CSE 114A Lecture 21

Agenda:
- Intro to type checking and type inference
- A type system for "mini-Nano"
- A tiny, terrible type inferencer (if time)

You can have a well-typed Expr that corresponds to an ill-typed program!

Plus (VBool True) (VInt 2)

(this is a Nano AST corresponding to the program

True + 2)

Our goal: statically rule out these ill-typed programs?

(\x → x) "hello" :: String

What's the type of this expression?

How do we know that this expression has type String without evaluating it?

- "hello" :: String
- when we apply a function to an argument, if the type of the argument matches the function's formal parameter type, the application expression overall has the function's return type as its type.

mini-Nano

(Numbers) (booleans) (variables) (addition)

$e ::= n \mid b \mid x \mid e_1 + e_2 \mid \backslash x \to e \mid e_1\ e_2 \mid \text{let } x = e_1 \text{ in } e_2$

(function definitions) (function calls) (let-expressions)

$T ::= \text{Int} \mid \text{Bool} \mid T_1 \to T_2$

$\backslash x \to x + x :: \text{Int} \to \text{Int}$

Lam "x" (Arith Add (Var "x") (Var "x"))

$\backslash x \to (\backslash y \to x + y) :: \text{Int} \to (\text{Int} \to \text{Int})$

$\backslash F \to F\ \text{True} :: (\text{Bool} \to \text{Bool}) \to \text{Bool}$

not :: Bool → Bool

$(\backslash F \to F\ \text{True})\ \text{not} :: \text{Bool}$

A typing relation

Gamma → $\Gamma \vdash e :: T$ "in type env. Γ, e has type T"

we'll need seven typing rules.

Let's get the easy ones today.

$e ::= n \mid b \mid x \mid e_1 + e_2 \mid \backslash x \to e \mid e_1\ e_2 \mid \text{let } x = e_1 \text{ in } e_2$

(✓ ✓)

inference rule

$$\frac{}{\Gamma \vdash n :: \text{Int}} \qquad \frac{}{\Gamma \vdash b :: \text{Bool}} \qquad \frac{e_1 :: \text{Int} \quad e_2 :: \text{Int}}{e_1 + e_2 :: \text{Int}}$$

$$\frac{\dfrac{\checkmark}{3 :: \text{Int}} \qquad \dfrac{\dfrac{\checkmark}{4 :: \text{Int}} \quad \dfrac{\checkmark}{5 :: \text{Int}}}{4 + 5 :: \text{Int}}}{3 + (4 + 5) :: \text{Int}}$$

derivation

typing judgment

$$\frac{(x, T) \text{ in } \Gamma}{\Gamma \vdash x :: T}$$

Environment:
[(x, 5), (f, closure...), (y, 7)]

Type environment:
[(x, Int), (f, Int→Int), (y, Int)]

Quiz question 1: how should we write the rule for addition expressions?

$$\frac{\Gamma \vdash e_1 :: \text{Int} \quad \Gamma \vdash e_2 :: \text{Int}}{\Gamma \vdash e_1 + e_2 :: \text{Int}}$$

x + 3