25-1

orig: $$\frac{\Gamma[X \mapsto T_D] \vdash M : T_R}{\Gamma \vdash \lambda X.\, M : T_D \to T_R}$$

free

new: $\lambda X. M$ into $\lambda X \underbrace{: T}_{\text{tax of type annotation}} .\ M$

"int x = 5" — Java

"x = 5" "y = 'foo'" — Python

`"List<posn<int>> l = new List<posn<int>> One(new Posn<int>(5, 6));"`

C++11 "auto x = 5" `"auto x = new List<Posn<int>> ..."`

---

fun (x) { .... $\underbrace{x + 5}$ .... }

↳ Ah ha! x is an int!

f = fun(x) { .... }

...

f(5) → Ha! x is an int!

Type Inference — ML, Haskell, Scala, Typed Racket

Constraint Generation
- look at program
- determine how values are used
- generate a system of equations

Constraint Solving
- solves the system
- returns annotations

ThePyType = num
+ str
+ array

Uni-typed

25-2 $\Gamma \vdash M : T$ (old type checker)

$\Gamma \vdash M : T \; ; C \; ; X$ (new type inference)

type enve, prog, type, system of eqns constraints, variables in the system

C = set of $T=T$

X = set of A

$$\frac{B(b) = T}{\Gamma \vdash b : T \; ; \emptyset \; ; \emptyset}$$ B is the constant type lookup

$$\frac{X \mapsto T \in \Gamma}{\Gamma \vdash X : T \; ; \emptyset \; ; \emptyset}$$

$$\frac{\Gamma \vdash M : T_m \; ; C_m \; ; X_m \qquad \Gamma \vdash N : T_N \; ; C_N \; ; X_N}{\Gamma \vdash (M \; N) : A \; ; C_m \cup C_N \cup \{T_m = T_N \to A\} \; ; X_m \cup X_N \cup \{A\}}$$ A is fresh

$$\frac{\Gamma[X \mapsto A] \vdash M : T_m ; C_m ; X_m}{\Gamma \vdash \lambda X : M : (A \to T_m) \; ; C_m \; ; X_m \cup \{A\}}$$

$$\frac{\Gamma \vdash M : T_m ; C_m ; X_m \qquad \Gamma \vdash N : T_N ; C_N ; X_n}{\Gamma \vdash (+ \; M \; N) : num \; ; C_m \cup C_n \cup \{T_m = Num, T_N = Num\} \; ; X_m \cup X_n}$$

$\lambda x.(\lambda f.$
$^{6}(^{2}(^{3}f \; ^{4}x) + \; ^{5}x)))$ (with ^{1} over $(\lambda f.$)

type: $X \to F \to Num$

constraints: $F = X \to A_2$

$A_2 = Num$

$X = Num$

vars: F, X, A_2

$$\frac{\dfrac{\Gamma_1 \vdash 4 : X ; \emptyset ; \emptyset \qquad \Gamma_1 \vdash 3 : F ; \emptyset ; \emptyset}{\Gamma_1 \vdash 2 : A_2 ; \emptyset \cup \emptyset \cup \{F = X \to A_2\} ; \{A_2\}} \qquad \Gamma_1 \vdash 5 : X ; \emptyset ; \emptyset}{\dfrac{\Gamma_1 = \emptyset[x \to X][f \to F] \vdash 6 : num \; ; \{F = X \to A_2, A_2 = Num, X = Num\} \; ; \{A_2\}}{\dfrac{\emptyset[x \to X] \vdash 1 : (F \to Num) \; ; \; '' \; ; \{A_2, F\}}{\emptyset \vdash 0 : (X \to (F \to Num)) \; ; \; '' \; ; \{A_2, F, X\}}}}$$

25-3

Systems of Linear Equations

$$\frac{x+y=16}{x+3y=8} \Rightarrow \frac{x=16-y}{x+3y=8} \Rightarrow \frac{x=16-y}{(16-y)+3y=8}$$

$$\Downarrow$$

$$\begin{matrix} y=-4 \\ x=16+4 \end{matrix} \Leftarrow \frac{x=16-y}{y=-4} \Leftarrow \frac{x=16-y}{16+2y=8}$$

$E = \cancel{N}X \mid E+E$
$\mid N$

$$\Downarrow$$

$y=-4$
$x=20$

Two sets: "solutions" and "constraints"
$X=E$ $E=E$

Algorithm:
0. Pick a constraint, $E_1 = E_2$
1. Pick a variable in it and reduce to $X = E_3$
2. Substitute X ~~for~~ with E_3 in other constraints and in the solutions
3. Add $X = E_3$ to the solutions
4. Repeat unless no constraints left

Gaussian Elimination

$$\begin{array}{c c} & \begin{matrix} x & y & c \end{matrix} \\ \begin{matrix} \text{con 1} \\ \text{con 2} \end{matrix} & \begin{bmatrix} 1 & 1 & 16 \\ 1 & 3 & 8 \end{bmatrix} \end{array}$$

$$\Downarrow$$

$$\begin{bmatrix} 1 & 1 & 16 \\ 0 & 2 & -8 \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} 1 & 1 & 16 \\ 0 & 1 & -4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 20 \\ 0 & 1 & -4 \end{bmatrix}$$

$x+y=16$
$x+3y=8$
$v+3z=20$

$$\begin{matrix} v & x & y & z & \\ 0 & 1 & 1 & 0 & 16 \\ 0 & 1 & 3 & 0 & 8 \\ 1 & 0 & 0 & 3 & 20 \end{matrix} \Rightarrow^* \begin{matrix} 0 & 1 & 0 & 0 & 20 \\ 0 & 0 & 1 & 0 & -4 \\ 1 & 0 & 0 & 3 & 20 \end{matrix} \Rightarrow^* \begin{matrix} x=20 \\ y=-4 \\ v \cancel{z} = 20-3z \\ z = \text{anything} \end{matrix}$$

has polymorphism = under-constrained

$$\begin{matrix} x+y=16 \\ x+3y=8 \\ x+2y=10 \end{matrix} \Rightarrow \begin{matrix} 1 & 1 & 16 \\ 1 & 3 & 8 \\ 1 & 2 & 10 \end{matrix} \Rightarrow \begin{matrix} 1 & 1 & 16 \\ 0 & 2 & -8 \\ 0 & 1 & -6 \end{matrix} \Rightarrow \begin{matrix} 1 & 1 & 16 \\ 0 & 1 & -4 \\ 0 & 1 & -6 \end{matrix} \Rightarrow \begin{matrix} 1 & 0 & 20 \\ 0 & 1 & -4 \\ 0 & 0 & -2 \end{matrix} = 0=-2$$

type error = over-constrained

25-4 $\mathcal{U}$ (unification) : $C \cancel{\Rightarrow}$, $C(X=T) \to C(X=T)$

If $\emptyset \vdash M : T ; C_M ; X_M$, then we want $S = \mathcal{U}(C_M, \emptyset)$

$R = T$

$\mathcal{U}(\emptyset, S) = S$

$\mathcal{U}(\{A = T\} \cup C, S) \equiv \mathcal{U}(C[A \leftarrow T], S[A \leftarrow T] \cup \{A = T\})$

$\mathcal{U}(\{T = A\} \cup C, S) = \mathcal{U}(\{A = T\} \cup C, S)$

$\mathcal{U}(\{T_1 \to T_2 = T_3 \to T_4\} \cup C, S) = \mathcal{U}(\{T_1 = T_3, T_2 = T_4\} \cup C, S)$

$\mathcal{U}(\{T = T\} \cup C, S) = \mathcal{U}(C, S)$

$\mathcal{U}(\{F = X \to A_2, A_2 = N, X = N\}, \emptyset)$ $\qquad R = X \Rightarrow F \Rightarrow N$

$= \mathcal{U}(\{A_2 = N, X = N\}, \{F = X \Rightarrow A_2\})$ $\qquad R = X \Rightarrow (X \Rightarrow A_2) \Rightarrow N$

$= \mathcal{U}(\{X = N\}, \{F = X \Rightarrow N, A_2 = N\})$ $\qquad R = X \Rightarrow (X \Rightarrow N) \Rightarrow N$

$= \mathcal{U}(\emptyset, \{F = N \Rightarrow N, A_2 = N, X = N\})$ $\qquad R = N \Rightarrow (N \Rightarrow N) \Rightarrow N$

$= \{F = N \Rightarrow N, A_2 = N, X = N\}$

Principal Typing — the most polymorphic type