Typsysteme

Contents

Constraints aus Var, Const, Abs, App

$$\mathit{Var}\ \frac{\Gamma_{sz}(\mathtt{s}) = \underline{\alpha_4}}{\Gamma_{sz} \vdash \mathtt{s} : \underline{\alpha_8}}$$

Regel: a4 = a8

$$C\ \frac{\mathtt{true} \in \mathit{Const}}{\Gamma, \mathtt{f} : \alpha_2 \vdash \mathtt{true} : \alpha_8}$$

Regel: a8 = bool (genau so mit int)

$$\Gamma_{sz} = \mathtt{s} : \alpha_4, \mathtt{z} : \alpha_6$$
$$\Gamma_{tf} = \mathtt{t} : \alpha_{12}, \mathtt{f} : \alpha_{14}$$

$$\mathit{Abs}\ \frac{\Gamma_{\underline{sz}} \vdash \mathtt{s}\ (\mathtt{s}\ \mathtt{z}) : \underline{\alpha_7}}{\mathtt{s} : \alpha_4 \vdash (\underline{\lambda \mathtt{z}}.\ \mathtt{s}\ (\mathtt{s}\ \mathtt{z})) : \underline{\alpha_5}}$$

Regel: a5 = a6 -> a7 (typ von abgeschnietener lambda -> typ von innerem Ausdruck)

$$\mathit{Abs}\ \frac{\mathtt{x} : \alpha_2 \cancel{\vdash \lambda \mathtt{y}.\ \mathtt{y}\ (\mathtt{x}\ \mathtt{y}) :} \alpha_3}{\vdash \lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{y}\ (\mathtt{x}\ \mathtt{y}) : \alpha_1}$$

Regel: a1 = a2 -> a3 (Typ von Lambda -> Typ von Inner)

$$\mathit{App}\ \frac{\mathit{Var}\ \dfrac{}{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{y} : \alpha_6} \qquad \mathit{App}\ \dfrac{}{\cancel{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{x}\ \mathtt{y} :}\ \alpha_7}}{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{y}\ (\mathtt{x}\ \mathtt{y}) : \alpha_5}$$

Regel: a6 = a7 --> a5, es kommt weiter keine Regel für a5

Grundlagen

Funktionstypen sind rechtsassoziativ

Typen

Unsichere Programme: Typfehler bei Auswertung

z.B. im λ-Kalkül: $(\lambda x.\, x + 42)$ *true* $\Rightarrow$ *true* $+ 42 = ???$

Typisierung: Weise (möglichst vielen) sicheren Programmen Typ zu, lehne unsichere ab

Einfache Typisierung: $\vdash (\lambda x.\, 2) : \text{bool} \to \text{int}$

$\vdash (\lambda x.\, 2) : \text{int} \to \text{int}$

$\vdash (\lambda f.\, 2) : (\text{int} \to \text{int}) \to \text{int}$

Polymorphe Typen: $\vdash (\lambda x.\, 2) : \alpha \to \text{int}$

Typen (Variablenkonvention: τ)

- Basistypen: bool, int, unit, ...
- Funktionstyp: $\tau_1 \to \tau_2$
- Typvariablen: $\alpha, \alpha_1, \alpha_2, \beta, \ldots$

Funktionstypen rechtsassoziativ: $\tau_1 \to \tau_2 \to \tau_3 \equiv \tau_1 \to (\tau_2 \to \tau_3)$

Regeln (Bedeutung)

Const: Jede Konstante hat i.A ein eigenen Typ: 42 : t_42, 43 : t_43 usw. Das ersparen wir uns und sagen, dass jede Konstante einen Typ t_c hat. -> Für jede Konstante ist Typ fest.
z.B 42: int

Var: variable x hat den Typ t, wenn in Γ vermerkt ist, dass die Variable x den Typ t hat

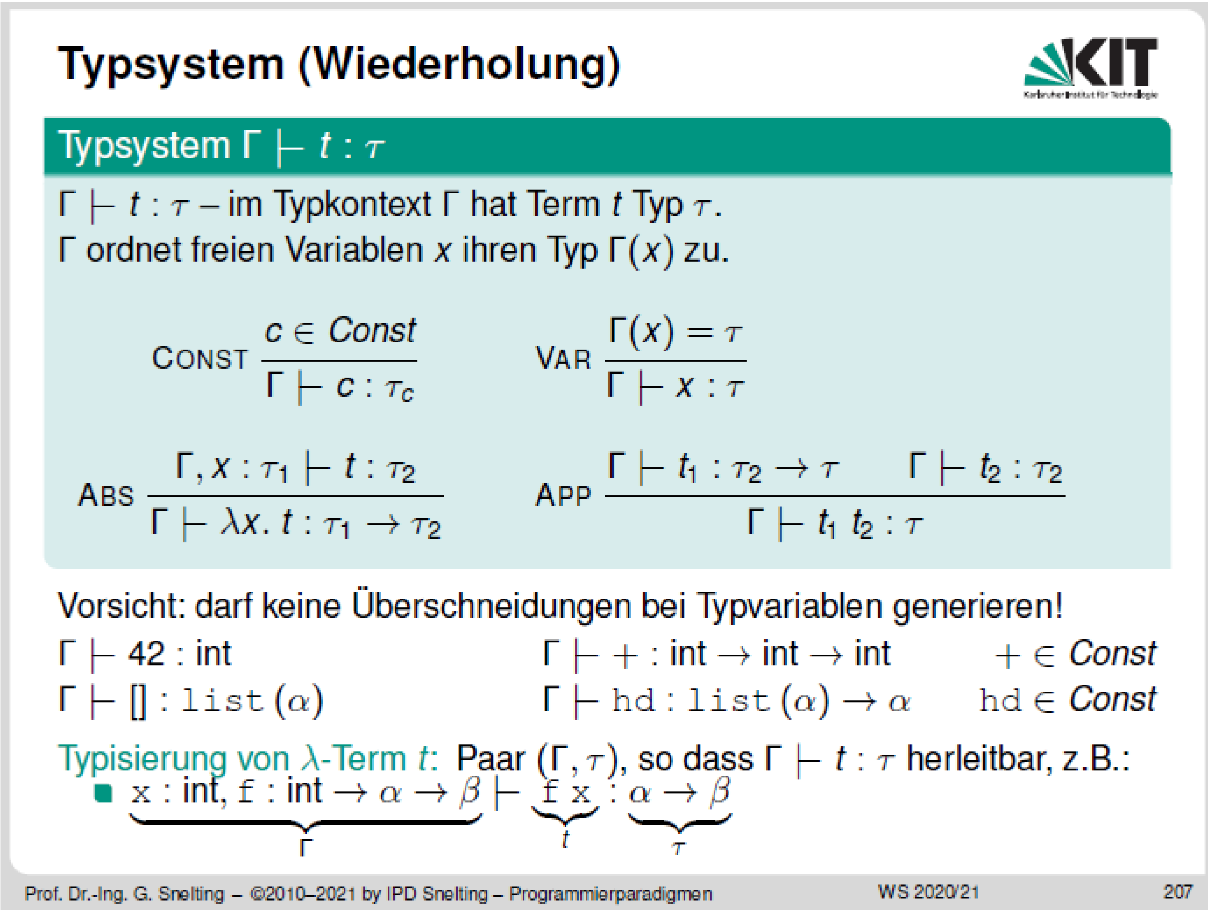
Abstraktion: (== Lambda-Abstraktion) Wir wollen den Typ des Lambda Ausdrucks bestimmen (Funktionstyp, wie t1 -> t2). Funktionsrumpf t hat Typ t2. t2 ist dann Typ des Funktionswertes (Ausgabe der Funktion). Dann ist t2 auch das Ergebnistyp der Lambda.
Normalerweise kommt x in t vor. (x ist frei in t).
In Γ soll man merken, dass x der Typ t1 hat.
Dann gilt \x . t : t1 -> t2

Also wenn gilt \x . t : t1 -> t2, dann kann man annehmen, dass x den Typ t1 hat -> nach Γ schreiben.

Applikation: Funktion t1 wird angewendet auf den aktuellen Parameter t2, und das Ding wollen wir typisieren. t1 muss ein Funktion sein

Zuerst alle Abs anwenden, dann App

Regeln (Abbildung)



Typherleitung

Man versucht Baum rückwärts zu konstruieren. Man fängt bei Zielaussage an und versucht die Voraussetzungen zu erfüllen.

Typisierbare Lambda-Terme

t ist typisierbar in Context Γ, wenn s mit Γ :- t : s existiert

Polymorhpie

Typschemata

Typschema

Für $n \in \mathbb{N}_0$ heißt $\forall\alpha_1. \ \dots \ \forall\alpha_n. \tau$ Typschema (Kürzel ϕ).
Es bindet freie Typvariablen $\alpha_1, \ \dots, \ \alpha_n$ in τ.

Beispiel: Typschema $\forall\alpha. \ \alpha \to \alpha$ steht für unendliche viele Typen, z.B.:
- int $\to$ int, bool $\to$ bool, ...
- (bool $\to$ bool) $\to$ (bool $\to$ bool), (int $\to$ bool) $\to$ (int $\to$ bool), ...

Instanziierung eines Typschemas

Für Typen $\tau_1, \ \dots, \ \tau_n$ ist der Typ $\tau\,[\alpha_1 \mapsto \tau_1, \ \dots, \ \alpha_n \mapsto \tau_n]$ eine Instanziierung vom Typschema $\forall\alpha_1. \ \dots \ \forall\alpha_n. \tau$.
Schreibweise: $(\forall\alpha_1. \ \dots \ \forall\alpha_n. \tau) \succeq \tau\,[\alpha_1 \mapsto \tau_1, \ \dots, \ \alpha_n \mapsto \tau_n]$

Zum Beispiel:
- $\forall\alpha. \ \alpha \to \alpha \succeq \text{int} \to \text{int}$
- $\forall\alpha. \ \alpha \to \alpha \succeq (\text{int} \to \text{int}) \to (\text{int} \to \text{int})$
- $\text{int} \succeq \text{int}$

Aber:
- $\alpha \to \alpha \not\succeq \text{int} \to \text{int}$
- $\alpha \not\succeq \text{bool}$
- $\forall\alpha. \ \alpha \to \alpha \not\succeq \text{bool}$

Angepasste Regeln:

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

Const-, App- und Abs-Regeln bleiben gleich.
Typschemata ϕ treten nur in Γ und in der Var-Regel auf.

`let`-Polymorphismus (Wiederholung)

Idee: `let`-gebundene Variablen getypt mit Typschema

Typabstraktion

Das Typschema $ta(\tau, \Gamma) = \forall\alpha_1. \forall\alpha_2. \ \dots \ \forall\alpha_n. \ \tau$ heißt Typabstraktion von τ relativ zu Γ, wobei $\alpha_i \in FV(\tau) \setminus FV(\Gamma)$

Alle freien Typvariablen von τ quantifiziert, die nicht frei in Typannahmen Γ
$\Rightarrow$ Verhindere Abstraktion von globalen Typvariablen im Schema

Let-Typregel

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

- Effizient (meist „quasilinear“)
- Jedoch exponentieller Worst Case!

Typinferenz für LET

$$\dfrac{\dfrac{\cdots}{\Gamma \vdash t_1 : \alpha_i} \qquad \dfrac{\cdots}{\Gamma' \vdash t_2 : \alpha_j}}{\dfrac{\Gamma \vdash \texttt{let}\ x = t_1\ \texttt{in}\ t_2 : \alpha_k}{\cdots}}$$

Allgemeine Vorgehensweise:

1. Sammle Constraints aus linkem Teilbaum in C_{let}
2. Berechne den *mgu* σ_{let} von C_{let}
3. Berechne $\Gamma' := \sigma_{let}(\Gamma), \mathrm{x} : ta(\sigma_{let}(\alpha_i), \sigma_{let}(\Gamma))$
4. Benutze Γ' in rechtem Teilbaum, sammle Constraints in C_{body}
5. Ergebnisconstraints sind $C'_{let} \cup C_{body} \cup \{\alpha_j = \alpha_k\}$ mit $C'_{let} := \{\alpha_n = \sigma_{let}(\alpha_n) \mid \sigma_{let}$ definiert für $\alpha_n\}$

Typinferenz für VAR

$$\dfrac{\Gamma(x) = \phi \quad \phi \succeq \tau}{\Gamma \vdash x : \alpha_i}$$

Allgemeine Vorgehensweise:

1. Schlage Typschema ϕ von x in Γ nach
2. Instanziiere alle $\forall$ im Typschema ϕ mit frischen Variablen, erhalte τ
3. Ergebnisconstraint ist $\alpha_i = \tau$

Herleitungsbaum für ein Lambda-Ausdruck erstellen

Ausdruck: **\x. \y. y (x y)**

Baum

$$\text{Abs}\,\dfrac{\text{Abs}\,\dfrac{\text{App}\,\dfrac{\text{Var}\,\dfrac{(\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4)\,(\mathrm{y})=\alpha_4}{\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4 \vdash \mathrm{y}:\alpha_6} \quad \text{App}\,\dfrac{\text{Var}\,\dfrac{(\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4)\,(\mathrm{x})=\alpha_2}{\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4 \vdash \mathrm{x}:\alpha_8} \quad \text{Var}\,\dfrac{(\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4)\,(\mathrm{y})=\alpha_4}{\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4 \vdash \mathrm{y}:\alpha_9}}{\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4 \vdash \mathrm{x}\ \mathrm{y}:\alpha_7}}{\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4 \vdash \mathrm{y}\ (\mathrm{x}\ \mathrm{y}):\alpha_5}}{\mathrm{x}:\alpha_2 \vdash \lambda\mathrm{y}.\ \mathrm{y}\ (\mathrm{x}\ \mathrm{y}):\alpha_3}}{\vdash \lambda\mathrm{x}.\ \lambda\mathrm{y}.\ \mathrm{y}\ (\mathrm{x}\ \mathrm{y}):\alpha_1}$$

$$C = \{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_4 = \alpha_6, \alpha_8 = \alpha_9 \to \alpha_7, \alpha_2 = \alpha_8, \alpha_4 = \alpha_9\}$$

$$\sigma_C = \begin{bmatrix} \alpha_1 \Rightarrow ((\alpha_7 \to \alpha_5) \to \alpha_7) \to (\alpha_7 \to \alpha_5) \to \alpha_5, \alpha_2 \Rightarrow (\alpha_7 \to \alpha_5) \to \alpha_7, \alpha_3 \Rightarrow (\alpha_7 \to \alpha_5) \to \alpha_5, \\ \alpha_4 \Rightarrow \alpha_7 \to \alpha_5, \alpha_6 \Rightarrow \alpha_7 \to \alpha_5, \alpha_8 \Rightarrow (\alpha_7 \to \alpha_5) \to \alpha_7, \alpha_9 \Rightarrow \alpha_7 \to \alpha_5 \end{bmatrix}$$

$$\sigma_C\,(\alpha_1) = ((\alpha_7 \to \alpha_5) \to \alpha_7) \to (\alpha_7 \to \alpha_5) \to \alpha_5$$

Schritte

1. Linke Lambda -> Abs-Regel
2. Falls es keine linke Lambda gibt, aber man der rechte Ausdruck auswerten kann -> App-Regel
3. Wenn es nur eine Variable/Konstante steht, dann Var-/Cons- Regel

Constraints für Unifikation (C):

Habe is mehrmals geprüft

Abs: a1 = a2 --> a3

$$\text{Abs}\,\dfrac{\text{Abs}\,\dfrac{}{\mathrm{x}:\alpha_2 \cancel{\vdash \lambda\mathrm{y}.\ \mathrm{y}\ (\mathrm{x}\ \mathrm{y}) :} \alpha_3}}{\vdash \lambda\mathrm{x}.\ \lambda\mathrm{y}.\ \mathrm{y}\ (\mathrm{x}\ \mathrm{y}):\alpha_1}$$

App: a6 = a7 --> a5, es kommt weiter keine Regel für a5

$$\text{Abs}\,\dfrac{\text{App}\,\dfrac{\text{Var}\,\dfrac{}{\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4 \vdash \mathrm{y}:\alpha_6} \quad \text{App}\,\dfrac{}{\cancel{\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4 \vdash \mathrm{x}\ \mathrm{y}:}\ \alpha_7}}{\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4 \vdash \mathrm{y}\ (\mathrm{x}\ \mathrm{y}):\alpha_5}}{}$$

Var/Char: a4 = a6

$$\text{Var}\,\dfrac{(\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4)\,(\mathrm{y})=\alpha_4}{\mathrm{x}:\alpha_2,\mathrm{y}:\alpha_4 \vdash \mathrm{y}:\alpha_6}$$

mgu bestimmen (σ_c)

1. Schreibe alle Variablen in einer Spalte (a1, a2, a3...)t, mache viel Platz zwischen Zeilen
2. Versuche alle Variable von hinten nach vorne zu „öffnen“, bis es keine weitere Substituion möglich ist :

a. Original: a4 = a5 -> a6, a5 = a7 -> a8
b. Subtitutiiert: a4 = (a7 -> a8) -> a6
c. Regel: a4 => (a7 -> a8) -> a6
d. **Immer klammern!**

3. Suche nach “implizite Gleichheiten”:
   a. Gegeben:
      i. a8 = a2 = a6
      ii. a6 = a7 -> a5
      iii. a8 = a9 -> a7
   b. Da a8 = a6, ist auch a7=a9 und a5=a7
   c. Neue Regeln: a7 -> a9, a5 -> a9
4. Benutze neue Wissen, um die Regeln aus 3) noch zu unifizieren
5. Öffne a1 am Ende
6. Prüfe alles noch mal

Lambda Ausdrücke Typisieren

Identität

(\x. x) ist beliebig typisierbar:

int -> int

[a] -> [a]

a -> a

(a -> a) -> (a -> a)

(a -> b -> (c -> a)) -> (a -> b -> (c -> a))

Y-Kombinator

Nicht Typisierbar

- $\omega = (\lambda \mathrm{x.\ x\ x})\ (\lambda \mathrm{x.\ x\ x})$ nicht typisierbar
  - Angenommen $\Gamma \vdash (\lambda \mathrm{x.\ x\ x})\ (\lambda \mathrm{x.\ x\ x}) : \tau$.
  - $\Rightarrow$ (App) Existiert τ' mit $\Gamma \vdash \lambda \mathrm{x.\ x\ x} : \tau' \to \tau$ (und $\Gamma \vdash \lambda \mathrm{x.\ x\ x} : \tau'$).
  - $\Rightarrow$ (Abs) $\Gamma, \mathrm{x} : \tau' \vdash \mathrm{x\ x} : \tau$
  - $\Rightarrow$ (App) $\Gamma, \mathrm{x} : \tau' \vdash \mathrm{x} : \tau' \to \tau$ und $\Gamma, \mathrm{x} : \tau' \vdash \mathrm{x} : \tau'$.
  - $\Rightarrow$ (Var) $\tau' = (\tau' \to \tau)$.
  - Typen sind endlich! $\Rightarrow$ Keine Lösung für τ'.
- Auch Y nicht typisierbar

Weitere nicht typisierbate Terme

Typisierbare λ-Terme

t typisierbar im Kontext Γ, falls τ mit $\Gamma \vdash t : \tau$ existiert.

- $(\lambda \mathtt{x}.\ \mathtt{x} + 42)$ *true* nicht typisierbar.
  - Angenommen, $\Gamma \vdash (\lambda \mathtt{x}.\ \mathtt{x} + 42)$ *true* $: \tau$.
  - $\Rightarrow$ (App) Existiert τ' mit $\Gamma \vdash$ *true* $: \tau'$ und $\Gamma \vdash \lambda \mathtt{x}.\ \mathtt{x} + 42 : \tau' \to \tau$.
  - $\Rightarrow$ (Const) $\tau' = \tau_{true} =$ bool und (Abs) $\Gamma, \mathtt{x} : \tau' \vdash \mathtt{x} + 42 : \tau$
  - $\Rightarrow$ (App) $\tau' = \tau =$ int, da $\tau_{+} =$ int $\to$ int $\to$ int. Widerspruch.

FAQ

Was macht Γ?

Γ : Tabelle, wo für jede frei Variable typ t steht

Wie sieht das richtige Var Regel aus (nicht polymorph)

$$\textit{Var}\ \frac{①:}{\mathtt{x} : \alpha_4 \vdash \mathtt{x} : \alpha_5}$$

1) (x : a4)(x) = a5

aber (x : a4)(x) = a4 wäre auch richtig.

Es kommt immer ein Constrant a4 = a5

Wie berechnet man C_0 und C_let?

$$\textit{Let}\ \dfrac{\textit{Abs}\ \dfrac{\textit{Var}\ \dfrac{(\mathtt{x} : \alpha_4)x = \alpha_5}{\mathtt{x} : \alpha_4 \vdash \mathtt{x} : \alpha_5}}{\vdash \lambda \mathtt{x}.\ \mathtt{x} : \alpha_2} \qquad \dfrac{\textit{Var}\ \dfrac{\Gamma'(\mathtt{f}) = \forall \alpha_5.\alpha_5 \to \alpha_5 \succeq \alpha_8 \to \alpha_8}{\Gamma' \vdash \mathtt{f} : \alpha_6} \qquad \dfrac{\Gamma'(\mathtt{f}) = \forall \alpha_5.\alpha_5 \to \alpha_5 \succeq \alpha_9 \to \alpha_9}{\Gamma' \vdash \mathtt{f} : \alpha_7}\ \textit{Var}}{\Gamma' \vdash \mathtt{f\ f} : \alpha_3}\ \textit{App}}{\vdash \mathbf{let}\ \mathtt{f} = \lambda \mathtt{x}.\ \mathtt{x}\ \mathbf{in}\ \mathtt{f\ f} : \alpha_1}$$

$$\textit{Let}\ \dfrac{\textit{Abs}\ \dfrac{\textit{Var}\ \dfrac{(\mathtt{x} : \alpha_4)x = \alpha_5}{\mathtt{x} : \alpha_4 \vdash \mathtt{x} : \alpha_5}}{\vdash \lambda \mathtt{x}.\ \mathtt{x} : \alpha_2} \qquad \dfrac{\textit{Var}\ \dfrac{\Gamma'(\mathtt{f}) = \forall \alpha_5.\alpha_5 \to \alpha_5 \succeq \alpha_8 \to \alpha_8}{\Gamma' \vdash \mathtt{f} : \alpha_6} \qquad \dfrac{\Gamma'(\mathtt{f}) = \forall \alpha_5.\alpha_5 \to \alpha_5 \succeq \alpha_9 \to \alpha_9}{\Gamma' \vdash \mathtt{f} : \alpha_7}\ \textit{Var}}{\Gamma' \vdash \mathtt{f\ f} : \alpha_3}\ \textit{App}}{\vdash \mathbf{let}\ \mathtt{f} = \lambda \mathtt{x}.\ \mathtt{x}\ \mathbf{in}\ \mathtt{f\ f} : \alpha_1}$$

(b) Bestimmen Sie die Typgleichungssysteme C_0 und C_{let} (vgl. Skript „Typinferenz für `let`“, Folie 324). [3 Punkte]

Beispiellösung:

$$C_0 = \{\alpha_1 = \alpha_3\}$$
$$C_{let} = \{\alpha_2 = \alpha_4 \to \alpha_5, \alpha_4 = \alpha_5\}$$

C_0 kommt also aus rechtem Teilb des LET's, und C_let aus linkem **TeilBAUM**

Wie berechnet man richtig Γ' und Polymorfe Var

$$\dfrac{\dfrac{\dfrac{①:}{\mathtt{x}:\alpha_4 \vdash \mathtt{x}:\alpha_5}\,\mathit{Var}}{\vdash \lambda\mathtt{x}.\ \mathtt{x}:\alpha_2}\,\mathit{Abs} \qquad \dfrac{\dfrac{②:}{\Gamma' \vdash \mathtt{f}:\alpha_6}\,\mathit{Var} \quad \dfrac{③:}{\Gamma' \vdash \mathtt{f}:\alpha_7}\,\mathit{Var}}{\Gamma' \vdash \mathtt{f\ f}:\alpha_3}\,\mathit{App}}{\vdash \mathbf{let}\ \mathtt{f} = \lambda\mathtt{x}.\ \mathtt{x}\ \mathbf{in}\ \mathtt{f\ f}:\alpha_1}\,\mathit{Let}$$

Verwenden Sie im Folgenden $\sigma_{let} = [\alpha_2 \Diamond \alpha_5 \to \alpha_5,\ \alpha_4 \Diamond \alpha_5]$.

(c) Berechnen Sie nun Γ' und vervollständigen Sie die Boxen ② und ③ in der gegebenen Typherleitung. [6 Punkte]

Beispiellösung:

$$\begin{aligned}\Gamma' &= \sigma_{let}(\emptyset), f : ta(\sigma_{let}(\alpha_2), \sigma_{let}(\emptyset))\\ &= f : ta(\alpha_5 \to \alpha_5, \emptyset)\\ &= f : \forall \alpha_5.\alpha_5 \to \alpha_5\end{aligned}$$
$$② : \Gamma'(f) = \forall \alpha_5.\alpha_5 \to \alpha_5 \succeq \alpha_8 \to \alpha_8$$
$$③ : \Gamma'(f) = \forall \alpha_5.\alpha_5 \to \alpha_5 \succeq \alpha_9 \to \alpha_9$$

mgu(a2) = a5 -> a5
in Γ' steht: f = FürAlle a5.a5->a5
Die variablen a8 und a9 sind neu.

Wie berechnet man das vollständige Typgleichungssystem C und mgu(C)?

Beispiellösung:

$$
\begin{aligned}
C = C_0 \cup C'_{let} \cup \{\, & \alpha_6 = \alpha_7 \to \alpha_3, \\
& \alpha_6 = \alpha_8 \to \alpha_8, \\
& \alpha_7 = \alpha_9 \to \alpha_9 \\
\} & \\
= \{\, & \alpha_1 = \alpha_3, \\
& \alpha_2 = \alpha_5 \to \alpha_5, \\
& \alpha_4 = \alpha_5, \\
& \alpha_6 = \alpha_7 \to \alpha_3, \\
& \alpha_6 = \alpha_8 \to \alpha_8, \\
& \alpha_7 = \alpha_9 \to \alpha_9 \\
\} &
\end{aligned}
$$

$$
\sigma = \begin{bmatrix}
\alpha_1 \Rightarrow \alpha_9 \to \alpha_9, \\
\alpha_2 \Rightarrow \alpha_5 \to \alpha_5, \\
\alpha_3 \Rightarrow \alpha_9 \to \alpha_9, \\
\alpha_4 \Rightarrow \alpha_5, \\
\alpha_6 \Rightarrow (\alpha_9 \to \alpha_9) \to (\alpha_9 \to \alpha_9), \\
\alpha_7 \Rightarrow \alpha_9 \to \alpha_9, \\
\alpha_8 \Rightarrow \alpha_9 \to \alpha_9
\end{bmatrix}
$$

Der vollständige Typinferenzbaum:

$$
\textit{Let}\ \dfrac{
\textit{Abs}\ \dfrac{\textit{Var}\ \dfrac{(\mathtt{x} : \alpha_4)x = \alpha_5}{\mathtt{x} : \alpha_4 \vdash \mathtt{x} : \alpha_5}}{\vdash \lambda\mathtt{x}.\ \mathtt{x} : \alpha_2}
\qquad
\dfrac{
\textit{Var}\ \dfrac{\Gamma'(\mathtt{f}) = \forall\alpha_5.\alpha_5 \to \alpha_5 \succeq \alpha_8 \to \alpha_8}{\Gamma' \vdash \mathtt{f} : \alpha_6}
\qquad
\dfrac{\Gamma'(\mathtt{f}) = \forall\alpha_5.\alpha_5 \to \alpha_5 \succeq \alpha_9 \to \alpha_9}{\Gamma' \vdash \mathtt{f} : \alpha_7}\ \textit{Var}
}{\Gamma' \vdash \mathtt{f\ f} : \alpha_3}\ \textit{App}
}{\vdash \mathbf{let}\ \mathtt{f} = \lambda\mathtt{x}.\ \mathtt{x}\ \mathbf{in}\ \mathtt{f\ f} : \alpha_1}
$$

C0, C_let (C‘_let wurde vorgegeben) und Reste in einer Menge hinzufügen.
Dann Unifizieren.

Wie berechnet man tau_poly?

(b) Betrachten Sie den Ausdruck

$$\mathbf{let}\ \mathtt{a} = \lambda\mathtt{x}.\ \lambda\mathtt{y}.\ \mathtt{true}\ \mathbf{in}\ \lambda\mathtt{f}.\ \mathtt{f}\ (\mathtt{a\ true})\ (\mathtt{a\ 17})$$

i. Wie lautet der polymorphe Typ $\tau_{\mathtt{a}}^{poly}$ von a?

Lösung

i. $\tau_{\mathtt{a}}^{poly} = \forall\alpha.\forall\beta.\alpha \to \beta \to \mathbf{bool}$

Idee: a=\x.\y.true == die Funktion, die 2 Argumente nimmt, und immer true ausgibt -> egal, was für einen Typ a und b haben.

Wie kann man kurz beweisen, dass ei nAusdruck nicht typisierbar ist?

Aufgabe:

iii. Ist der Ausdruck

$$\lambda \mathtt{a}.\ \lambda \mathtt{f}.\ \mathtt{f}\ (\mathtt{a}\ \mathtt{true})\ (\mathtt{a}\ 17)$$

typisierbar? Begründen Sie Ihre Antwort *kurz*.

Lösung:

iii. Der Ausdruck ist nicht typisierbar, da a in ihm einmal als Funktion $\mathbf{bool} \to \alpha$ und einmal als Funktion $\mathbf{int} \to \beta$ verwendet wird, was die Typregeln ohne let-Polymorphismus nicht erlauben.

Aufgaben

Typgleichungsystem aus WS16/17

ii. Unten sehen Sie einen Herleitungsbaum für einen allgemeinsten Typen unter der Typannahme $\Gamma = \mathtt{a} : \tau_{\mathtt{a}}^{poly}$. Geben Sie das zugehörige Typgleichungssystem an und ergänzen Sie außerdem, was an den mit Ⓐ bzw. Ⓑ markierten Stellen einzutragen ist.

$$
\textit{Abs}\;\dfrac{\textit{App}\;\dfrac{\textit{App}\;\dfrac{\textit{Var}\;\dfrac{(\Gamma, \mathtt{f} : \alpha_2)(\mathtt{f}) = \alpha_5}{\Gamma, \mathtt{f} : \alpha_2 \vdash \mathtt{f} : \alpha_5} \quad \textit{App}\;\dfrac{\textit{Var}\;\dfrac{\textcircled{A}}{\Gamma, \mathtt{f} : \alpha_2 \vdash \mathtt{a} : \alpha_7} \quad C\;\dfrac{\mathtt{true} \in \mathit{Const}}{\Gamma, \mathtt{f} : \alpha_2 \vdash \mathtt{true} : \alpha_8}}{\Gamma, \mathtt{f} : \alpha_2 \vdash \mathtt{a}\ \mathtt{true} : \alpha_6}}{\Gamma, \mathtt{f} : \alpha_2 \vdash \mathtt{f}\ (\mathtt{a}\ \mathtt{true}) : \alpha_4} \quad \textit{App}\;\dfrac{\textit{Var}\;\dfrac{\textcircled{B}}{\Gamma, \mathtt{f} : \alpha_2 \vdash \mathtt{a} : \alpha_{10}} \quad C\;\dfrac{17 \in \mathit{Const}}{\Gamma, \mathtt{f} : \alpha_2 \vdash 17 : \alpha_{11}}}{\Gamma, \mathtt{f} : \alpha_2 \vdash \mathtt{a}\ 17 : \alpha_9}}{\Gamma, \mathtt{f} : \alpha_2 \vdash \mathtt{f}\ (\mathtt{a}\ \mathtt{true})\ (\mathtt{a}\ 17) : \alpha_3}}{\Gamma \vdash \lambda \mathtt{f}.\ \mathtt{f}\ (\mathtt{a}\ \mathtt{true})\ (\mathtt{a}\ 17) : \alpha_1}
$$

i. $\tau_{\mathtt{a}}^{poly} = \forall \alpha. \forall \beta. \alpha \to \beta \to \mathbf{bool}$

$$
\begin{aligned}
\alpha_1 &= \alpha_2 \to \alpha_3 \\
\alpha_4 &= \alpha_9 \to \alpha_3 \\
\alpha_5 &= \alpha_6 \to \alpha_4 \\
\alpha_2 &= \alpha_5 \\
\alpha_7 &= \alpha_8 \to \alpha_6 \\
\alpha_{10} &= \alpha_{11} \to \alpha_9 \\
\alpha_8 &= \mathbf{bool} \\
\alpha_{11} &= \mathbf{int} \\
\alpha_7 &= \alpha' \to \beta' \to \mathbf{bool} \\
\alpha_{10} &= \alpha'' \to \beta'' \to \mathbf{bool}
\end{aligned}
$$

Ⓐ $: (\Gamma, \mathtt{f} : \alpha_2)\,(a) = \forall \alpha. \forall \beta. \alpha \to \beta \to \mathbf{bool} \succeq \alpha' \to \beta' \to \mathbf{bool}$

Ⓑ $: (\Gamma, \mathtt{f} : \alpha_2)\,(a) = \forall \alpha. \forall \beta. \alpha \to \beta \to \mathbf{bool} \succeq \alpha'' \to \beta'' \to \mathbf{bool}$

SS17

(a) Geben Sie einen λ-Term t vom Typ [3 Punkte]

$$\tau_t = ((\alpha \to \beta) \to (\beta \to \gamma)) \to (\alpha \to \beta) \to \alpha \to \gamma$$

an.
Beispiellösung: $t = \lambda \mathtt{f}.\ \lambda \mathtt{g}.\ \lambda \mathtt{x}.\ (\mathtt{f}\ \mathtt{g})\ (\mathtt{g}\ \mathtt{x})$

(b) Im Folgenden betrachten wir den λ-Ausdruck [16 Punkte]

$$\mathbf{let}\ \mathtt{b} = \mathtt{t}\ (\lambda \mathtt{x}.\ \mathtt{x})\ \mathbf{in}\ \lambda \mathtt{y}.\ \lambda \mathtt{z}.\ (\mathtt{b}\ \mathtt{y})\ (\mathtt{b}\ \mathtt{z})$$

i. Geben Sie den polymorphen Typen $\tau_{\mathtt{b}}^{poly}$ von $\mathtt{b}$ unter der Typannahme $\Gamma_{\mathtt{t}} = \mathtt{t} : \forall\alpha.\ \forall\beta.\ \forall\gamma.\ \tau_t$ an! (3 Punkte)
Beispiellösung: $\tau_{\mathtt{b}}^{poly} = \forall\alpha.\ (\alpha \to \alpha) \to \alpha \to \alpha$

ii. Es seien

$$\Gamma_{\mathtt{tb}} = \Gamma_{\mathtt{t}}, \mathtt{b} : \tau_{\mathtt{b}}^{poly}$$
$$\Gamma_{\mathtt{tby}} = \Gamma_{\mathtt{tb}}, \mathtt{y} : \alpha_2$$
$$\Gamma_{\mathtt{tbyz}} = \Gamma_{\mathtt{tby}}, \mathtt{z} : \alpha_4$$

Unten sehen Sie einen Herleitungsbaum für einen allgemeinsten Typen von $\lambda \mathtt{y}.\ \lambda \mathtt{z}.\ (\mathtt{b}\ \mathtt{y})\ (\mathtt{b}\ \mathtt{z})$ unter der Typannahme $\Gamma_{\mathtt{tb}}$. Geben Sie das zugehörige Typgleichungssystem an und ergänzen Sie außerdem, was an den mit (A) bzw. (B) markierten Stellen einzutragen ist. (11 Punkte)

$$
\begin{prooftree}
\AxiomC{(A)}
\LeftLabel{\textit{Var}}
\UnaryInfC{$\Gamma_{\mathtt{tbyz}} \vdash \mathtt{b} : \alpha_7$}
\AxiomC{$\Gamma_{\mathtt{tbyz}}(\mathtt{y}) = \alpha_8$}
\LeftLabel{\textit{Var}}
\UnaryInfC{$\Gamma_{\mathtt{tbyz}} \vdash \mathtt{y} : \alpha_8$}
\LeftLabel{\textit{App}}
\BinaryInfC{$\Gamma_{\mathtt{tbyz}} \vdash \mathtt{b}\ \mathtt{y} : \alpha_6$}
\AxiomC{(B)}
\LeftLabel{\textit{Var}}
\UnaryInfC{$\Gamma_{\mathtt{tbyz}} \vdash \mathtt{b} : \alpha_{10}$}
\AxiomC{$\Gamma_{\mathtt{tbyz}}(\mathtt{z}) = \alpha_{11}$}
\LeftLabel{\textit{Var}}
\UnaryInfC{$\Gamma_{\mathtt{tbyz}} \vdash \mathtt{z} : \alpha_{11}$}
\LeftLabel{\textit{App}}
\BinaryInfC{$\Gamma_{\mathtt{tbyz}} \vdash (\mathtt{b}\ \mathtt{z}) : \alpha_9$}
\LeftLabel{\textit{App}}
\BinaryInfC{$\Gamma_{\mathtt{tbyz}} \vdash (\mathtt{b}\ \mathtt{y})\ (\mathtt{b}\ \mathtt{z}) : \alpha_5$}
\LeftLabel{\textit{Abs}}
\UnaryInfC{$\Gamma_{\mathtt{tby}} \vdash \lambda \mathtt{z}.\ (\mathtt{b}\ \mathtt{y})\ (\mathtt{b}\ \mathtt{z}) : \alpha_3$}
\LeftLabel{\textit{Abs}}
\UnaryInfC{$\Gamma_{\mathtt{tb}} \vdash \lambda \mathtt{y}.\ \lambda \mathtt{z}.\ (\mathtt{b}\ \mathtt{y})\ (\mathtt{b}\ \mathtt{z}) : \alpha_1$}
\end{prooftree}
$$

Beispiellösung:

- Typgleichungen:

$$
\begin{aligned}
\alpha_1 &= \alpha_2 \to \alpha_3 \\
\alpha_3 &= \alpha_4 \to \alpha_5 \\
\alpha_6 &= \alpha_9 \to \alpha_5 \\
\alpha_7 &= \alpha_8 \to \alpha_6 \\
\alpha_7 &= (\alpha' \to \alpha') \to \alpha' \to \alpha' \\
\alpha_2 &= \alpha_8 \\
\alpha_{10} &= \alpha_{11} \to \alpha_9 \\
\alpha_{10} &= (\alpha'' \to \alpha'') \to \alpha'' \to \alpha'' \\
\alpha_4 &= \alpha_{11}
\end{aligned}
$$

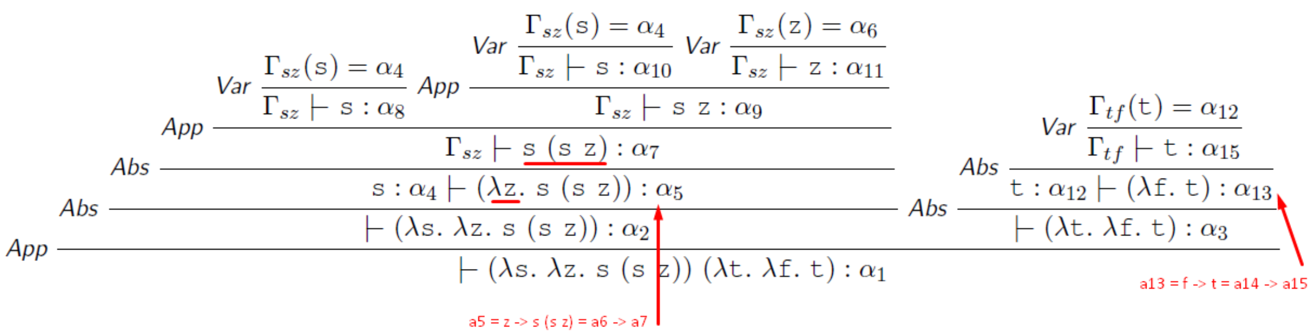
- (A): $\Gamma_{\mathtt{tbyz}}(\mathtt{b}) = \forall\alpha.\ (\alpha \to \alpha) \to \alpha \to \alpha, \forall\alpha.\ (\alpha \to \alpha) \to \alpha \to \alpha \succeq (\alpha' \to \alpha') \to \alpha' \to \alpha'$
- (B): $\Gamma_{\mathtt{tbyz}}(\mathtt{b}) = \forall\alpha.\ (\alpha \to \alpha) \to \alpha \to \alpha, \forall\alpha.\ (\alpha \to \alpha) \to \alpha \to \alpha \succeq (\alpha'' \to \alpha'') \to \alpha'' \to \alpha''$

iii. Geben Sie einen allgemeinsten Typen von $\lambda \mathtt{y}.\ \lambda \mathtt{z}.\ (\mathtt{b}\ \mathtt{y})\ (\mathtt{b}\ \mathtt{z})$ unter der Typannahme $\Gamma_{\mathtt{tb}}$ an! (2 Punkte)
Beispiellösung: $((\alpha \to \alpha) \to \alpha \to \alpha) \to (\alpha \to \alpha) \to \alpha \to \alpha$

b.2) Mit \x.x (k -> k) merkt man, dass k = (a -> b) und k = (b -> g), und also a=b=k=g

Komplizierte Abs

$$\Gamma_{sz} = \mathtt{s} : \alpha_4, \mathtt{z} : \alpha_6$$
$$\Gamma_{tf} = \mathtt{t} : \alpha_{12}, \mathtt{f} : \alpha_{14}$$



(a) Geben Sie das Constraint-System für diesen Herleitungsbaum an. [9 Punkte]

Beispiellösung:

$$\begin{aligned} \alpha_2 &= \alpha_3 \to \alpha_1 & \alpha_4 &= \alpha_{10} \\ \alpha_2 &= \alpha_4 \to \alpha_5 & \alpha_6 &= \alpha_{11} \\ \alpha_5 &= \alpha_6 \to \alpha_7 & \alpha_3 &= \alpha_{12} \to \alpha_{13} \\ \alpha_8 &= \alpha_9 \to \alpha_7 & \alpha_{13} &= \alpha_{14} \to \alpha_{15} \\ \alpha_4 &= \alpha_8 & \alpha_{12} &= \alpha_{15} \\ \alpha_{10} &= \alpha_{11} \to \alpha_9 & & \end{aligned}$$

Komplizierte mgu

(c) Geben Sie einen allgemeinsten Unifikator für die folgende Constraintmenge in Listenform $[\ldots, \alpha_i \Leftarrow \tau_i, \ldots]$ an. [6 Punkte]

$$\begin{aligned} C = \{ & \\ & \alpha_5 = \alpha_4 \to \alpha_6, \\ & \alpha_1 = \alpha_2 \to \alpha_3 \to \alpha_4, \\ & \alpha_1 = \alpha_5 \to \alpha_6, \\ \} & \end{aligned}$$

Die funktionstypen sind rechtsassoziativ: a1 = a2 -> a3 -> a4 = a2 -> (a3 -> a4)

1. Man merkt, dass es 2 Regeln für a1 gibt.

Daraus folgt, dass:

   1. a2 = a5
   2. a6 = a3 -> a4

2. Dann alles öffnen

   a2 = a5 = a4 -> (a3 -> a4)
   a6 = a3 -> a4
   a5 = a4 -> (a3 -> a4)
   a1 = (a4 -> (a3 -> a4)) -> (a3 -> a4)

3. Unifikator bilden:

   a2 => a4 -> (a3 -> a4)
   a6 => a3 -> a4
   a5 => a4 -> (a3 -> a4)
   a1 => (a4 -> (a3 -> a4)) -> (a3 -> a4)