

COSE212: Programming Languages

Lecture 16 — Automatic Type Inference (2)

Hakjoo Oh
2026 Fall

Goal

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

Language: PROC

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

Type Equations

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

$$t_0 = t_f \rightarrow t_1$$

$$t_1 = t_x \rightarrow t_4$$

$$t_3 = \text{int}$$

$$t_4 = \text{int}$$

$$t_2 = \text{int}$$

$$t_f = \text{int} \rightarrow t_3$$

$$t_f = t_x \rightarrow t_4$$

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

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

Deriving Type Equations

- Algorithm for generating equations:

$$\mathcal{V} : (Id \rightarrow T) \times E \times T \rightarrow TyEqn$$

- $\mathcal{V}(\Gamma, e, t)$ 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:

- ▶ $\mathcal{V}([x \mapsto \text{int}], x+1, \alpha) =$
- ▶ $\mathcal{V}(\emptyset, \text{proc } (x) \text{ (if } x \text{ then } 1 \text{ else } 2), \alpha \rightarrow \beta) =$

- 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

$$\mathcal{V}(\Gamma, n, t) =$$

$$\mathcal{V}(\Gamma, x, t) =$$

$$\mathcal{V}(\Gamma, e_1 + e_2, t) =$$

$$\mathcal{V}(\Gamma, \text{iszero } e, t) =$$

$$\mathcal{V}(\Gamma, \text{if } e_1 \ e_2 \ e_3, t) =$$

$$\mathcal{V}(\Gamma, \text{let } x = e_1 \text{ in } e_2, t) =$$

$$\mathcal{V}(\Gamma, \text{proc } (x) \ e, t) =$$

$$\mathcal{V}(\Gamma, e_1 \ e_2, t) =$$

Example

$$\begin{aligned} & \mathcal{V}(\emptyset, (\text{proc } (x) (x)) \ 1, \alpha) \\ &= \mathcal{V}(\emptyset, \text{proc } (x) (x), \alpha_1 \rightarrow \alpha) \wedge \mathcal{V}(\emptyset, 1, \alpha_1) && \text{new } \alpha_1 \\ &= \alpha_1 \rightarrow \alpha \dot{=} \alpha_2 \rightarrow \alpha_3 \wedge \mathcal{V}([x \mapsto \alpha_2], x, \alpha_3) \wedge \alpha_1 \dot{=} \text{int} && \text{new } \alpha_2, \alpha_3 \\ &= \alpha_1 \rightarrow \alpha \dot{=} \alpha_2 \rightarrow \alpha_3 \wedge \alpha_2 \dot{=} \alpha_3 \wedge \alpha_1 \dot{=} \text{int} \end{aligned}$$

Exercise 1

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

Exercise 2

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

Exercise 3

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

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 .