4471028: 프로그래밍언어

Lecture 13 — 타입 추론 (3)
Type Inference (3)

임현승

2020 봄학기

$\mathcal{L}^{fun}$ 타입 추론

$\boxed{\textsf{\textbf{typeof}} : \mathit{Exp} \to \mathit{Type}}$

$$
\begin{array}{llcl}
\mathit{Exp} & E & ::= & n \\
 & & | & x \\
 & & | & E + E \\
 & & | & E - E \\
 & & | & \texttt{iszero}\ E \\
 & & | & \texttt{if}\ E\ \texttt{then}\ E\ \texttt{else}\ E \\
 & & | & \texttt{let}\ x = E\ \texttt{in}\ E \\
 & & | & \texttt{fun}\ x\ \texttt{->}\ E \\
 & & | & E\ E \\
\\
\mathit{Type} & T & ::= & \textsf{int} \\
 & & | & \textsf{bool} \\
 & & | & T \to T \\
 & & | & \alpha\ (\in \mathit{TyVar})
\end{array}
$$

타입 방정식 유도

- 타입 방정식(type equation):

$$\textit{TyEqn} \qquad U ::= \emptyset \mid T \doteq T \ \wedge \ U$$

- 생성 알고리즘(generation algorithm):

$$\mathcal{V} : (\textit{Var} \to \textit{Type}) \times \textit{Exp} \times \textit{Type} \to \textit{TyEqn}$$

$\mathcal{V}(\Gamma, E, T)$ generates a constraint U such that

$$\Gamma \vdash E : T$$

is true if U is satisfied.

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

타입 방정식 유도

$$
\begin{aligned}
\mathcal{V}(\Gamma, n, T) &= T \doteq \mathsf{int} \\
\mathcal{V}(\Gamma, x, T) &= T \doteq \Gamma(x) \\
\mathcal{V}(\Gamma, E_1 + E_2, T) &= T \doteq \mathsf{int} \land \mathcal{V}(\Gamma, E_1, \mathsf{int}) \land \mathcal{V}(\Gamma, E_2, \mathsf{int}) \\
\mathcal{V}(\Gamma, \texttt{iszero}\ E, T) &= T \doteq \mathsf{bool} \land \mathcal{V}(\Gamma, E, \mathsf{int}) \\
\mathcal{V}(\Gamma, \texttt{if}\ E_1\ \texttt{then}\ E_2\ \texttt{else}\ E_3, T) &= \mathcal{V}(\Gamma, E_1, \mathsf{bool}) \land \mathcal{V}(\Gamma, E_2, T) \land \mathcal{V}(\Gamma, E_3, T) \\
\mathcal{V}(\Gamma, \texttt{let}\ x = E_1\ \texttt{in}\ E_2, T) &= \mathcal{V}(\Gamma, E_1, \alpha) \land \mathcal{V}([x \mapsto \alpha]\Gamma, E_2, T)\ (\text{new } \alpha) \\
\mathcal{V}(\Gamma, \texttt{fun}\ x\ \texttt{->}\ E, T) &= T \doteq \alpha_1 \to \alpha_2 \land \mathcal{V}([x \mapsto \alpha_1]\Gamma, E, \alpha_2) \\
&\quad (\text{new } \alpha_1, \alpha_2) \\
\mathcal{V}(\Gamma, E_1\ E_2, T) &= \mathcal{V}(\Gamma, E_1, \alpha \to T) \land \mathcal{V}(\Gamma, E_2, \alpha)\ (\text{new } \alpha)
\end{aligned}
$$

연습 문제

- $\mathcal{V}(\emptyset, (\texttt{fun}\ x\ \texttt{->}\ x)\ 1, \alpha)$
- $\mathcal{V}(\emptyset, \texttt{fun}\ f\ \texttt{->}\ f\ 11, \alpha)$
- $\mathcal{V}([x \mapsto \mathsf{bool}], \texttt{if}\ x\ \texttt{then}\ (x-1)\ \texttt{else}\ 0, \alpha)$
- $\mathcal{V}(\emptyset, \texttt{fun}\ f\ \texttt{->}\ \texttt{iszero}\ (f\ f), \alpha)$

치환식(Substitution)

타입 방정식의 해는 치환식으로 표현됨:

$$S \in \mathit{Subst} = \mathit{TyVar} \to \mathit{Type}$$

치환식을 타입에 적용:

$$\begin{array}{rcl}
S(\text{int}) & = & \text{int} \\
S(\text{bool}) & = & \text{bool} \\
S(\alpha) & = & \begin{cases} T & \text{if } \alpha \mapsto T \in S \\ \alpha & \text{otherwise} \end{cases} \\
S(T_1 \to T_2) & = & S(T_1) \to S(T_2)
\end{array}$$

통합(Unification)

현재 치환식을 등식(equality) $T_1 \doteq T_2$를 이용하여 업데이트.

$$\textbf{unify} : \mathit{Type} \times \mathit{Type} \times \mathit{Subst} \to \mathit{Subst}$$

$$\begin{array}{rcl}
\textbf{unify}(\text{int}, \text{int}, S) & = & S \\
\textbf{unify}(\text{bool}, \text{bool}, S) & = & S \\
\textbf{unify}(\alpha, T, S) & = & \begin{cases} \text{fail} & \alpha \text{ occurs in } T \\ \text{extend } S \text{ with } \alpha \doteq T & \text{otherwise} \end{cases} \\
\textbf{unify}(T, \alpha, S) & = & \textbf{unify}(\alpha, T, S) \\
\textbf{unify}(T_1 \to T_2, T_1' \to T_2', S) & = & \text{let } S' = \textbf{unify}(T_1, T_1', S) \text{ in} \\
 & & \text{let } T_3 = S'(T_2) \text{ in} \\
 & & \text{let } T_4 = S'(T_2') \text{ in} \\
 & & \quad \textbf{unify}(T_3, T_4, S') \\
\textbf{unify}(_, _, _) & = & \text{fail}
\end{array}$$

Extension of S with $\alpha \doteq T$:

$$[\alpha \mapsto T]\{\alpha_1 \mapsto \{\alpha \mapsto T\}(T_1) \mid \alpha_1 \mapsto T_1 \in S\}$$

연습 문제

- **unify**$(\alpha, \text{int} \to \text{int}, \emptyset) =$
- **unify**$(\alpha, \text{int} \to \alpha, \emptyset) =$
- **unify**$(\alpha \to \beta, \text{int} \to \text{int}, \emptyset) =$
- **unify**$(\alpha \to \beta, \text{int} \to \alpha, \emptyset) =$

방정식 풀기

$$\textbf{unifyall} : \mathit{TyEqn} \times \mathit{Subst} \rightarrow \mathit{Subst}$$

$$\begin{array}{rcl}
\textbf{unifyall}(\emptyset, S) & = & S \\
\textbf{unifyall}((T_1 \doteq T_2) \ \wedge \ U, S) & = & \text{let } S' = \textbf{unify}(S(T_1), S(T_2), S) \\
 & & \text{in } \textbf{unifyall}(U, S')
\end{array}$$

typeof

$$
\begin{array}{l}
\textbf{typeof}(E) = \\
\quad \textbf{let } S = \textbf{unifyall}(\mathcal{V}(\emptyset, E, \alpha), \emptyset) \quad (\text{new } \alpha) \\
\quad \textbf{in } S(\alpha)
\end{array}
$$

연습 문제

- **typeof**$((\texttt{fun}\ x\ \texttt{->}\ x)\ 1)$
- **typeof**$(\texttt{let}\ x = 1\ \texttt{in}\ \texttt{fun}\ y\ \texttt{->}\ x + y)$

요약: 타입 추론

정적 타입 시스템의 설계와 구현:

- 타입을 추론하기 위한 선언적 논리 규칙(declarative logical rules)
- 타입을 추론하기 위한 알고리즘적 절차(algorithmic procedure)