idea: use parameters in each block



Minimal SSA: each var / temp only used in one basic block  
 → Optimisation can operate on individual basic block

### # SSA & FP

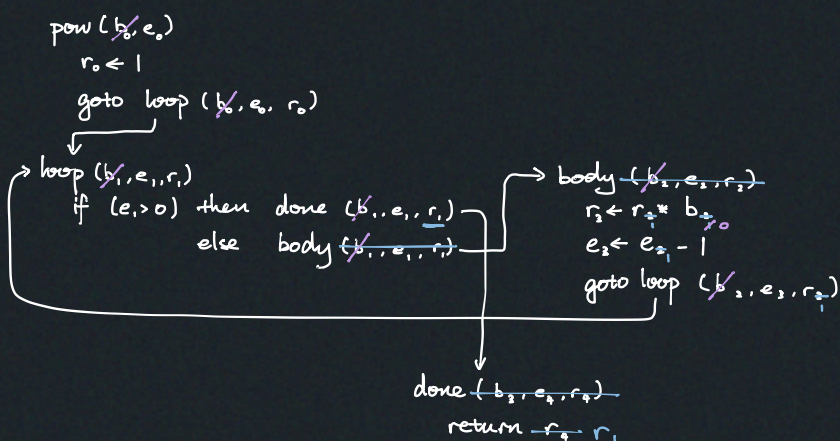
- SSA progs can be viewed as functional prog
  - ↳ only tail calls
  - ↳ no HOFs
  - ↳ no shadowing

```

Above: let pow(b0, e0) = let r0 = 1 in loop(r0, b0, e0)
       let loop(b1, e1, r1) = if e1 > 0 then done(b1, e1, r1)
                               else body(b1, e1, r1)
       let body(b2, e2, r2) = let r2 = r1 * b1 in
                               let e2 = e1 - 1 in
                               loop(r2, e2, b2)
       let done(b3, e3, r3) = r3
    
```

### # Steps

1. Convert to minimal SSA
2. Determine & remove unnecessary parameters
  - ↳ Blocks called only at one place
  - ↳ Params that are not live
  - ↳ Params whose number's not changed within block, just passed through from arg to goto, viz. not updated



In general,  $x_i$  in label  $l(\dots, x_i, \dots)$  can be removed if  $\exists k$  st. all jumps to  $l$  have form  $goto\ l(\dots, x_k, \dots)$  or  $goto\ l(\dots, x_i, \dots)$ . Then replace  $x_i$  with  $x_k$

3. Code gen — move things into args before jumping

```

pow:
  r0 ← 1
  e1 ← e0
  r1 ← r0
  goto loop
    
```

### # $\phi$ function

- organise call / jumps differently
- for each var, list all possible temps

```

loop block:
  e1 = { e0 if from pow
        e2 if from body
    
```

```

→ loop:
  e1 =  $\phi(e0, e2)$ 
  r1 =  $\phi(r0, r2)$  ← useful for liveness & SSA minimisation
  if (e1 > 0) then done else body
    
```