ENE4014: Programming Languages

Lecture 14 — Automatic Type Inference (2)

Woosuk Lee
2024 Spring

Goal

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

Language

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

$$
\begin{array}{rcl}
T & \to & \mathsf{int} \\
 & | & \mathsf{bool} \\
 & | & T \to T \\
 & | & \alpha\ (\in \mathit{TyVar})
\end{array}
$$

Type Equations

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

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

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

$$\begin{array}{rcl} TyEqn & \to & \emptyset \\ & \mid & T \doteq T \ \wedge \ TyEqn \end{array}$$

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:
  - $\mathcal{V}([x \mapsto \text{int}], \texttt{x+1}, \alpha) = \alpha \doteq \text{int}$
  - $\mathcal{V}(\emptyset, \texttt{proc}\ (x)\ (\texttt{if}\ x\ \texttt{then}\ 1\ \texttt{else}\ 2), \alpha \to \beta) =$
    $\alpha \doteq \text{bool} \land \beta \doteq \text{int}$
- 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

$$
\begin{array}{rcl}
\mathcal{V}(\Gamma, n, t) & = & t \doteq \mathsf{int} \\
\mathcal{V}(\Gamma, x, t) & = & t \doteq \Gamma(x) \\
\mathcal{V}(\Gamma, e_1 + e_2, t) & = & t \doteq \mathsf{int} \ \wedge \ \mathcal{V}(\Gamma, e_1, \mathsf{int}) \ \wedge \ \mathcal{V}(\Gamma, e_2, \mathsf{int}) \\
\mathcal{V}(\Gamma, \texttt{iszero}\ e, t) & = & t \doteq \mathsf{bool} \ \wedge \ \mathcal{V}(\Gamma, e, \mathsf{int}) \\
\mathcal{V}(\Gamma, \texttt{if}\ e_1\ e_2\ e_3, t) & = & \mathcal{V}(\Gamma, e_1, \mathsf{bool}) \ \wedge \ \mathcal{V}(\Gamma, e_2, t) \ \wedge \ \mathcal{V}(\Gamma, e_3, t) \\
\mathcal{V}(\Gamma, \texttt{let}\ x = e_1\ \texttt{in}\ e_2, t) & = & \mathcal{V}(\Gamma, e_1, \alpha) \ \wedge \ \mathcal{V}([x \mapsto \alpha]\Gamma, e_2, t) \ (\text{new } \alpha) \\
\mathcal{V}(\Gamma, \texttt{proc}\ (x)\ e, t) & = & t \doteq \alpha_1 \rightarrow \alpha_2 \ \wedge \ \mathcal{V}([x \mapsto \alpha_1]\Gamma, e, \alpha_2) \\
 & & (\text{new } \alpha_1, \alpha_2) \\
\mathcal{V}(\Gamma, e_1\ e_2, t) & = & \mathcal{V}(\Gamma, e_1, \alpha \rightarrow t) \ \wedge \ \mathcal{V}(\Gamma, e_2, \alpha) \ (\text{new } \alpha)
\end{array}
$$

Example

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

Exercise 1

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

Exercise 2

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

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