ENE4014: Programming Languages

Lecture 12 — Type System
(3) Manual Type Annotation

Woosuk Lee
2024 Spring

Typing Rules

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

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

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

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

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

Implementation: First Try

Can we implement the type checker by recursively (like interpreter)?

$$
\begin{array}{l}
\text{let rec } \textbf{typeof } \boldsymbol{\Gamma}\ \boldsymbol{E} = \\
\quad \text{match } \boldsymbol{E} \text{ with} \\
\quad \mid \boldsymbol{n} \rightarrow \text{int} \\
\quad \mid \boldsymbol{x} \rightarrow \boldsymbol{\Gamma}(\boldsymbol{x}) \\
\quad \mid \boldsymbol{E_1} + \boldsymbol{E_2} \rightarrow \\
\qquad \text{let } \boldsymbol{t_1} = \textbf{typeof } \boldsymbol{\Gamma}\ \boldsymbol{E_1} \\
\qquad \text{let } \boldsymbol{t_2} = \textbf{typeof } \boldsymbol{\Gamma}\ \boldsymbol{E_2} \\
\qquad\quad \text{if } \boldsymbol{t_1} = \text{int and } \boldsymbol{t_2} = \text{int then int} \\
\qquad\quad \text{else raise TypeError} \\
\qquad\quad \vdots
\end{array}
$$

Challenge

Given a program E, how to check $[] \vdash E : t$? Nontrivial, because of the following type rule:

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

Two approaches:

- *Type Annotation*: Programmers are required to supply the type of the function argument. Used in C, C++, Java, etc.
- *Type Inference*: Type checker attempts to automatically infer types. Only possible if the language is carefully designed. Used in ML, Haskell, etc.

Language with Type Annotation

Consider the language with (recursive) procedures:

$$
\begin{array}{rcl}
E & \to & n \\
& \mid & x \\
& \mid & E + E \\
& \mid & E - E \\
& \mid & \texttt{iszero}\ E \\
& \mid & \texttt{if}\ E\ \texttt{then}\ E\ \texttt{else}\ E \\
& \mid & \texttt{let}\ x = E\ \texttt{in}\ E \\
& \mid & \texttt{proc}\ (x : t)\ E \\
& \mid & \texttt{letrec}\ t_1\ f(x : t_2) = E\ \texttt{in}\ E \\
& \mid & E\ E
\end{array}
$$

Typing Rules

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

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

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

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

$$\frac{[x \mapsto t_1]\Gamma \vdash E : t_2}{\Gamma \vdash \texttt{proc}\ (x : t_1)\ E : t_1 \to t_2} \qquad \frac{\begin{array}{c}[x \mapsto t_2, f \mapsto (t_2 \to t_1)]\Gamma \vdash E_1 : t_1 \\ [f \mapsto (t_2 \to t_1)]\Gamma \vdash E_2 : t\end{array}}{\Gamma \vdash \texttt{letrec}\ t_1\ f(x : t_2) = E_1\ \texttt{in}\ E_2 : t}$$

Example 1

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

Example 2

$$\frac{}{[\,] \vdash \texttt{letrec int } \boldsymbol{dbl}\ (\boldsymbol{x} : \mathsf{int}) = \texttt{if iszero } \boldsymbol{x} \texttt{ then } \mathbf{0} \\ \texttt{else } (\boldsymbol{dbl}\ (\boldsymbol{x} - \mathbf{1})) + \mathbf{2} \texttt{ in } \boldsymbol{dbl}\ \mathbf{2} :}$$

Example 3

$$\frac{\vdots}{[\,] \vdash \texttt{proc}\ (f : (\mathsf{bool} \to \mathsf{int}))\ \texttt{proc}\ (n : \mathsf{int})\ (f\ (\texttt{iszero}\ n))}$$

Type Check Algorithm

Now we can implement the type checking algorithm recursively:

let rec **typeof** $\boldsymbol{\Gamma}$ $\boldsymbol{E} =$
  match $\boldsymbol{E}$ with
  | $\texttt{proc}\ (\boldsymbol{x} : \boldsymbol{t_1})\ \boldsymbol{E_1} \rightarrow$

$\vdots$