Übungsblatt 12

Abgabe der Lösungen: Do, 01.08., 12:00 *per E-Mail* an die Tutoren

PRÄSENZAUFGABEN

Eine Annahme

Wenn nicht anders angegeben, bezeichnet $\Gamma \vdash s : \alpha$ die Situation, dass eine entsprechende Typinferenz im System F à la Curry existiert.

Übung 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 $(a \times b) \mapsto \forall\, r.\ (a \to b \to r) \to r$.

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 $\vdash pair\ :\ \forall a\ b\ .\ a \to b \to (a \times b)$ 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) $fst\ :\ \forall a\ b\ .\ (a \times b) \to a$

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

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

   $$swap :\ \forall a\ b\ .\ (a \times b) \to (b \times a)$$

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

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

Übung 2 Listen in System F (à la Curry)

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

$$\mathbf{List}\ a \mapsto \forall r.\ r \to (a \to r \to r) \to r$$

In diesem Fall ergeben sich die folgenden "Konstruktor-Funktionen":

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

Für einen gegebenen (und durch *nil* und *cons* konstruierten) Term t des Typs **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 **List** a) gilt:

$$\begin{array}{ll} f\ \mathit{nil} & \to^*_{\beta\delta} u \\ f\ (\mathit{cons}\ x\ l) & \to^*_{\beta\delta} s[l \mapsto f\, l] \end{array}$$

1. Zeigen Sie, dass $\vdash \mathit{nil} : \forall a.\mathbf{List}\ a$ und $\vdash \mathit{cons} : \forall a.a \to \mathbf{List}\ a \to \mathbf{List}\ a$.

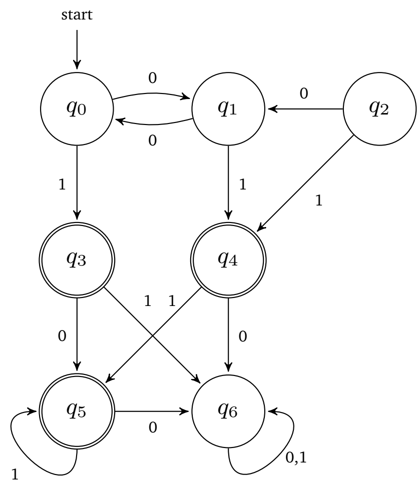
2. Schreiben Sie eine Funktion *length*, welche die Länge einer Liste berechnet. Es soll gelten:

$$\begin{array}{ll} \mathit{length}\ \mathit{nil} & \to^*_{\beta\delta} \mathit{zero} \\ \mathit{length}\ (\mathit{cons}\ x\ l) & \to^*_{\beta\delta} \mathit{succ}\ (\mathit{length}\ l) \end{array}$$

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

Übung 3 Minimierung von Automaten

1. Minimieren Sie den folgenden DFA:



2. Minimieren Sie den folgenden DFA $A = (\{q, r, s, t, u, v\}, \{a, b\}, \delta, q, \{t, u\})$ mit der tabellarisch gegebenen Zustandsüberführungsfunktion

δ	q	r	s	t	u	v
a	r	r	t	u	t	q
b	s	s	u	r	q	s

Hausaufgaben

Hinweis. Da ein Teil der Bearbeitungszeit in die vorlesungsfreie Zeit fällt, gibt es auf die Hausaufgaben keine Punkte; abgegebene Aufgaben (bitte per Mail!) werden aber auf Wunsch korrigiert. Der Stoff der Aufgaben ist natürlich dennoch klausurrelevant.

Übung 4 Listen in System F (à la Curry) II

1. Schreiben Sie eine Funktion zur Listenkonkatenation. Das heißt, schreiben Sie eine Funktion *append*, so dass:

$$\begin{aligned}&\mathit{append}\ \ \mathit{nil} && r \to^*_{\beta\delta} r\\ &\mathit{append}\ (\mathit{cons}\ x\ \ l)\ && r \to^*_{\beta\delta} \mathit{cons}\ x\ (\mathit{append}\ l\ r)\end{aligned}$$

   Zeigen Sie außerdem, dass $\Gamma_0 \vdash \mathit{append} : \forall a.\mathbf{List}\ a \to \mathbf{List}\ a \to \mathbf{List}\ a$, wobei:

$$\Gamma_0 = \{\mathit{nil} : \forall a.\mathbf{List}\ a, \mathit{cons} : \forall a.a \to \mathbf{List}\ a \to \mathbf{List}\ a\}$$

2. Schreiben Sie eine Funktion *join*, die eine Liste von Listen als Argument erwartet und die durch (von links nach rechts erfolgende) Konkatenation der einzelnen Listen entstehende Gesamtliste berechnet. Das heißt, für Ihre Funktion soll gelten:

$$\begin{aligned}&\mathit{join}\ \ \mathit{nil} && \to^*_{\beta\delta} \mathit{nil}\\ &\mathit{join}\ \ (\mathit{Cons}\ x\ \ l) && \to^*_{\beta\delta} \mathit{append}\ x\ (\mathit{join}\ l)\end{aligned}$$

   Zeigen Sie, dass $\Gamma_0 \vdash \mathit{join} : \forall a.\mathbf{List}\ (\mathbf{List}\ a) \to \mathbf{List}\ a$, wobei

$$\Gamma_0 = \{\mathit{nil} : \forall a.\mathbf{List}\ a, \mathit{append} : \forall a.\mathbf{List}\ a \to \mathbf{List}\ a \to \mathbf{List}\ a\}$$

Übung 5 Binäre Bäume in System F

In ähnlicher Weise wie für Listen kodieren wir den Typ binärer Bäume mit beschrifteten *Blättern* in System F mittels $\mathit{BinTree}\ a \mapsto \forall s.(a \to s) \to (s \to s \to s) \to s$.

1. Definieren Sie die "Konstruktor-Funktionen" *leaf* und *bin* derart, dass eine beliebige Funktion über binären Bäumen in der folgenden Form definiert werden kann:

$$\begin{aligned}&f\ \ (\mathit{leaf}\ \ x) \to^*_{\beta\delta} g\ x\\ &f\ \ (\mathit{bin}\ \ l\ \ r) \to^*_{\beta\delta} h\ (f\ l)\ (f\ \ r)\end{aligned}$$

   wobei g und h geeignete Funktionen sind.

2. Zeigen Sie, dass $\vdash \mathit{leaf} : \forall a.a \to \mathit{BinTree}\ a$.

Übung 6   Minimierung von Automaten II

Minimieren Sie den folgenden DFA mittels des tabellarischen Algorithmus aus der Vorlesung: