19

$$\frac{\overbrace{\Gamma[x \mapsto D]}^{\text{in}} \vdash \overset{\text{in}}{\widehat{e}} : \overset{\text{out}}{\widehat{R}}}{\underbrace{\Gamma}_{\text{env}} \vdash \underbrace{(\lambda x.\ e)}_{\text{expr}} : \underbrace{D \Rightarrow R}_{\text{output}}}$$

input

guess D

```
f(int x) { ... }
      D

f(x)     { ... }
```

↗ guess what x is

try 'em all
→ slow

$$\frac{\Gamma \vdash f : D \Rightarrow R \qquad \Gamma \vdash a : D}{\Gamma \vdash (f\ a) : R}$$

$T ::= \ldots \mid T \Rightarrow T \mid \text{list}\ T$

Coq: isort (l : list nat) → (a : list Nat | same_elements l a ∧ ordered a)

~~return~~ ret (insert (fst, isort (rest)) + PROOF)

```
f( _ , x) {
  return x + 7 == x / 2;
}
```

how x is used tells you possible types

constraint generation ⟹ constraint solving

(20) $x + 3y + 1z = 9$
$x + y - z = 1$
$3x + 11y + 5z = 35$

$x = 9 - 3y - 1z$
~~$y = 1 - z - (9 - 3y$~~

P1

$$\begin{bmatrix} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 11 & 5 & 35 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 2 & 2 & 8 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & -2 & -3 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Linear Constraints = "LHS = RHS"
LHS=RHS = " constant * var "
| " LHS + RHS "

Ty = Ty
Ty = var | int | str
| Ty ⇒ Ty

e := num
| e + e
| λx. e
| e e
| x
| (if e e e)

C.G. (var, e) = set of constraints where var is the type of e

$CG(\ell, num) = \{ \ell = Int \}$

$CG(\ell, e_1 + e_2) = \{ \hat{e}_1 = Int, \hat{e}_2 = Int, \ell = Int \} \cup CG(\hat{e}_1, e_1)\ CG(\hat{e}_2, e_2)$

$CG(\ell, \lambda x. e) = \{ \ell = \hat{x} \rightarrow \hat{e} \} \cup CG(\hat{e}, e)$

$CG(\ell, x) = \{ \hat{x} = \ell \}$

$CG(\ell, (e_1\ e_2)) = \{ \hat{e}_1 = \hat{e}_2 \rightarrow \ell \} \cup CG(\hat{e}_1, e_1) \cup CG(\hat{e}_2, e_2)$

$CG(\ell, (\text{if } e_1\ e_2\ e_3)) = \{ \hat{e}_1 = Bool, \ell = \hat{e}_2, \ell = \hat{e}_3 \}$
$CG(\hat{e}_1, e_1) \cup CG(\hat{e}_2, e_2) \cup CG(\hat{e}_3, e_3)$

(21)

CG(top, (λx. x+3))

$= \{\ top = \hat{x} \Rightarrow body,$ — λ

$body = Int,$ — +

$lhs = Int,$

$rhs = Int;$

$\hat{x} = lhs,$ — var

$rhs = Int\ \}$ — num

$\{\}$ ↓ $solved_0$

$unsolved_0$

Solve : {unsolved} x {solved}

→ {unsolved}' x {solved}'

$S_0 = \{\}$

$un_1 = \{\ body = Int\ \ldots\ \}$ $\quad S_1 = \{\ top = \hat{x} \Rightarrow body\ \}$

> select "body = Int"

$un_2 = \{\ lhs = Int, \ldots\ \}$

$S_2 = \{\ body = Int,\ top = \hat{x} \Rightarrow Int\ \}$

> "lhs = Int"

$un_3 = \{\ rhs = Int,\ \hat{x} = \boxed{Int},\ rhs = Int\ \}$ $\quad \{\ lhs = Int,\ body = Int,\ top = \hat{x} \Rightarrow Int\ \}$

> "rhs = Int"

$un_4 = \{\ \hat{x} = Int,\ Int = Int\ \}$ $\quad \{\ rhs = Int\ \ lhs = Int\ \ body = Int,\ top = \hat{x} \Rightarrow Int\ \}$

> "$\hat{x}$ = Int"

$\{\ Int = Int\ \}$ $\quad \{\ \hat{x} = Int, \ldots, top = int \Rightarrow$

> "Int = Int"

$\{\}$

↓

learned: $int \Rightarrow int$

22

$((\lambda x.\ x+3)\ \text{true})$

select " Int = Bool "

$\hookrightarrow$ no type possible

$\hat{x} = \text{Int}$ $\quad \hat{x} = \text{Bool}$

$\text{Int} = \text{Bool}$ $\quad (\hat{x} = \text{Int})$

un $\quad$ sol

$\{$ $\text{top} = \hat{x} \Rightarrow \text{Bool}$

$\}$

$(\lambda x.\ \text{true})$ $\quad (\lambda x.\ x)$

polymorphic $\Longrightarrow$ constraints are "loose"

```
map f l = case l with
            [] => []
            x :: xs => (f x) :: (map f xs)
```

$\text{map} : (\hat{x} \Rightarrow \text{lhs}) \times (\text{list } \hat{x}) \Rightarrow (\text{list lhs})$

Principal Typing Theorem

---

select $(T_1 \Rightarrow T_2) = (S_1 \Rightarrow S_2)$

nothing goes in sol

add $\{$ things to unsol

$T_1 = S_1$

$T_2 = S_2$

var = rhs

rhs cannot mention var

---

type-infer : $(x\ \ x)$ $= \cancel{\{\ \hat{x} = \hat{x} \Rightarrow \text{top}\ \}}$

$= \{\ f = a \Rightarrow \text{top},$

$f = \hat{x},$

$a = \hat{x}\ \}$ $\quad = un_0$

$sol_1 = \{ f = a \Rightarrow \text{top} \}$

$un_1 = \{ a \Rightarrow \text{top} = \hat{x},$

$a = \hat{x} \}$

$sol_2 = sol_1 \cup \{ \hat{x} = a \Rightarrow \text{top} \}$

$un_2 = \{ a = a \Rightarrow \text{top} \}$

$sol_3 = \{ f = (a \Rightarrow \text{top}) \Rightarrow \text{top}$ $\quad (\quad \Rightarrow \text{top})$

$\hat{x} = (a \Rightarrow \text{top}) \Rightarrow \text{top}$

$a = (a \Rightarrow \text{top}) \Rightarrow \text{top} \}$

(23)

```
let id x = x  in
if (id true)   => { id = bool -> bool }
   (id 5)      => { id = int -> int }
   (id 6)

if ((λx.x) true)          Copy code in
   ((λx.x) 5)               type infer
   ((λx.x) 6)                   C++ / ML

if (id <bool> true)       Haskell / Typed Racket
   (id <int> 5)             (+ hacks for
   (id <int> 6)               easy cases)

let-polymorphism
        System-F
```