Übungen zur "Theorie der Programmierung" - SoSe2015

Freitag 12:15-13:45, Martenstr. 3, Raum 02.133-113

Daniel Hausmann

daniel.hausmann@fau.de

Friedrich-Alexander-Universität Erlangen
Department Informatik

Lehrstuhl 8

May 26, 2015

Wiederholung aus der Vorlesung

Es sei $\underline{V}$ ein Menge von **Typvariablen**, B eine Menge von **Basistypen**

Typen

$$\alpha, \beta := a \mid \mathbf{b} \mid \alpha \rightarrow \beta$$

für $a \in \underline{V}, \mathbf{b} \in B$.

Kontext

$$\Gamma = \{x_1 : \alpha_1, \ldots, x_n : \alpha_n\}$$

für Term-Variablen $x_i \in V$ und Typen α_i mit $i \neq j \Rightarrow x_i \neq x_j$.

Schreibweise: $\Gamma \vdash t : \alpha$ (wobei $FV(t) \subseteq \{x_1, \ldots, x_n\}$).

Wiederholung aus der Vorlesung

Typisierung à la Curry:

$$(\mathrm{Ax})\ \frac{}{\Gamma \vdash x : \alpha}\ x : \alpha \in \Gamma$$

$$(\rightarrow_e)\ \frac{\Gamma \vdash t : \alpha \rightarrow \beta \qquad \Gamma \vdash s : \alpha}{\Gamma \vdash t\ s : \beta}$$

$$(\rightarrow_i)\ \frac{\Gamma, x : \alpha \vdash t : \beta}{\Gamma \vdash \lambda x.t : \alpha \rightarrow \beta}$$

Übung 8 - Typprüfung simpler Terme

Zeigen Sie, dass die folgenden Zusicherungen korrekt sind, indem Sie eine korrekte Typinferenz angeben.

1. $x : \textbf{int}, add : \textbf{int} \to \textbf{int} \to \textbf{int} \vdash \lambda y.add\ x\ (add\ x\ y) : \textbf{int} \to \textbf{int}$.
2. $length : \textbf{string} \to \textbf{int}, name : \textbf{person} \to \textbf{string} \vdash \lambda x.length\ (name\ x) : \textbf{person} \to \textbf{int}$
3. $\vdash \lambda f\ x\ y\ .\ f\ y\ x : (\textbf{int} \to \textbf{char} \to \textbf{string}) \to (\textbf{char} \to \textbf{int} \to \textbf{string})$
4. $\vdash \lambda f\ x\ y\ .\ f\ y\ x : (\alpha \to \beta \to \gamma) \to (\beta \to \alpha \to \gamma)$, für jede Wahl der Typen α, β and γ.
5. $\vdash \lambda x\ y\ .\ x : \alpha \to \beta \to \alpha$, für jede Wahl der Typen α, β and γ.

Wiederholung aus der Vorlesung

Seien Γ, Γ' Kontexte, s und t λ-Terme, α Typ.

Weakening-Lemma

$$\text{Wenn } \Gamma \vdash t : \alpha \text{ und } \Gamma \subseteq \Gamma', \text{ dann } \Gamma' \vdash t : \alpha$$

Inversionslemma

- $\Gamma \vdash x : \alpha \Rightarrow x : \alpha \in \Gamma$
- $\Gamma \vdash t\ s : \beta \Rightarrow \exists \alpha. \Gamma \vdash t : \alpha \to \beta, \Gamma \vdash s : \alpha$
- $\Gamma \vdash \lambda x.t : \gamma \Rightarrow \exists \alpha, \beta. \gamma = \alpha \to \beta, \Gamma, x : \alpha \vdash t : \beta$

Satz (Subjekt-Reduktion)

$$\text{Wenn } \Gamma \vdash t : \alpha \text{ und } t \to^* s, \text{ dann } \Gamma \vdash s : \alpha.$$

Übung 9 - Untypisierbare Terme

Verwenden Sie das Inversionslemma, um zu zeigen, daß:

1. $\nvdash \lambda x.xx : \alpha$, für jeden Typ α.
2. $y : \textbf{char} \nvdash \lambda x.yx : \alpha$, für jeden Typ α.

Wiederholung aus der Vorlesung

Prinzipaltyp

σ heißt **allgemeinste Lösung** von $\Gamma \vdash t : \alpha$ gdw. σ allgemeinste Typsubstitution mit $\Gamma\sigma \vdash t : \alpha\sigma$ ist. σ (bzw. $\sigma(a)$) heißt **Prinzipaltyp** von $\Gamma \vdash t$ gdw. σ allgemeinste Lösung von $\Gamma \vdash t : a$ ist, wobei a frisch ist.

Algorithm W [Hindley / Milner]

$PT(\Gamma; t; \alpha)$ berechnet Menge von Typgleichungen mit $\sigma = mgu(PT(\Gamma; t; \alpha))$ allgemeinste Lösung von $\Gamma \vdash t : \alpha$, **wenn** Lösung existiert.

- $PT(\Gamma; x; \alpha) = \{\alpha \doteq \beta \mid x : \beta \in \Gamma\}$
- $PT(\Gamma; \lambda x.t; \alpha) = PT(\Gamma, x : a; t; b) \cup \{a \to b \doteq \alpha\}$ für a, b frisch
- $PT(\Gamma; s\ t; \alpha) = PT(\Gamma; s; a \to \alpha) \cup PT(\Gamma; t; a)$, für a frisch

Übung 10 - Inferenz von Prinzipaltypen

Leiten Sie den Prinzipaltyp der folgenden λ-Terme unter dem jeweils gegebenen Kontext her:

1. $\Gamma = \emptyset$, $t = \lambda f\ g\ x\ .\ f\ (g\ x)$.
2. $\Gamma = \{add : \mathbf{int} \to \mathbf{int} \to \mathbf{int}, length : \mathbf{string} \to \mathbf{int}\}$,
   $t = \lambda x\ .\ add\ (length\ x)$

$IPL^{\rightarrow}$

Implikationsfragment der intuitionistischen propositionalen Logik:

$$\varphi, \psi ::= a \mid \varphi \rightarrow \psi \qquad\qquad a \in \underline{V}$$

Sequentenkalkül:

$$(\rightarrow_E)\ \frac{\Gamma \vdash \varphi \rightarrow \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi} \qquad (\rightarrow_I)\ \frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} \qquad (\mathrm{Ax})\ \frac{}{\Gamma \vdash \varphi}\ (\varphi \in \Gamma)$$

Curry-Howard-Isomorphismus

$$\vdash \varphi \Leftrightarrow \varphi \text{ inhabited}$$

Übung 11 - Programme sind Beweise?!

Problem der **type-inhabitation**: λ-Term eines gegebenen Typs α finden. Im Folgenden bezeichnen p, q und r Typvariablen.

1. Finden Sie für jeden folgenden Typ α einen Term s mit $\vdash s : \alpha$.
   1. $p \to p$
   2. $p \to (q \to p)$
   3. $(p \to (q \to r)) \to (p \to q) \to p \to r$
   4. $((p \to q) \to r) \to q \to r$
2. Prüfen Sie (beispielsweise mittels Wahrheitstafeln), ob die Typen der vorherigen Teilaufgabe – als **aussagenlogische Formeln** interpretiert – aussagenlogische Tautologien sind. Ist das Ergebnis Ihrer Prüfung angesichts der Curry-Howard-Korrespondenz eine Überraschung?
3. Verwenden Sie die Curry-Howard-Korrespondenz, um zu zeigen, dass es keinen λ-Term *coerce* gibt, sodass $\vdash \mathit{coerce} : a \to b$ (für Typvariablen a und b).