COSE212: Programming Languages

Lecture 11 — Type System
(2) Design

Hakjoo Oh
2019 Fall

Language

$$
\begin{aligned}
E \;\rightarrow\; & n \\
\mid\; & x \\
\mid\; & E + E \\
\mid\; & E - E \\
\mid\; & \texttt{iszero}\; E \\
\mid\; & \texttt{if}\; E \;\texttt{then}\; E \;\texttt{else}\; E \\
\mid\; & \texttt{let}\; x = E \;\texttt{in}\; E \\
\mid\; & \texttt{proc}\; x\; E \\
\mid\; & E\; E
\end{aligned}
$$

Language

$$\frac{}{\rho \vdash n \Rightarrow n} \qquad \frac{}{\rho \vdash x \Rightarrow \rho(x)} \qquad \frac{\rho \vdash E_1 \Rightarrow n_1 \quad \rho \vdash E_2 \Rightarrow n_2}{\rho \vdash E_1 + E_2 \Rightarrow n_1 + n_2}$$

$$\frac{\rho \vdash E \Rightarrow 0}{\rho \vdash \texttt{iszero } E \Rightarrow true} \qquad \frac{\rho \vdash E \Rightarrow n}{\rho \vdash \texttt{iszero } E \Rightarrow false} \; n \neq 0$$

$$\frac{\rho \vdash E_1 \Rightarrow true \quad \rho \vdash E_2 \Rightarrow v}{\rho \vdash \texttt{if } E_1 \texttt{ then } E_2 \texttt{ else } E_3 \Rightarrow v} \qquad \frac{\rho \vdash E_1 \Rightarrow false \quad \rho \vdash E_3 \Rightarrow v}{\rho \vdash \texttt{if } E_1 \texttt{ then } E_2 \texttt{ else } E_3 \Rightarrow v}$$

$$\frac{\rho \vdash E_1 \Rightarrow v_1 \quad [x \mapsto v_1]\rho \vdash E_2 \Rightarrow v}{\rho \vdash \texttt{let } x = E_1 \texttt{ in } E_2 \Rightarrow v}$$

$$\frac{}{\rho \vdash \texttt{proc } x \; E \Rightarrow (x, E, \rho)}$$

$$\frac{\rho \vdash E_1 \Rightarrow (x, E, \rho') \quad \rho \vdash E_2 \Rightarrow v \quad [x \mapsto v]\rho' \vdash E \Rightarrow v'}{\rho \vdash E_1 \; E_2 \Rightarrow v'}$$

Types

Types are defined inductively:

$$
\begin{aligned}
T \;\rightarrow\; &\text{int} \\
\mid\; &\text{bool} \\
\mid\; &T \rightarrow T
\end{aligned}
$$

Examples:

- int
- bool
- int $\rightarrow$ int
- bool $\rightarrow$ int
- int $\rightarrow$ (int $\rightarrow$ bool)
- (int $\rightarrow$ int) $\rightarrow$ (bool $\rightarrow$ bool)
- (int $\rightarrow$ int) $\rightarrow$ (bool $\rightarrow$ (bool $\rightarrow$ int))

Types of Expressions

In order to compute the type of an expression, we need *type environment*:

$$\Gamma : Var \rightarrow T$$

Notation:

$\Gamma \vdash e : t \Leftrightarrow$ Under type environment Γ, expression e has type t.

Examples

- $[] \vdash 3 : \mathsf{int}$
- $[x \mapsto \mathsf{int}] \vdash x : \mathsf{int}$
- $[] \vdash 4 - 3 :$
- $[x \mapsto \mathsf{int}] \vdash x - 3 :$
- $[] \vdash \mathtt{iszero}\ 11 :$
- $[] \vdash \mathtt{proc}\ (x)\ (x - 11) :$
- $[] \vdash \mathtt{proc}\ (x)\ (\mathtt{let}\ y = x - 11\ \mathtt{in}\ (x - y)) :$
- $[] \vdash \mathtt{proc}\ (x)\ (\mathtt{if}\ x\ \mathtt{then}\ 11\ \mathtt{else}\ 22) :$
- $[] \vdash \mathtt{proc}\ (x)\ (\mathtt{proc}\ (y)\ \mathtt{if}\ y\ \mathtt{then}\ x\ \mathtt{else}\ 11) :$
- $[] \vdash \mathtt{proc}\ (f)\ (\mathtt{if}\ (f\ 3)\ \mathtt{then}\ 11\ \mathtt{else}\ 22) :$
- $[] \vdash (\mathtt{proc}\ (x)\ x)\ 1 :$
- $[f \mapsto \mathsf{int} \rightarrow \mathsf{int}] \vdash (f\ (f\ 1)) :$

Typing Rules

Inductive rules for assigning types to expressions:

$$\overline{\Gamma \vdash n : \mathsf{int}} \qquad \overline{\Gamma \vdash x : \Gamma(x)}$$

$$\frac{\Gamma \vdash E_1 : \mathsf{int} \quad \Gamma \vdash E_2 : \mathsf{int}}{\Gamma \vdash E_1 + E_2 : \mathsf{int}} \qquad \frac{\Gamma \vdash E_1 : \mathsf{int} \quad \Gamma \vdash E_2 : \mathsf{int}}{\Gamma \vdash E_1 - E_2 : \mathsf{int}}$$

$$\frac{\Gamma \vdash E : \mathsf{int}}{\Gamma \vdash \mathtt{iszero}\ E : \mathsf{bool}} \qquad \frac{\Gamma \vdash E_1 : \mathsf{bool} \quad \Gamma \vdash E_2 : t \quad \Gamma \vdash E_3 : t}{\Gamma \vdash \mathtt{if}\ E_1\ \mathtt{then}\ E_2\ \mathtt{else}\ E_3 : t}$$

$$\frac{\Gamma \vdash E_1 : t_1 \quad [x \mapsto t_1]\Gamma \vdash E_2 : t_2}{\Gamma \vdash \mathtt{let}\ x = E_1\ \mathtt{in}\ E_2 : t_2} \qquad \frac{\Gamma \vdash E_1 : t_1 \to t_2 \quad \Gamma \vdash E_2 : t_1}{\Gamma \vdash E_1\ E_2 : t_2}$$

$$\frac{[x \mapsto t_1]\Gamma \vdash E : t_2}{\Gamma \vdash \mathtt{proc}\ x\ E : t_1 \to t_2}$$

We say that a closed expression E has type t iff we can derive $[] \vdash E : t$.

Example 1

$$\overline{[] \vdash \texttt{iszero}\ (1+2) : \texttt{bool}}$$

Example 2

$$\overline{[] \vdash \mathtt{proc}\ (x)\ (x - 11) : \mathsf{int} \to \mathsf{int}}$$

Example 3

$$[] \vdash \texttt{proc } (x) \ (\texttt{if } x \texttt{ then } \mathbf{11} \texttt{ else } \mathbf{22}) : \texttt{bool} \rightarrow \texttt{int}$$

Example 4

$$\overline{[] \vdash (\texttt{proc } (x) \ x) \ \mathbf{1} : \texttt{int}}$$

Example 5

$$\overline{[] \vdash \mathtt{proc}\ (x)\ (\mathtt{proc}\ (y)\ \mathtt{if}\ y\ \mathtt{then}\ x\ \mathtt{else}\ \mathbf{11}) : \mathsf{int} \rightarrow (\mathsf{bool} \rightarrow \mathsf{int})}$$

Property 1 (Multiple Types)

Type assignment may not be unique:

- proc x x:

$$\frac{[x \mapsto \mathsf{int}] \vdash x : \mathsf{int}}{[] \vdash \mathtt{proc}\ x\ x : \mathsf{int} \to \mathsf{int}}$$

$$\frac{[x \mapsto \mathsf{bool}] \vdash x : \mathsf{bool}}{[] \vdash \mathtt{proc}\ x\ x : \mathsf{bool} \to \mathsf{bool}}$$

$$\frac{[x \mapsto (\mathsf{int} \to \mathsf{int})] \vdash x : \mathsf{int} \to \mathsf{int}}{[] \vdash \mathtt{proc}\ x\ x : (\mathsf{int} \to \mathsf{int}) \to (\mathsf{int} \to \mathsf{int})}$$

- proc (f) $(f\ 3)$ has type $(\mathsf{int} \to t) \to t$ for any t.
- The type of proc (f) proc (x) $(f\ (f\ x))$?

Property 2 (Soundness)

The type system is sound:

- If a closed expression E is well-typed

$$[] \vdash E : t$$

for some $t \in T$, E does not have type error and produce a value:

$$[] \vdash E \Rightarrow v$$

- Furthermore, the type of v is t. In other words, if E has a type error, we cannot find t such that $[] \vdash E : t$.
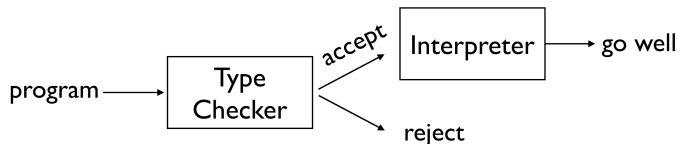- Examples:
  - (proc (x) x) 1
  - (proc (x) (x 3)) 4

Property 3 (Incompleteness)

The type system is incomplete: even though some programs do not have type errors, they do not have types according to the type system:

- if iszero 1 then 11 else (iszero 22))
- (proc (f) $(f\ f))$ (proc $x\ x$)

Implementation

Implement a type checker according to the design:

- The type checker accepts a program E only if $[] \vdash E : t$ for some t.
- Otherwise, E is rejected.