COSE212: Programming Languages

Lecture 14 — Automatic Type Inference (2)

Hakjoo Oh
2024 Fall

Goal

- So far we have informally discussed how to derive type equations.
- In this lecture, we define the procedure precisely.

Language

$$
\begin{array}{rcl}
E & \rightarrow & n \\
& | & x \\
& | & E + E \\
& | & E - E \\
& | & \texttt{iszero}\ E \\
& | & \texttt{if}\ E\ \texttt{then}\ E\ \texttt{else}\ E \\
& | & \texttt{let}\ x = E\ \texttt{in}\ E \\
& | & \texttt{proc}\ x\ E \\
& | & E\ E \\
\\
T & \rightarrow & \mathsf{int} \\
& | & \mathsf{bool} \\
& | & T \rightarrow T \\
& | & \alpha\ (\in \mathit{TyVar})
\end{array}
$$

Type Equations

- Type equations are conjunctions of "type equalities": e.g.,

$$
\begin{array}{lcl}
t_0 & = & t_f \to t_1 \\
t_1 & = & t_x \to t_4 \\
t_3 & = & \mathsf{int} \\
t_4 & = & \mathsf{int} \\
t_2 & = & \mathsf{int} \\
t_f & = & \mathsf{int} \to t_3 \\
t_f & = & t_x \to t_4
\end{array}
$$

- Type equations ($TyEqn$) are defined inductively:

$$
\begin{array}{lcl}
TyEqn & \to & \emptyset \\
 & | & T \doteq T \ \wedge \ TyEqn
\end{array}
$$

Deriving Type Equations

- Algorithm for generating equations:

$$\mathcal{V} : (Var \to T) \times E \times T \to TyEqn$$

- $\mathcal{V}(\Gamma, e, t)$ generates the condition for e to have type t in Γ:

$$\Gamma \vdash e : t \text{ iff } \mathcal{V}(\Gamma, e, t) \text{ is satisfied.}$$

- Examples:
  - $\mathcal{V}([x \mapsto \text{int}], \texttt{x+1}, \alpha) =$
  - $\mathcal{V}(\emptyset, \texttt{proc}\ (x)\ (\texttt{if}\ x\ \texttt{then}\ 1\ \texttt{else}\ 2), \alpha \to \beta) =$
- To derive type equations for closed expression E, we call $\mathcal{V}(\emptyset, E, \alpha)$, where α is a fresh type variable.

Deriving Type Equations

$$
\begin{aligned}
\mathcal{V}(\Gamma, n, t) &= \\
\mathcal{V}(\Gamma, x, t) &= \\
\mathcal{V}(\Gamma, e_1 + e_2, t) &= \\
\mathcal{V}(\Gamma, \texttt{iszero}\ e, t) &= \\
\mathcal{V}(\Gamma, \texttt{if}\ e_1\ e_2\ e_3, t) &= \\
\mathcal{V}(\Gamma, \texttt{let}\ x = e_1\ \texttt{in}\ e_2, t) &= \\
\mathcal{V}(\Gamma, \texttt{proc}\ (x)\ e, t) &= \\
\\
\mathcal{V}(\Gamma, e_1\ e_2, t) &=
\end{aligned}
$$

Example

$$
\begin{array}{ll}
\mathcal{V}(\emptyset, (\texttt{proc}\ (x)\ (x))\ 1, \alpha) & \\
= \mathcal{V}(\emptyset, \texttt{proc}\ (x)\ (x), \alpha_1 \to \alpha) \wedge \mathcal{V}(\emptyset, 1, \alpha_1) & \text{new } \alpha_1 \\
= \alpha_1 \to \alpha \doteq \alpha_2 \to \alpha_3 \wedge \mathcal{V}([x \mapsto \alpha_2], x, \alpha_3) \wedge \alpha_1 \doteq \mathsf{int} & \text{new } \alpha_2, \alpha_3 \\
= \alpha_1 \to \alpha \doteq \alpha_2 \to \alpha_3 \wedge \alpha_2 \doteq \alpha_3 \wedge \alpha_1 \doteq \mathsf{int} &
\end{array}
$$

Exercise 1

$$\mathcal{V}(\emptyset, \texttt{proc}\ (f)\ (f\ 11), \alpha)$$

Exercise 2

$$\mathcal{V}([x \mapsto \text{bool}], \texttt{if } x \texttt{ then } (x - 1) \texttt{ else } 0, \alpha)$$

Exercise 3

$$\mathcal{V}(\emptyset, \texttt{proc }(f)\ (\texttt{iszero }(f\ f)), \alpha)$$

Summary

We have defined the algorithm for deriving type equations from program text:

- Given a program $\boldsymbol{E}$, call $\boldsymbol{\mathcal{V}(\emptyset, E, \alpha)}$ to derive type equations.
- Solve the equations and find the type assigned to $\boldsymbol{\alpha}$.