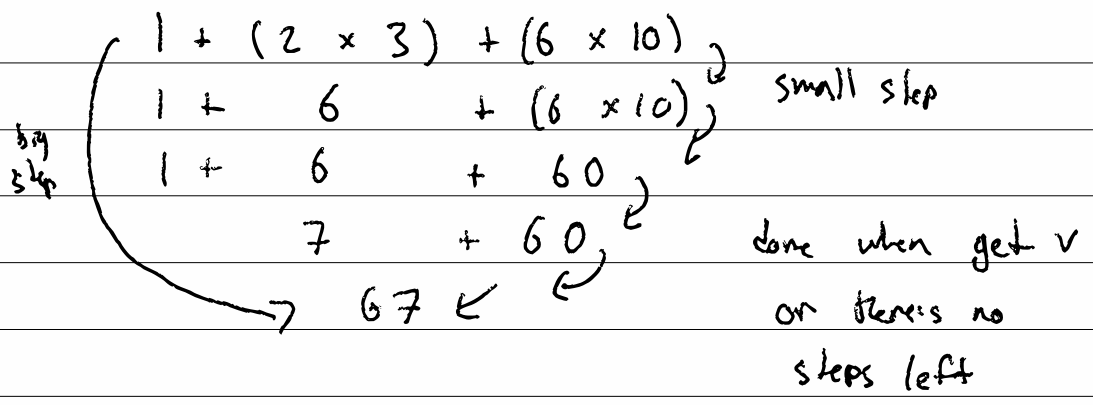


$2-1/$ J_1 $e := v \mid (e \ e \ \dots) \mid (\text{if } e \ e \ e)$
 $v := b$
 $b := \text{num} \mid \text{bool} \mid \text{prim}$
 $\text{prim} := +, -, *, /, \leq, <, =, >, \geq, \dots$

big step semantics $\text{interp} : e \rightarrow v$
 $\text{interp} (1 \ 1 \ 0) = \perp$ $\delta(1, 1, 0) = \perp$
 math isn't like this!

equational theory $a + b = b + a$
 $1 + 1 = 2$
 $2 + 3 = 2 + 2 + 1 = 4 + 1 = 5$

2-2 / small-step evaluation



```
interp e =  
  let e' = step e  
      if e == e' then  
        return e  
      o.w., interp e'
```

2-3/ step : $e \rightarrow e$

$\left. \begin{array}{l} \text{step (if } v \text{ et } ef) = ef \text{ where } v \neq \text{false} \\ \text{step (if false et } ef) = ef \\ \text{step (p } v \dots) = \delta(p, v \dots) \\ \text{step } v = v \end{array} \right\} \begin{array}{l} \text{step} \\ \text{if} \\ \text{atom} \\ \text{rules} \end{array}$

$\left[\begin{array}{l} \text{step (if } e \&v \text{ et } ef) \\ = \text{(if (step } e) \text{ et } ef) \\ \text{step (v } \dots \text{ e} \&v \text{ earg } \dots) = \\ \text{(v } \dots \text{ (step } e) \text{ earg } \dots) \end{array} \right] \begin{array}{l} \text{congruence} \\ \text{rules} \\ \text{or structural} \\ \text{rules} \end{array}$

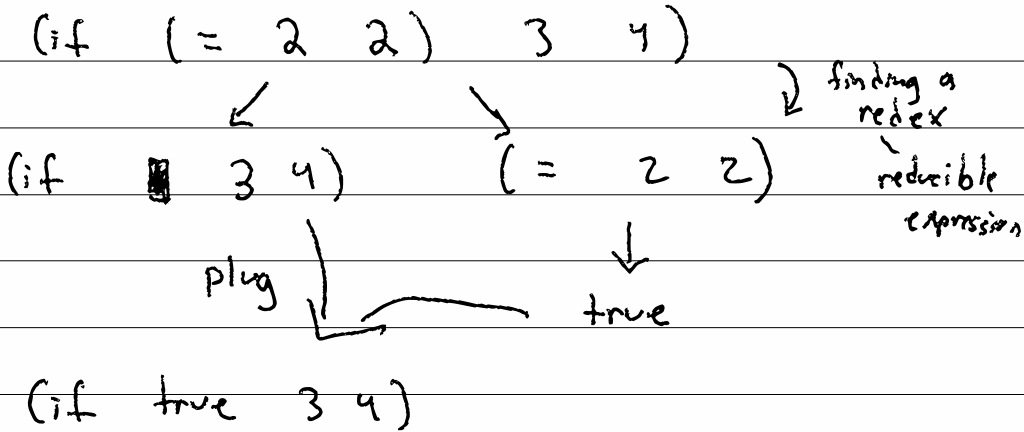
2-4/ A context is a "program with a hole".

$C ::= \text{hole}$ — ~~✗~~
 $\quad | \text{if } C \ e \ e$ if1C
 $\quad | \text{if } e \ C \ e$ if2C
 $\quad | \text{if } e \ e \ C$ if3C
 $\quad | (e \dots C \ e \dots)$ appC

(+ 1 (if (+ 2 ~~✗~~) 3 4))

$\rightarrow \text{appC } [+ , []]$
 $\quad \text{if1C } (\text{appC } [+ 2] \text{ hole } [])$
 $\quad \quad 3 \ 4)$
 $\quad \quad [])$

2-5/



$$C[e] = \text{plug}(C, e)$$

$$\text{plug} : C \times e \Rightarrow e$$

$$\text{plug } \square e = e \quad \text{plug } (\text{if } C \ e_1 \ e_2) \ e = (\text{if } C[e] \ e_1 \ e_2)$$

$$\text{plug } (\text{appC } (b \dots) \ C \ (a \dots)) \ e = (b \dots C[e] \ a \dots)$$

2-b/ step $v = v$

step $c[(\text{if } \text{false } c_+ \text{ } e_+)] = \{e_+\}$

step $c[(\text{if } \text{true } e_+ \text{ } e_+)] = \{e_+\}$

step $c[(P \ v \ \dots)] = \{ \delta(P, v \ \dots) \}$

~~step $e = e'$, then step $c[e] = c[e']$~~

2-7 When are two programs "the same"?

$$\begin{aligned} 1+1 &= 1+1 \\ &= 1+(2-1) \end{aligned}$$

two programs are the same as
 $\forall c. \text{eval } c[x] = \text{eval } c[y]$

~~object~~ observational equivalence