Lecture 11 — Type System

(2) Design

CSE307: Programming Languages

Minseok Jeon

2026 Spring

Language

$$
\begin{array}{rcl}
E & \to & n \\
 & | & x \\
 & | & E + E \\
 & | & E - E \\
 & | & \texttt{iszero}\ E \\
 & | & \texttt{if}\ E\ \texttt{then}\ E\ \texttt{else}\ E \\
 & | & \texttt{let}\ x = E\ \texttt{in}\ E \\
 & | & \texttt{proc}\ x\ E \\
 & | & E\ E
\end{array}
$$

Language

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

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

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

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

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

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

Types

Types are defined inductively:

$$\begin{array}{rcl} \boldsymbol{T} & \to & \mathsf{int} \\ & | & \mathsf{bool} \\ & | & \boldsymbol{T} \to \boldsymbol{T} \end{array}$$

Examples:

- int
- bool
- $\mathsf{int} \to \mathsf{int}$
- $\mathsf{bool} \to \mathsf{int}$
- $\mathsf{int} \to (\mathsf{int} \to \mathsf{bool})$
- $(\mathsf{int} \to \mathsf{int}) \to (\mathsf{bool} \to \mathsf{bool})$
- $(\mathsf{int} \to \mathsf{int}) \to (\mathsf{bool} \to (\mathsf{bool} \to \mathsf{int}))$

Types of Expressions

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

$$\boldsymbol{\Gamma} : \mathit{Var} \rightarrow \boldsymbol{T}$$

Notation:

$$\boldsymbol{\Gamma} \vdash \boldsymbol{e} : \boldsymbol{t} \Leftrightarrow \text{Under type environment } \boldsymbol{\Gamma}, \text{ expression } \boldsymbol{e} \text{ has type } \boldsymbol{t}.$$

Examples

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

Typing Rules

Inductive rules for assigning types to expressions:

$$\frac{}{\Gamma \vdash n : \text{int}} \qquad \frac{}{\Gamma \vdash x : \Gamma(x)}$$

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

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

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

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

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

Example 1

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

Example 2

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

Example 3

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

Example 4

$$\frac{}{[] \vdash (\texttt{proc}\ (x)\ x)\ 1 : \mathsf{int}}$$

Example 5

$$[] \vdash \texttt{proc}\ (x)\ (\texttt{proc}\ (y)\ \texttt{if}\ y\ \texttt{then}\ x\ \texttt{else}\ 11) : \mathsf{int} \to (\mathsf{bool} \to \mathsf{int})$$

Property 1 (Multiple Types)

Type assignment may not be unique:

- $\texttt{proc}\ x\ x$:

$$\frac{[x \mapsto \text{int}] \vdash x : \text{int}}{[] \vdash \texttt{proc}\ x\ x : \text{int} \to \text{int}}$$

$$\frac{[x \mapsto \text{bool}] \vdash x : \text{bool}}{[] \vdash \texttt{proc}\ x\ x : \text{bool} \to \text{bool}}$$

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

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

Property 2 (Soundness)

The type system is sound:

- If a closed expression $\boldsymbol{E}$ is well-typed

$$[] \vdash \boldsymbol{E} : \boldsymbol{t}$$

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

$$[] \vdash \boldsymbol{E} \Rightarrow \boldsymbol{v}$$

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

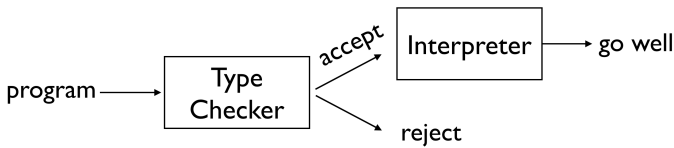
Property 3 (Incompleteness)

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 $\boldsymbol{E}$ only if $[] \vdash \boldsymbol{E} : \boldsymbol{t}$ for some $\boldsymbol{t}$.
- Otherwise, $\boldsymbol{E}$ is rejected.