21-1

$B ::= \ldots \mid \text{Even} \mid \text{Odd}$ (not. [illegible])

$\Delta(0) = E \qquad \Delta(1) = \text{Odd} \qquad \Delta(2) = \text{Even}$

$\Delta(+)(E)(E) = E \qquad E \times O = O \qquad O \times E = O \qquad O \times O = E$

$\Delta(/2)(E) = E \qquad \Delta(/2)(O) = \bot \qquad 5/2$ or $(+1\,4)/2$ are stuck

---

$\models P$ means $P \in$ set of progs that run forever

$\not\models b \qquad \not\models X \qquad \not\models \lambda X.M$

$\dfrac{\models M}{\models M\ N} \qquad \dfrac{\models N}{\models M\ N} \qquad \models \Omega$

When is a type system correct?

$\llcorner$ accurate predictions (connection w/ typesys & semantics)

$\Gamma \vdash M : T \qquad \text{eval}(M) = \text{ans} \left(\begin{smallmatrix} b \\ \text{ifun} \end{smallmatrix}\right)$

$M \Rightarrow^* N$ (maybe N is V)

$\forall M \in \text{progs}.\ \emptyset \vdash M : T \rightarrow \text{eval}(M) = b \wedge \emptyset \vdash b : T$

$\rightarrow M \Rightarrow^* V$ and $\emptyset \vdash V : T$

20-3 was "simply-type λ-calculus"

has "strong normalization" = All programs have values

all decidable. (Σ_0)

[illegible] Adding recursion

$$M ::= \dots \mid (\text{fix } M)$$
$$E ::= \dots \mid \text{fix } E$$

$$E[\text{fix } (\lambda x.M)] \to E[M[x \leftarrow \text{fix } (\lambda x.M)]]$$

$$\frac{\Gamma \vdash M : (T_1 \to T_2) \to (T_1 \to T_2)}{\Gamma \vdash \text{fix } M : (T_1 \to T_2)}$$

old: $\emptyset \vdash M : T \Rightarrow M \Rightarrow^* V$ and $\emptyset \vdash V : T$

new: or it runs forever

progress: "Programs w/ types are not stuck"

$$\Gamma \vdash M : T \Rightarrow \exists N.\ M \Rightarrow N$$

preservation: "Type is preserved through eval"

$$\Gamma \vdash M : T \wedge M \Rightarrow N \to \Gamma \vdash N : T$$

$$\frac{\dfrac{\emptyset[x \to \text{Num}] \vdash x : \text{num}}{\emptyset \vdash (\lambda x:\text{num}.x) : (\text{num} \to \text{num}) \to (\text{num} \to \text{num})}}{\emptyset \vdash \text{fix } (\lambda x:\text{Num}.\ x) : (n \to n) \to (n \to n)}$$

$$\text{fix } (\lambda x:N.x) \Rightarrow x[x \leftarrow \text{fix } (\lambda x:n.x)]$$

```
(fix (λ fac:(num -> num)
        λn : num.
         if   n == 0
          then  1
        o.w.   n × fac (n-1) )) 5    =  5 × 4 × 3 × 2 × 1
```

21-3

Normal ISWIM (w/o a t.s.) had recursion but S.T. ISWIM didn't

fix M → Y M

λ-calculus term m

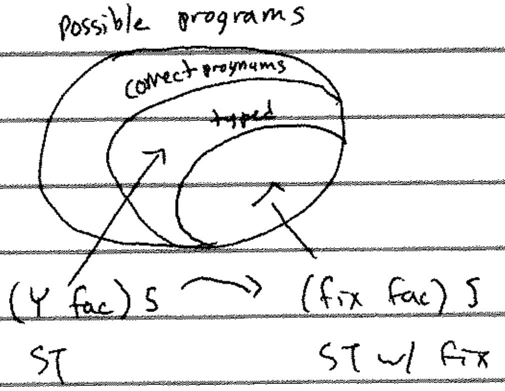
$\emptyset \not\vdash Y : T$

Y has no-type in S.T. ISWIM

we know! Y fac doesn't get stuck!

"If M passes the typesystem, then it will not get stuck."

"If M gets stuck, it will not pass type system"



Maybe T = None
 | Some T

<optional> (C++, boost)

Conditionals

M ::= .... | true | false | if M M M

V ::= .... | true | false

E ::= .... | if E M M

"PHP Wat?"

E[if true M N] → E[M]

E[if false M N] → E[N]

T ::= .... | Bool

$\Gamma \vdash$ false : Bool

$\Gamma \vdash$ true : Bool

	union	inter
$\Gamma \vdash F : R$	S	S
$\Gamma \vdash T : R$	R	R
$\Gamma \vdash C : Bool$		
$\Gamma \vdash$ if C T F : R	R∪S	R∩S

T ::= .... | T∪T
 .... | T∩T

$\Gamma \vdash M : T; P_T; P_F$

"Occurrence typing"

$$\frac{\Gamma \vdash C : Bool; P_T; P_F \qquad \Gamma \cap P_T \vdash T : R; P_{TT}; P_{TF} \qquad \Gamma \cap P_F \vdash F : S; P_{FT}; P_{FF}}{\Gamma \vdash \text{if}\ C\ T\ F : R \cup S;\ P_{TT} \wedge P_{FT};\ P_{FF} \wedge P_{FF}}$$

(if T 5 F) → 5

(if F 5 F) → F

(if X 5 false) → 5

→ false