Programmier-Paradigmen

Tutorium – Gruppe 2 & 8

Henning Dieterichs

Typinferenz

Wiederholung: Herleitbarkeit

- Für Term t, Typ τ und Typannahmen Γ heißt „$\Gamma \vdash t : \tau$“:
  - Für t kann unter Typannahmen Γ der Typ τ hergeleitet werden
- „⊢“ ist definiert durch:

$$\textsc{Const}: \frac{c \in \mathit{Const}}{\Gamma \vdash c : \tau_c} \qquad \textsc{Var}: \frac{\Gamma(x) = \tau' \qquad \tau' \succeq \tau}{\Gamma \vdash x : \tau}$$

$$\textsc{App}: \frac{\Gamma \vdash t_1 : \tau_2 \to \tau \qquad \Gamma \vdash t_2 : \tau_2}{\Gamma \vdash t_1\ t_2 : \tau} \qquad \textsc{Abs}: \frac{\Gamma, x : \tau_1 \vdash t : \tau_2 \qquad \tau_1 \text{ kein Typschema}}{\Gamma \vdash \lambda x.\ t : \tau_1 \to \tau_2}$$

$$\textsc{Let}: \frac{\Gamma \vdash t_1 : \tau_1 \qquad \Gamma, x : ta(\tau_1, \Gamma) \vdash t_2 : \tau_2}{\Gamma \vdash \texttt{let}\ x = t_1\ \texttt{in}\ t_2 : \tau_2}$$

Herleitungsbaum vs. Typinferenz

- Herleitungsbäume beweisen, dass ein Term unter gegebenen Typannahmen einen bestimmten Typ haben kann
- Typinferenz sucht einen Typ τ zu einem gegebenem Term t und Typannahmen Γ, sodass:
  - $\forall \tau' : \Gamma \vdash t : \tau' \Rightarrow \exists \sigma \in Subst_{Typ} : \tau' = \sigma(\tau)$.
  - τ heißt dann allgemeinster Typ von t.
- Typinferenz sollte korrekt und vollständig sein

Aufgabe 1: λ-Terme und die Herleitung ihrer allgemeinsten Typen

$$t_1 = \lambda \mathtt{z}.\ \mathtt{z}$$
$$t_2 = \lambda \mathtt{f}.\ \lambda \mathtt{x}.\ \mathtt{f}\ \mathtt{x}$$
$$t_3 = \lambda \mathtt{f}.\ \lambda \mathtt{x}.\ \mathtt{f}\ (\mathtt{f}\ \mathtt{x})$$
$$t_4 = \lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{y}\ (\mathtt{x}\ \mathtt{y})$$

Führen Sie für jeden dieser Terme eine Typinferenz durch. Gehen Sie dabei vor, wie auf den Folien 313ff. beschrieben:

1. Erstellen Sie zum gegebenen Term t_j zunächst einen Herleitungsbaum und verwenden Sie überall frische Typvariablen α_i.
2. Extrahieren Sie gemäß der Typisierungsregeln ein Gleichungssystem C für die α_i.
3. Bestimmen Sie einen allgemeinsten Unifikator σ_C, der C löst.
4. Bestimmen Sie einen allgemeinsten Typen von t_j als $\sigma_C(\alpha_1)$, wobei α_1 die für t_j gewählte Typvariable ist.

Aufgabe 2: Typabstraktion

In der Typabstraktion $ta(\tau, \Gamma)$ werden nicht *alle* freien Typvariablen von τ quantifiziert, sondern nur die, die nicht frei in den Typannahmen Γ vorkommen.

Überlegen Sie anhand des λ-Terms λx. **let** y = x **in** y x was passiert, wenn man diese Beschränkung aufhebt!

3 Typinferenz, `let`-Polymorphismus

Bestimmen Sie einen allgemeinsten Typ für den Ausdruck **`let`** k = λx. λy. x **`in`** k a (k b c) unter der Typannahme $\Gamma = \mathtt{a} : \mathsf{int}, \mathtt{b} : \mathsf{bool}, \mathtt{c} : \mathsf{char}$. Gehen Sie hierzu vor, wie auf den Folien 332ff. beschrieben: Extrahieren Sie für das abgedruckte Skelett einer Typherleitung die Constraint-Menge C_{let} und berechnen Sie einen allgemeinsten Unifikator $mgu(C_{let}) =: \sigma_{let}$ für die linke Teilherleitung der `let`-Regel. Bestimmen Sie dann die vereinfachte Constraint-Menge C'_{let}, Γ' sowie Constraints $C_0 \cup C_1$ für den Rest des Herleitungsbaums (C_0: Constraints, die vor Betreten des linken Teilbaums eingesammelt werden; C_1: Constraints, die nach Verlassen des linken Teilbaums eingesammelt werden). Geben Sie anschließend einen allgemeinsten Unifikator σ_C von $C := C'_{let} \cup C_0 \cup C_1$ an.

Aufgabe 3: let-Polymorphismus

- Bestimme Typ von $\texttt{let k} = \lambda\texttt{x}.\ \lambda\texttt{y}.\ \texttt{x in k a (k b c)}$
  - Unter Typannahme $\Gamma = \texttt{a} : \text{int}, \texttt{b} : \text{bool}, \texttt{c} : \text{char}$

$$\textsc{Const}: \frac{c \in \mathit{Const}}{\Gamma \vdash c : \tau_c}$$

$$\textsc{Var}: \frac{\Gamma(x) = \tau' \qquad \tau' \succeq \tau}{\Gamma \vdash x : \tau}$$

$$\textsc{App}: \frac{\Gamma \vdash t_1 : \tau_2 \to \tau \qquad \Gamma \vdash t_2 : \tau_2}{\Gamma \vdash t_1\ t_2 : \tau}$$

$$\textsc{Abs}: \frac{\Gamma, x : \tau_1 \vdash t : \tau_2 \qquad \tau_1 \text{ kein Typschema}}{\Gamma \vdash \lambda x.\ t : \tau_1 \to \tau_2}$$

$$\textsc{Let}: \frac{\Gamma \vdash t_1 : \tau_1 \qquad \Gamma, x : ta(\tau_1, \Gamma) \vdash t_2 : \tau_2}{\Gamma \vdash \texttt{let}\ x = t_1\ \texttt{in}\ t_2 : \tau_2}$$