Theoretische Informatik II

Dr. Eva Richter / Holger Arnold
Universität Potsdam, Theoretische Informatik, Sommersemester 2008

Übungsblatt 3 (Version 4) — Abgabetermin: 13.5.2008, 12.00 Uhr

---

Der λ-Kalkül

Da verschiedene Quellen zum λ-Kalkül unterschiedliche Notationen verwenden, sind hier noch einmal die wichtigsten Definitionen zusammengefasst.

Definition 1 (λ-Terme)**.** Sei $\mathcal{V}$ eine abzählbar unendliche Menge von Variablen und sei $\mathcal{C}$ eine abzählbare Menge von Konstanten. Die Menge $\Lambda(\mathcal{V}, \mathcal{C})$ der *λ-Terme über $\mathcal{V}$ und $\mathcal{C}$* ist die Menge von Ausdrücken, die auf folgende Weise gebildet werden können:

1. Variablen und Konstanten: Jede Variable $x \in \mathcal{V}$ und jede Konstante $c \in \mathcal{C}$ ist ein λ-Term.
2. Abstraktion: Für alle Variablen x und alle λ-Terme s ist $\lambda x \cdot s$ ein λ-Term.
3. Applikation: Für alle λ-Terme s und t ist $s\ t$ ein λ-Term.
4. Klammerung: Für alle λ-Terme s ist (s) ein λ-Term.

Meist nimmt man $\mathcal{V}$ und $\mathcal{C}$ als gegeben an und spricht allgemein von *λ-Termen*. Wenn nicht anders angegeben, bezeichnen x, y, z Variablen, c Konstanten und s, t, u, v beliebige λ-Terme. Applikation ist linksassoziativ: $s\ t\ u$ entspricht $(s\ t)\ u$. Abstraktion ist rechtsassoziativ: $\lambda x \cdot \lambda y \cdot s$ entspricht $\lambda x \cdot (\lambda y \cdot s)$. Applikation bindet stärker als Abstraktion: $\lambda x \cdot s\ t$ entspricht $\lambda x \cdot (s\ t)$. Für $\lambda x \cdot \lambda y \cdot s$ schreibt man kurz $\lambda x\ y \cdot s$.

Definition 2 (Freie und gebundene Variablen)**.** Für einen λ-Term s sind die Menge $\mathrm{fv}(s)$ der freien Variablen in s und die Menge $\mathrm{bv}(s)$ der gebundenen Variablen in s definiert durch

$$\begin{aligned} \mathrm{fv}(x) &= \{x\} & \mathrm{bv}(x) &= \emptyset \\ \mathrm{fv}(c) &= \emptyset & \mathrm{bv}(c) &= \emptyset \\ \mathrm{fv}(t\ u) &= \mathrm{fv}(t) \cup \mathrm{fv}(u) & \mathrm{bv}(t\ u) &= \mathrm{bv}(t) \cup \mathrm{bv}(u) \\ \mathrm{fv}(\lambda x \cdot t) &= \mathrm{fv}(t) - \{x\} & \mathrm{bv}(\lambda x \cdot t) &= \mathrm{bv}(t) \cup \{x\}. \end{aligned}$$

Ein λ-Term heißt *geschlossen* oder *Kombinator*, wenn er keine freien Variablen enthält. Terme s und t, die sich nur in den Namen gebundener Variablen unterscheiden, werden als äquivalent betrachtet. Man bezeichnet solche Terme als *α-konvertierbar* und schreibt $s \equiv t$.

Definition 3 (Substitutionen)**.** Sei x eine Variable und t ein λ-Term. Die *Substitution von t für x*, geschrieben $[t/x]$, ist die wie folgt definierte Abbildung auf λ-Termen:

$$\begin{aligned} x[t/x] &= t & & \\ y[t/x] &= y & &\text{falls } x \neq y \\ c[t/x] &= c & & \\ (s_1\ s_2)[t/x] &= s_1[t/x]\ s_2[t/x] & & \\ (\lambda y \cdot s)[t/x] &= \lambda y \cdot s & &\text{falls } x = y \text{ oder } x \notin \mathrm{fv}(s) \\ (\lambda y \cdot s)[t/x] &= \lambda y \cdot (s[t/x]) & &\text{falls } x \neq y \text{ und } y \notin \mathrm{fv}(t) \\ (\lambda y \cdot s)[t/x] &= \lambda z \cdot (s[z/y][t/x]) & &\text{sonst, für ein } z \notin \mathrm{fv}(s) \cup \mathrm{fv}(t). \end{aligned}$$

Definition 4 (Reduktion von λ-Termen, Normalform)**.** Auf den λ-Termen ist eine Ableitungsrelation $\longrightarrow$ definiert. Die wichtigste Ableitungsregel des λ-Kalküls ist die β-Reduktion[1], die gegeben ist durch:

$$(\lambda x \cdot s)\ t \longrightarrow s[t/x].$$

Einen Teilterm eines λ-Terms, an dem eine Reduktion möglich ist, bezeichnet man als *Redex*. Ist für einen λ-Term t keine β-Reduktion mehr möglich, dann heißt t in *Normalform*. Gilt $s \xrightarrow{*} t$ für zwei λ-Terme s und t, wobei $\xrightarrow{*}$ der reflexive, transitive und substitutive Abschluss der β-Reduktion ist, dann heißt s *reduzierbar zu* t.

Nicht jeder λ-Term besitzt eine Normalform, aber wenn eine Normalform existiert, dann ist diese (bis auf α-Konversion) eindeutig. Sie kann dann dadurch erhalten werden, dass stets das Redex reduziert wird, dessen λ am weitesten links steht.

Definition 5 (λ-Gleichheit)**.** Die von α-Konversion und β-Reduktion erzeugte Kongruenzrelation $\approx$ bezeichnet man als λ-*Gleichheit*. Das heißt, $\approx$ ist der reflexive, symmetrische, transitive und substitutive Abschluss von α-Konversion und β-Reduktion.

Für zwei λ-Terme s und t gilt $s \approx t$ genau dann, wenn es eine Folge $u_1, \ldots, u_k$ von Termen und einen Term $t' \equiv t$ gibt, so dass $s \xrightarrow{*} u_1 \xleftarrow{*} u_2 \xrightarrow{*} \ldots \xleftarrow{*} u_{k-1} \xrightarrow{*} u_k \xleftarrow{*} t'$ gilt. Die Relation $\approx$ kann auch durch folgende Ableitungsregeln charakterisiert werden:

$$\alpha\text{-Konversion}\ \frac{s \equiv t}{s \approx t} \qquad \beta\text{-Reduktion}\ \frac{s \longrightarrow t}{s \approx t}$$

$$\text{Reflexivität}\ \frac{}{t \approx t} \qquad \text{Symmetrie}\ \frac{s \approx t}{t \approx s} \qquad \text{Transitivität}\ \frac{s \approx t \qquad t \approx u}{s \approx u}$$

$$\text{Substitution}_1\ \frac{s \approx t}{u\ s \approx u\ t} \qquad \text{Substitution}_2\ \frac{s \approx t}{\lambda x \cdot s \approx \lambda x \cdot t}$$

Der folgende Satz beschreibt eine wichtige Eigenschaft des λ-Kalküls:

Satz 6 (Satz von Church-Rosser)**.**

1. *Wenn $s \xrightarrow{*} t_1$ und $s \xrightarrow{*} t_2$ für Terme s, t_1 und t_2 gelten, dann gibt es Terme u und u' mit $t_1 \xrightarrow{*} u$, $t_2 \xrightarrow{*} u'$ und $u \equiv u'$.*
2. *Wenn $s_1 \approx s_2$ für Terme s_1 und s_2 gilt, dann gibt es Terme t und t' mit $s_1 \xrightarrow{*} t$, $s_2 \xrightarrow{*} t'$ und $t \equiv t'$.*
3. *Wenn $s \approx t_1$ und $s \approx t_2$ für Terme s, t_1 und t_2 gelten, wobei t_1 und t_2 in Normalform sind, dann ist $t_1 \equiv t_2$.*

[1] Es gibt je nach Variation des Kalküls noch weitere Ableitungsregeln, unter Anderem die weiter oben erwähnte α-Konversion und die η-Reduktion, die sicherstellt, dass sich gleich verhaltende Funktionsterme auf einen gemeinsamen Term reduzierbar sind. Für unsere Zwecke genügt es aber, die β-Reduktion zu betrachten. Die Bezeichnung „Reduktion“ ist dabei etwas irreführend, denn $s \xrightarrow{*} t$ bedeutet nicht, dass t in irgendeiner Weise kleiner als s sein muss.

Rekursiv definierte Funktionen im λ-Kalkül

Da Funktionen im λ-Kalkül keine Namen besitzen, benötigt man zur Konstruktion rekursiv definierter Funktionen spezielle Kombinatoren:

Definition 7. Ein λ-Term F heißt *Fixpunktkombinator*, wenn für alle λ-Terme t gilt: $\mathsf{F}\ t \approx t\ (\mathsf{F}\ t)$.

Der bekannteste Fixpunktkombinator ist der Y-Kombinator, der definiert ist als

$$\mathsf{Y} = \lambda f \cdot (\lambda x \cdot f\ (x\ x))\ (\lambda x \cdot f\ (x\ x)).$$

Eine rekursive Funktion f lässt sich durch eine Rekursionsgleichung der Form $f\ x = t[f, x]$ spezifizieren, wobei $t[f, x]$ ein λ-Term ist, der f und x als freie Variablen enthalten kann. Mit einem Fixpunktkombinator wie Y lässt sich ein λ-Term f definieren, der diese Rekursionsgleichung erfüllt:

$$f = \mathsf{Y}\ (\lambda f\ x \cdot t[f, x]).$$

Für diese Definition ist auch die Schreibweise $\mathsf{let\ rec}\ f\ x = t[f, x]$ üblich.

Darstellung von Wahrheitswerten, Paaren und Zahlen im λ-Kalkül

Wahrheitswerte Die übliche Darstellung für die Wahrheitswerte true und false im λ-Kalkül ist gegeben durch

$$\begin{aligned}\mathsf{true} &= \lambda x\ y \cdot x \text{ und}\\ \mathsf{false} &= \lambda x\ y \cdot y.\end{aligned}$$

Damit lässt sich ein Kombinator für bedingte Ausdrücke wie folgt definieren:

$$\mathsf{if} \cdot \mathsf{then} \cdot \mathsf{else} \cdot = \lambda s\ t_1\ t_2 \cdot s\ t_1\ t_2.$$

Paare Geordnete Paare lassen sich im λ-Kalkül wie folgt repräsentieren:

$$\begin{aligned}\langle \cdot, \cdot \rangle &= \lambda s\ t\ f \cdot f\ s\ t,\\ \mathsf{fst} &= \lambda p \cdot p\ \mathsf{true},\\ \mathsf{snd} &= \lambda p \cdot p\ \mathsf{false}.\end{aligned}$$

Für n-Tupel mit $n > 2$ gelte $\langle s_1, \ldots, s_n \rangle = \langle s_1, \langle s_2, \ldots, s_n \rangle \rangle$.

Natürliche Zahlen Die natürliche Zahl n kann im λ-Kalkül als die n-fache Anwendung einer Funktion auf ein Argument dargestellt werden; n wird dann dargestellt durch $\overline{n} = \lambda f\ x \cdot f^n\ x$ mit $f^0\ x = x$ und $f^{n+1}\ x = f\ (f^n\ x)$. Man bezeichnet Zahlen in dieser Darstellung als *Church-Numerale*. Die arithmetischen Funktionen succ, iszero, add und mult sind auf Church-Numeralen durch folgende λ-Terme definiert:

$$\begin{aligned}\mathsf{succ} &= \lambda n\ f\ x \cdot f\ (n\ f\ x),\\ \mathsf{iszero} &= \lambda n \cdot n\ (\lambda x \cdot \mathsf{false})\ \mathsf{true},\\ \mathsf{add} &= \lambda m\ n\ f\ x \cdot m\ f\ (n\ f\ x),\\ \mathsf{mult} &= \lambda m\ n\ f\ x \cdot m\ (n\ f)\ x.\end{aligned}$$

Für diese Terme können auch die üblichen Infix-Schreibweisen verwendet werden, also $+1$ für succ, $= 0$ für iszero, $+$ für add und $*$ für mult.

Mit den folgenden Aufgaben können Sie überprüfen, ob Sie die Darstellung von Wahrheitswerten, Paaren und Zahlen im λ-Kalkül verstanden haben.

1. Zeigen Sie, dass gilt: $\mathsf{if\ true\ then}\ t_1\ \mathsf{else}\ t_2 \approx t_1$ und $\mathsf{if\ false\ then}\ t_1\ \mathsf{else}\ t_2 \approx t_2$
2. Geben Sie λ-Terme für die logischen Operatoren and, or und not an.
3. Zeigen Sie, dass gilt: $\mathsf{fst}\ \langle s, t\rangle \approx s$ und $\mathsf{snd}\ \langle s, t\rangle \approx t$.
4. Zeigen Sie, dass gilt: $\mathsf{mult}\ \overline{0}\ \overline{n} \approx \overline{0}$, $\mathsf{mult}\ \overline{1}\ \overline{n} \approx \overline{n}$ und $\mathsf{succ} \approx \mathsf{add}\ \overline{1}$.
5. Prüfen Sie nach, dass die λ-Terme succ, iszero, add und mult tatsächlich die entsprechenden Funktionen berechnen.
6. Welche arithmetischen Funktionen auf Church-Numeralen werden durch die folgenden λ-Terme repräsentiert?
   (a) $\lambda n\ f\ x \cdot n\ f\ (f\ x)$,
   (b) $\lambda m\ n \cdot n\ m$.

Aufgabe 3.1

1. Bestimmen Sie die Normalform des λ-Terms $(\lambda f\ x \cdot f\ (f\ x))\ (\lambda y \cdot y)\ x$.
2. Zeigen Sie, dass $(\lambda f\ x \cdot f\ (f\ x))\ (\lambda y \cdot x\ y) \approx \lambda w \cdot x\ (x\ w)$ gilt.
3. Zeigen Sie, dass die λ-Terme $(\lambda x \cdot x\ x)\ (\lambda x \cdot x\ x)$ und $(\lambda x \cdot x\ x\ x)\ (\lambda x \cdot x\ x\ x)$ keine Normalform besitzen. Besitzt $(\lambda x \cdot y)\ ((\lambda x \cdot x\ x)\ (\lambda x \cdot x\ x))$ eine Normalform?

Aufgabe 3.2

Sei f folgender λ-Term:

$$f = \mathsf{Y}(\lambda f\ g\ x\ y \cdot \mathsf{if\ iszero}\ y\ \mathsf{then}\ x\ \mathsf{else}\ f\ g\ (g\ x\ y)\ (\mathsf{pred}\ y)).$$

Dabei steht pred für die weiter unten definierte Vorgängerfunktion, für die gilt: $\mathsf{pred}\ \overline{0} \approx \overline{0}$ und $\mathsf{pred}\ \overline{n+1} \approx \overline{n}$.

1. Erklären Sie, warum die Definition von f keine rekursive Gleichung ist, obwohl f auf beiden Seiten des Gleichheitszeichens steht.
2. Beschreiben Sie die Funktion, die durch den λ-Term f repräsentiert wird. Welche Funktionen berechnen die λ-Terme $f\ \mathsf{add}\ \overline{0}$ und $f\ \mathsf{mult}\ \overline{1}$?

   In der funktionalen Programmierung ist der Kombinator f als fold *bekannt.*

Aufgabe 3.3

1. Erklären Sie die Funktionsweise der durch den folgenden λ-Term definierten Vorgängerfunktion:
$$\mathsf{pred} = \lambda n \cdot \mathsf{fst}\ (n\ (\lambda p \cdot \langle \mathsf{snd}\ p, \mathsf{succ}\ (\mathsf{snd}\ p)\rangle)\ \langle \overline{0}, \overline{0}\rangle)$$
2. Geben Sie mit Hilfe von pred einen λ-Term sub an, für den gilt:
$$\mathsf{sub}\ \overline{m}\ \overline{n} \approx \begin{cases} \overline{0} & \text{falls } m \leq n, \\ \overline{m-n} & \text{sonst.} \end{cases}$$

Aufgabe 3.4

Welche Bedingung muss ein Fixpunktkombinator erfüllen? Beweisen Sie, dass der λ-Term $\mathsf{T} = (\lambda x\ y \cdot y\ (x\ x\ y))\ (\lambda x\ y \cdot y\ (x\ x\ y))$ ein Fixpunktkombinator ist.

Hausaufgabe 3.5

Beweisen oder widerlegen Sie folgende Aussage: „Für jeden λ-Term s, der eine Normalform besitzt, gilt: wenn $s \longrightarrow t_1 \longrightarrow t_2 \longrightarrow \dots$ eine von s ausgehende Folge von β-Reduktionen ist, dann sind höchstens endlich viele Terme in $\{t_1, t_2, \dots\}$ *nicht* α-konvertierbar, d.h. die Menge der Äquivalenzklassen $\{t_1, t_2, \dots\}/\equiv$ ist endlich."

Hausaufgabe 3.6

Beschreiben Sie die Funktion, die durch den folgenden λ-Term f repräsentiert wird:

$$f = \mathsf{Y}(\lambda f\ x\ y \cdot \mathsf{if\ iszero}\ y\ \mathsf{then}\ \overline{1}\ \mathsf{else\ mult}\ x\ (f\ x\ (\mathsf{pred}\ y))).$$

Hausaufgabe 3.7

Um den ganzzahligen Anteil der Quadratwurzel einer natürlichen Zahl n zu bestimmen, genügt es, zu zählen, wie viele Elemente der Folge der ungeraden Zahlen sich von n subtrahieren lassen, ohne dass das Ergebnis negativ wird (dieser Algorithmus basiert auf der sogenannten Pell-Gleichung). Zum Beispiel ist $47 - 1 - 3 - 5 - 7 - 9 - 11 = 11$, d.h. von 47 lassen sich 6 Elemente der Folge subtrahieren, $\lfloor\sqrt{47}\rfloor$ ist also 6 ($\sqrt{47} = 6.8556546\dots$). Geben Sie einen λ-Ausdruck sqrt an, für den $\mathsf{sqrt}\ \overline{n} \approx \overline{\lfloor\sqrt{n}\rfloor}$ gilt und der diesen Algorithmus implementiert. Sie können alle λ-Terme verwenden, die in diesem Übungsblatt gegeben sind.

Hausaufgaben Für jede Hausaufgabe können Sie maximal 3 Punkte bekommen. Die Punkte werden nach folgenden Regeln vergeben:

3 Punkte	= die Aufgabe wurde im Wesentlichen korrekt gelöst
2 Punkte	= die Aufgabe wurde nur teilweise gelöst
1 Punkt	= die Lösung der Aufgabe enthielt größere Fehler oder Lücken
0 Punkte	= die Aufgabe wurde nicht gelöst oder enthielt sehr viele Fehler oder Lücken

Sprechzeiten Haben Sie Fragen, Anregungen oder Probleme? Lassen Sie es uns wissen!

- Sprechen Sie in den Übungen Ihre Tutorin bzw. Ihren Tutor an.
- Holger Arnold, Raum 1.21, holger@cs.uni-potsdam.de
  Sprechzeiten: mittwochs 14.00–15.00 Uhr und nach Vereinbarung
- Dr. Eva Richter, Raum 1.25, erichter@cs.uni-potsdam.de
  Sprechzeiten: dienstags 13.30–15.00 Uhr und nach Vereinbarung