

Theorie der Programmierung

Übung 10 - System F

Leon Vatthauer

3. Juli 2023

System F ist eine Erweiterung des einfach getypten λ -Kalküls um Polymorphie. Wir erweitern die Typgrammatik:

$$\alpha, \beta ::= a \mid \alpha \rightarrow \beta \mid \forall a. \alpha \quad (a \in \mathbf{V})$$

Die Typisierungsregeln sind:

$$\begin{array}{l} \text{(Ax)} \frac{}{\Gamma \vdash x : \alpha} \quad (x : \alpha \in \Gamma) \quad (\rightarrow_i) \frac{\Gamma[x \mapsto \alpha] \vdash t : \beta}{\Gamma \vdash \lambda x. t : \alpha \rightarrow \beta} \quad (\forall_i) \frac{\Gamma \vdash s : \alpha}{\Gamma \vdash s : \forall a. \alpha} \quad a \notin FV(\Gamma) \\ \\ (\rightarrow_e) \frac{\Gamma \vdash t : \alpha \rightarrow \beta \quad \Gamma \vdash s : \alpha}{\Gamma \vdash t s : \beta} \quad (\forall_e) \frac{\Gamma \vdash s : \forall a. \alpha}{\Gamma \vdash s : (\alpha[\beta/a])} \end{array}$$

Aufgabe 1

Produkte in System F (à la Curry)

Wir kodieren das kartesische Produkt der Typen a und b in System F unter Verwendung des Typs $(a \times b) := \forall r.(a \rightarrow b \rightarrow r) \rightarrow r$

1. Für diese Kodierung ist die “Konstruktions-Funktion” $pair$, welche aus zwei Elementen der Typen a bzw. b ein Paar des Typs $(a \times b)$ konstruiert, wie folgt definiert:

$$pair = \lambda x y.(\lambda f. f x y)$$

Zeigen Sie, dass $\vdash pair : \forall a b. a \rightarrow b \rightarrow (a \times b)$ in System F gilt.

Aufgabe 1

Produkte in System F (à la Curry)

Wir kodieren das kartesische Produkt der Typen a und b in System F unter Verwendung des Typs $(a \times b) := \forall r.(a \rightarrow b \rightarrow r) \rightarrow r$

1. Für diese Kodierung ist die “Konstruktions-Funktion” $pair$, welche aus zwei Elementen der Typen a bzw. b ein Paar des Typs $(a \times b)$ konstruiert, wie folgt definiert:

$$pair = \lambda x y.(\lambda f. f x y)$$

Zeigen Sie, dass $\vdash pair : \forall a b. a \rightarrow b \rightarrow (a \times b)$ in System F gilt.

2. Geben Sie zu jeder der folgenden Funktionssignaturen eine Implementierung der jeweiligen Funktion (d.h. einen λ -Term) an und zeigen Sie, dass Ihre Implementierung den benötigten Typ hat:

(a) $fst : \forall a b. (a \times b) \rightarrow a$

(b) $snd : \forall a b. (a \times b) \rightarrow b$

Aufgabe 1

Produkte in System F (à la Curry)

Wir kodieren das kartesische Produkt der Typen a und b in System F unter Verwendung des Typs $(a \times b) := \forall r.(a \rightarrow b \rightarrow r) \rightarrow r$

3. Schreiben Sie unter Verwendung der obigen Funktionen eine weitere Funktion

$$\text{swap} : \forall a b.(a \times b) \rightarrow (b \times a)$$

und zeigen Sie, dass sie den korrekten Typ hat. Finden Sie also einen λ -Term s , so dass $\Gamma \vdash s : \forall a b.(a \times b) \rightarrow (b \times a)$, wobei

$$\Gamma := \{ \text{pair} : \forall a b.a \rightarrow b \rightarrow (a \times b), \\ \text{fst} : \forall a b.(a \times b) \rightarrow a, \\ \text{snd} : \forall a b.(a \times b) \rightarrow b \}$$

Aufgabe 2

Listen in System F (à la Curry)

Listen können in System F unter Verwendung des folgenden Typs kodiert werden:

$$\mathbf{List} \ a := \forall r. r \rightarrow (a \rightarrow r \rightarrow r) \rightarrow r$$

In diesem Fall ergeben sich die folgenden “Konstruktor-Funktionen”:

$$\begin{aligned} \mathit{nil} &= \lambda u \ f. \ u \\ \mathit{cons} &= \lambda x \ l. \lambda u \ f. \ f \ x \ (l \ u \ f) \end{aligned}$$

Für einen gegebenen (und durch nil und cons konstruierten) Term t des Typs $\mathbf{List} \ a$ verhält sich der Term $'t \ u \ (\lambda x \ l. \ s)'$ also genau so wie eine Funktion f für die für alle x (des Typs a) und alle l (des Typs $\mathbf{List} \ a$) gilt:

$$\begin{aligned} f \ \mathit{nil} &\rightarrow_{\beta\delta}^* u \\ f \ (\mathit{cons} \ x \ l) &\rightarrow_{\beta\delta}^* s[l \mapsto f \ l] \end{aligned}$$

1. Zeigen Sie, dass:

- (a) $\vdash \mathit{nil} : \forall a. \mathbf{List} \ a$
- (b) $\vdash \mathit{cons} : \forall a. a \rightarrow \mathbf{List} \ a \rightarrow \mathbf{List} \ a$

Aufgabe 2

Listen in System F (à la Curry)

Listen können in System F unter Verwendung des folgenden Typs kodiert werden:

$$\mathbf{List} \ a \ := \ \forall r. r \rightarrow (a \rightarrow r \rightarrow r) \rightarrow r$$

In diesem Fall ergeben sich die folgenden “Konstruktor-Funktionen”:

$$\begin{aligned} \mathit{nil} &= \lambda u \ f. \ u \\ \mathit{cons} &= \lambda x \ l. \lambda u \ f. \ f \ x \ (l \ u \ f) \end{aligned}$$

2. Schreiben Sie eine Funktion `length`, welche die Länge einer Liste berechnet. Es soll gelten:

$$\begin{aligned} \mathit{length} \ \mathit{nil} &\quad \rightarrow_{\beta\delta}^* \ \mathit{zero} \\ \mathit{length} \ (\mathit{cons} \ x \ l) &\rightarrow_{\beta\delta}^* \ \mathit{succ} \ (\mathit{length} \ l) \end{aligned}$$

Zeigen Sie, dass $\Gamma_0 \vdash \mathit{length} : \forall a. \mathbf{List} \ a \rightarrow \mathbb{N}$, wobei $\Gamma_0 := \{\mathit{zero} : \mathbb{N}, \mathit{succ} : \mathbb{N} \rightarrow \mathbb{N}\}$.