23-1/ $e = v \mid x \mid (e \quad e)$

$v = \lambda x.\, e \mid p \mid b$

$p = +, -, *, \ldots$ $\qquad \Gamma = \emptyset$

$b = \text{bool}, \text{num}, \text{etc}$ $\qquad \mid \Gamma[x \mapsto T]$

$T = B \mid T \to T$

$B = \text{base types}, \ldots$

$\Delta : b/p \to T$

$\Delta(+) = \text{Num} \to \text{Num} \to \text{Num}$

$\Delta(\text{true}) = \text{Bool} \qquad \Delta(5) = \text{Num}$

↙ typing judgements

Before: $\vdash e : T$ "proves that e has type T"

Now: $\Gamma \vdash e : T$ $\qquad \emptyset[x \mapsto \text{Num}][y \mapsto \text{Bool}] \vdash x : \text{Num}$

23-2/

$$\frac{\Delta(b) = T}{\Gamma \vdash b : T}\,A \qquad \frac{\Delta(p) = T}{\Gamma \vdash p : T}\,B \qquad \frac{\Gamma(x) = T}{\Gamma \vdash x : T}\,C$$

$$\frac{\Gamma \vdash e_1 : T_{Dom} \rightarrow T_{rng} \qquad \Gamma \vdash e_2 : T_{dom}}{\Gamma \vdash (e_1 \;\; e_2) : T_{rng}}\,D$$

$$\frac{\Gamma[x \mapsto T_{Dom}] \vdash e : T_{rng}}{\Gamma \vdash \lambda x, e : T_{Dom} \rightarrow T_{rng}}\,E$$

23-3/ $\emptyset \vdash ((\lambda x.\ ((+\ 1)\ x))\ 2) : \mathrm{Num}$

D

$\emptyset \vdash \lambda x.\ ((+\ 1)\ x) : \mathrm{Num} \to \mathrm{Num} \qquad \emptyset \vdash 2 : \mathrm{Num}$ A

$\emptyset[x \mapsto \mathrm{Num}] \vdash ((+\ 1)\ x) : \mathrm{Num} \qquad \Delta(2) = \mathrm{Num}$

$x : \mathrm{Num} \vdash (+\ 1) : \mathrm{Num} \to (N \to N) \quad x : \mathrm{Num} \vdash x : \mathrm{Num}$ C

$x : N \vdash + : \mathrm{Num} \to (N \to N) \quad x : N \vdash 1 : \mathrm{Num} \quad x : \mathrm{Num}\ (x) = \mathrm{Num}$

A

$\Delta(+) = N \to N \to N \qquad \Delta(1) = \mathrm{Num}$

23-4 typeof : Gamma → Expr → Maybe Type

```
typeof Γ const(b) = Δ(b)
typeof Γ prim(p) = Δ(p)
typeof Γ var(x) = Γ(x)
typeof Γ app(e1, e2) = do
    Fun(tdom, trng) ← typeof Γ e1
    tx ← typeof Γ e2
    if tdom == tx then return trng
    else nothing
typeof Γ lam(x, e) = do                unbound
     trng ← typeof Π[x ↦ tdom] e
     return Fun(tdom, trng)
```

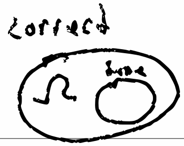
23-5/

$$v = \lambda \cancel{x}. e$$

$$v = \lambda x \underbrace{: T}_{\text{type tax}} . \; e$$

$$\frac{\Gamma[x \mapsto T_d] \vdash e : T_r}{\Gamma \vdash \lambda x : T_d . \; e \;\vdash T_d \Rightarrow T_r}$$

```
typeof  Γ  lam(x, tdom, e) = do
    trng ← typeof  Γ[x ↦ tdom]  e
    return  Fun(tdom, trng)
```



23-6

untyped : $(\lambda x.\ x\ x)\ \underbrace{(\lambda x.\ x\ x)}$ $\qquad \Omega$

$= \qquad\qquad\qquad\qquad\qquad\qquad \to \Omega$

$(\lambda x.\ x\ x)\ (\lambda x.\ x\ x)$

typed :

$$\emptyset \vdash (\lambda x{:}T.\ x\ x)\ (\lambda x{:}P.\ x\ x) :$$

$$\emptyset \vdash (\lambda x{:}T.\ x\ x) : \underbrace{T} \to \underbrace{Q} \qquad \emptyset \vdash (\lambda x{:}P.\ x\ x)\ P \to \underbrace{\ }$$

$$\emptyset[x \mapsto T] \vdash (x\ \ x) : Q$$

$$\emptyset[x \mapsto T] \vdash x : S \to Q \qquad \emptyset[x \mapsto T] \vdash x : S$$

$$T = S \to Q \qquad\qquad T = S$$

is there an S s.t. $S = S \to Q$?

$$S_1 = S_1 \to S_2 \qquad S_2 = Q \qquad (S_1 \to S_2) = (S_1 \to S_2) \to Q$$

23-7/ $v = \lambda\ \cancel{x:} T . e$ $\qquad \lambda\ \underbrace{T_r}\ f\ (x : \underbrace{T_d}) . e$

~~$\lambda\ f\ (x:T) . e$~~

type tax

$$\frac{\Gamma[x \mapsto T_d][f \mapsto T_d \to T_r] \vdash e : T_r}{\Gamma \vdash \lambda\ T_r\ f(x:T_d) : T_d \to T_r}$$

23-8/ $\mathrm{true} := \lambda x.\ \lambda y.\ x$

$\mathrm{false} := \lambda x.\ \lambda y.\ y$

$\mathrm{if} := \lambda c.\ \lambda x.\ \lambda y.\ ((c\ x)\ y)$

$e = \ldots \mid \mathrm{if}\ e\ e\ e$

$$\frac{\Gamma \vdash e_c : \mathrm{Bool} \qquad \Gamma \vdash e_t : \cancel{T_1}\ T \qquad \Gamma \vdash e_f : \cancel{T_2}\ T}{\Gamma \vdash \mathrm{if}\ e_c\ e_t\ e_f : \cancel{T_1}\ \ \cancel{T_2}\ \ T_1 \cap T_2}$$

T $\quad \boxed{T_1 \cup T_2}$ ← Typed Racket

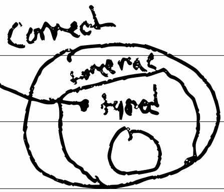
↳ occurrence typing

```
(+ 1 (if false 2 "two"))
(+ 1 (if true "two" 2))
```

```
23-9) (let  x = if  (< (read) 5)  then  2
                                  else  "two"

    if  (string? x)  then
      str length  x
    else   x • 4  )  :  Num
```

Typed Racket allows ——→ Py, JS, Ruby, PHP


correct
general
typed