1-3 why semantics?

- prove that two programs do the same thing
- one program matches math

context [your program / fun]

B, com [x + 2] = B, com [x << 1]

$\forall b, b' \forall B_L, B_r.$ $(B_L \bullet b)\ r_b\ (B_L \bullet b')$ [context, left]

if $b\ r_b\ b'$

$(b \bullet B_r)\ r_b\ (b' \bullet B_r)$ [context, right]

if $b\ r_b\ b'$

context-compatible closure of a relation

$\rightarrow_{\_}$ context-compat of the single-step

$\twoheadrightarrow_{\_}$ refl + transitive closure of $\rightarrow_{\_}$

$=_{\_}$ symmetry of $\twoheadrightarrow_{\_}$

eval$_{_}$ (x : stx) = { y s.t. $x \twoheadrightarrow y$ and y is a result

a result is a subset of stx

B = T | F | (B • B)

R = T | F

2-1 interpreter : programs in A that tell you what B programs do

small-step : simulate B in math

$$p \to p' \to \dots \to \text{ans}$$

big-step : defines in math the meaning

$$p \Downarrow \text{ans} \quad (p \twoheadrightarrow \text{ans})$$

denotational : compiler (in math) from B to math

$$c(p) = \text{math-thing}$$

---

1) employ professors 2) answers qs for programmers

q_1 : what can programs do? (what's possible)

q_2 : did I / we do the right thing?

- p eval(p) = ans is ans right?
- will ~~any~~ program give an ans? always?
- the same ans?

Deterministic : $p \twoheadrightarrow v \Rightarrow$ there's exactly one path from p to v

Function : $[\text{evalr}(B_0) = R_1 \wedge \text{evalr}(B_0) = R_2 \Rightarrow R_1 = R_2]$

$$\text{evalr}(B) = \begin{cases} F & \text{if } B =_r F \\ T & \text{if } B =_r T \end{cases}$$

$$\forall B_0, R_1, R_2 \; [B_0 =_r R_1 \wedge B_0 =_r R_2 \Rightarrow R_1 = R_2]$$

$M = (1+1)+(2+2)$

$L = 2+(2+2)$ $(1+1)+4$

$N = 2+4$

$\Downarrow$

6

$M =_r N \models$
1. $M = N$ (refl)
2. $N =_r M$ (sym)
3. $M \to N$ (s.s.)
4. $(L,\ M =_r L,\ L =_r N)$ (trans)

2-2

$B_0 \twoheadrightarrow R_1$ $\qquad$ $B_0 \twoheadrightarrow R_2$

$R_i \in \{F, T\}$ $\qquad$ $R_1 \twoheadrightarrow x$, $x = R_1$ (only refl)

~~Church-Rosser Property:~~ $\boxed{\text{Consistency:}}$

$\forall M, N,$ If $M =_r N$, exists L, $M \twoheadrightarrow_r L$ and $N \twoheadrightarrow_r L$.

proof.

Induction on $M =_r N$.

(sCs) Case $M \twoheadrightarrow_r N$. $L = N$.

(sym) case $M =_r N$ because $N =_r M$.

IH: (exists L. $N \twoheadrightarrow_r L$ and $M \twoheadrightarrow_r L$)

exact IH.

(refl) case $M =_r N$ because $M = N$. $L = N$.

(trans) case $M =_r N$ because $M =_r L_0$ and $L_0 =_r N$.

IH$_1$: ($\exists L_1$. $M \twoheadrightarrow_r L_1$ and $L_0 \twoheadrightarrow_r L_1$)

IH$_2$: ($\exists L_2$. $L_0 \twoheadrightarrow_r L_2$ and $N \twoheadrightarrow_r L_2$)

$$
\begin{tikzcd}
M \arrow[r, phantom, "=_r"] \arrow[dr, two heads] & L_0 \arrow[r, phantom, "=_r"] \arrow[d, two heads] \arrow[dr, two heads] & N \arrow[d, two heads] \\
& L_1 & L_2
\end{tikzcd}
$$

apply Diamond Property

$\Rightarrow$

$$
\begin{tikzcd}
L_1 \arrow[dr, two heads] & & L_2 \arrow[dl, two heads] \\
& L_3 &
\end{tikzcd}
$$

ans is L_3!

2-3 Diamond property called Church-Rosser Property

If $L \twoheadrightarrow_r M$ and $L \twoheadrightarrow_r N$, then $\exists L'$.
$M \twoheadrightarrow_r L'$ and $N \twoheadrightarrow_r L'$.

$$\begin{tikzcd} & L \arrow[dl] \arrow[dr] & \\ M & & N \end{tikzcd} \Rightarrow \begin{tikzcd} & L \arrow[dl] \arrow[dr] & \\ M \arrow[dr] & & N \arrow[dl] \\ & L' & \end{tikzcd}$$

$\cancel{X} \twoheadrightarrow_r Y$ = $X \; r \; Y$ (base)

$X = Y$ (refl)

$\exists Z.$ $X \rightarrow_r Z$ and $Z \rightarrow_r Y$ (trans)

$$\frac{b \rightarrow_r b'}{(B \bullet b) \rightarrow_r (B \bullet b')}$$ ← compat, left

compat, right

case refl. $L' = M$

case base.

$L \; r \; M$ and $L \; r \; N$

→ contradiction!

case trans.

case compat, left:

$L = (B_m \bullet b_m)$ $b_m \rightarrow_r b_m'$ $L = (B_n \bullet b_n)$ $b_n \rightarrow_r b_n'$

$M = (B_m \bullet b_m')$ $N = (B_n \bullet b_n')$

$B_m = B_n$ $b_m = b_n$

$b \rightarrow_r b'_m \wedge b \rightarrow_r b'_n$

IH: $\exists b_1$, $b'_m \rightarrow_r b_1 \wedge b'_n \rightarrow_r b_1$

case compat, right:

similar

$M =_r R$ iff $M \twoheadrightarrow_r R$

↓

$\twoheadrightarrow_r$ + sym

sym doesn't matter

$\forall B_0 \in B. \exists R_0 \in R. \text{eval}(B_0) = R_0.$ $\neg \forall R_0. \text{eval}(B_0) \neq R_0$

interp: $B \Rightarrow R$

proof → normal math

→ constructive

↳ give a witness