Recitation 19: Type Checking

Type System

Decides whether

e — ill typed

\ well typed

\ what is the type?

HasType(e, t, ctx)

ctx ⊢ e : t

Defn: An expression e is well-typed in context ctx if there is type t with ctx ⊢ e : t

Lexer tokens
Parser exprs
Semantic Analysis { Type Checking
Evaluator values

true + 3 → stuck

env: Var → Val

ctx: Var → Types

$\langle env, \text{let } x = 5 \text{ in } x+x \rangle \Rightarrow^{10} v$ if $\langle env[x \to 5], x+x \rangle \to^{10} v$

let f x = x + 1

f : int → int

ctx {f : int → int}

f y : int

Type System for SimPL

e ::= x | i | b | e_1 bop e_2 | if e_1 then e_2 else e_3 | let x = e_1 in e_2

bop ::= + | * | ≤

ctx : e ⊢ t

t ::= int | bool

Base Cases

$ctx \vdash i : int$
$ctx \vdash b : bool$
$\{x : t, \ldots\} \vdash x : t$

Inductive cases

Binop

if $ctx \vdash e_1 : int$
and $ctx \vdash e_2 : int$
$ctx \vdash e_1 + e_2 : int$
$ctx \vdash e_1 * e_2 : int$
$ctx \vdash e_1 \leq e_2 : bool$

Let

if $ctx \vdash e_1 : t_1$
and $ctx[e_1 \to t_1] \vdash e_2 : t$

$ctx \vdash let\ x = e_1\ in\ e_2 : t$

Conditionals

if $ctx \vdash e_1 : bool$
and $ctx \vdash e_2 : t$
and $ctx \vdash e_3 : t$
$ctx \vdash if\ e_1\ then\ e_2\ else\ e_3 : t$