λ-Kalkül und Typen
3.Teil

Thorsten Altenkirch
email: alti@informatik.uni-muenchen.de

7. Januar 1997

3 Kombinatorische Logik

Ein wesentliches Merkmal des λ-Kalküls ist die Existenz des λ-Operators, der Variablen bindet. Aus diesem Grund paßt der λ-Kalkül nicht in das Framework der universellen Algebra. Im folgenden Abschnitt werden wir die *kombinatorische Logik* untersuchen, die auf Schönfinkel und Curry zurückgeht. Hier wird auf das λ verzichtet und stattdessen werden sogenannte Kombinatoren eingeführt, die zu geschlossenen λ-Termen korrespondieren. Kombinatorische Logik ist *algebraisch* [1] , aber dies hat seinen Preis: Wir werden einen Operator einführen, der λ-Terme in Terme der kombinatorischen Logik abbildet, aber dieser Operator erhält nur die schwache Gleichheit.

Eine andere Motivation für die kombinatorische Logik kommt aus dem Vergleich zu Logik. Die Regeln des einfach getypten λ-Kalkül entsprechen den Regeln der natürlichen Herleitug (*natural deduction*) für intuitionistische Aussagenlogik. Eine Alternative ist Hilbert's Formulierung, die eine Regel (*modus ponens*):

$$\frac{\vdash A \to B \qquad \vdash A}{\vdash B}$$

und zwei Axiome

$$\begin{array}{l} A \to B \to A \\ (A \to B \to C) \to (A \to B) \to A \to C \end{array}$$

verwendet. Dies korrespondiert zur kombinatorischen Logik.

Für die Terme der kombinatorischen Logik läßt sich eine Termersetzungssystem angeben, das Church-Rosser und stark normalisierend ist. Die starke Normalisierung ist ein Korollar aus den Resultaten des vorigen Abschnitt. Aufgrund der einfacheren Struktur der kombinatorische Logik lassen sich aber alternative Konstruktionen angeben. Wir werden einen Normalisierungsbeweis präsentieren, der mit rein semantischen Mitteln arbeitet und der zu einem interessanten Algorithmus führt. Man spricht hier vor *reduktionsfreier Normalisierung* oder *Normalisierung durch Auswertung*.

[1] Ob der λ-Kalkül algebraisch ist oder nicht, ist umstritten - siehe Peter Selinger's Aufsatz [Sel96]. Dies und ein paar mails der TYPES-mailinglist ist auf der Kurs-Webpage verfügbar.

3.1 Syntax, Typisierung und Gleichungen

Die Typen der (einfach getypten) kombinatorischen Logik sind die des einfach getypten λ-Kalküls. Wir definieren die Menge der Terme $(\mathrm{Tm}^{\mathrm{SK}})^2$ induktiv:

$$\frac{x \in V}{x \in \mathrm{Tm}^{\mathrm{SK}}} \qquad \frac{\sigma, \tau \in \mathrm{Ty}}{\mathrm{K}_{\sigma\tau} \in \mathrm{Tm}^{\mathrm{SK}}} \qquad \frac{\sigma, \tau, \rho \in \mathrm{Ty}}{\mathrm{S}_{\sigma\tau\rho} \in \mathrm{Tm}^{\mathrm{SK}}} \qquad \frac{M, N \in \mathrm{Tm}^{\mathrm{SK}}}{MN \in \mathrm{Tm}^{\mathrm{SK}}}$$

Wir präsentieren kombinatorsiche Logik mit Variablen, insbesondere um einen direkten Vergleich mit dem λ-Kalkül durchführen zu können. Es gibt allerdings keine gebundenen Variablen und die Variablen werden wie Variablen in mehrsortigen Algebren verwendet. Insbesondere verwenden wir auch Substitution $M[x := N]$ in diesem Sinne.

Wir definieren die Relation $- \vdash^{\mathrm{SK}} - : - \subseteq \mathrm{Con} \times \mathrm{Tm}^{\mathrm{SK}} \times \mathrm{Ty}$ induktiv, wobei die Regeln **var** und **app** vom einfach getypten λ-Kalkül

$$\frac{}{\Gamma, x : \sigma, \Gamma' \vdash^{\mathrm{SK}} x : \sigma}\,\mathbf{var} \qquad \frac{\Gamma \vdash^{\mathrm{SK}} M : \sigma \to \tau \qquad \Gamma \vdash^{\mathrm{SK}} N : \sigma}{\Gamma \vdash^{\mathrm{SK}} MN : \tau}\,\mathbf{app}$$

um folgende Axiome erweitert wird:

$$\frac{}{\Gamma \vdash^{\mathrm{SK}} \mathrm{K}_{\sigma\tau} : \sigma \to \tau \to \sigma}\,\mathrm{K} \qquad \frac{}{\Gamma \vdash^{\mathrm{SK}} \mathrm{S}_{\sigma\tau\rho} : (\sigma \to \tau \to \rho) \to (\sigma \to \tau) \to \sigma \to \rho}\,\mathrm{S}$$

Satz 1.10 (Eindeutigkeit der Typisierung) gilt auch für kombinatorische Logik (hier ist die Markierung der Kombinatoren essentiell). Ebenso lassen sich die Lemmas 1.11 - 1.14 sinngemäß für $\vdash_{\mathrm{SK}}$ zeigen. Tatsächlich ist der Beweis einfacher, da es keine gebundenen Variablen gibt.

Analog zum einfach getypten λ-Kalkül werden Gleichungstheorien definiert. Wir führen das Axiomensystem SK ein, wobei wir folgende Gleichungen für die Kombinatoren verlangen:

$$\frac{\Gamma \vdash^{\mathrm{SK}} M : \sigma \qquad \Gamma \vdash^{\mathrm{SK}} N : \tau}{\Gamma \rhd \mathrm{K}_{\sigma\tau} MN = M : \sigma}\,\mathrm{K}^{=}$$

$$\frac{\Gamma \vdash^{\mathrm{SK}} M : \sigma \to \tau \to \rho \qquad \Gamma \vdash^{\mathrm{SK}} N : \sigma \to \tau \qquad \Gamma \vdash^{\mathrm{SK}} L : \sigma}{\Gamma \rhd \mathrm{S}_{\sigma\tau\rho} MNL = ML(NL) : \rho}\,\mathrm{S}^{=}$$

Wir definieren das Axiomensystem $\mathrm{Ax}_{\mathrm{SK}}$, wobei wir einige struturelle Regeln des λ-Kalküls für die kombinatorsiche Logik übernehmen:

$$\mathrm{Ax}_{\mathrm{SK}} = \{\mathbf{Refl}, \mathbf{Sym}, \mathbf{Trans}, \mathbf{Weak}, \mathbf{App}, \mathbf{Subst}, \mathrm{K}^{=}, \mathrm{S}^{=}\}$$

Swap läßt sich aus **Weak** und **Subst** herleiten.

Aufgabe 3.1 *Definiere einen Kombinator* $\mathrm{I}^{\sigma} = \mathrm{SKK} : \sigma \to \sigma$ *und gebe eine Typherleitung an und zeige, daß für alle* $\Gamma \vdash M : \sigma$ *gilt:*

$$\vdash_{\mathrm{SK}} \Gamma \rhd \mathrm{I}^{\sigma} M = M : \sigma$$

3.1.1 Probleme mit Subst

Die Inklusion der **Subst**-Regel in Ax_{W} (in 1.4.) ist nicht korrekt, da sich die ξ-Regel aus **Subst** herleiten läßt (die beiden Regeln sind also äquivalent, da wir bereits gezeigt haben, daß sich **Subst** aus ξ herleiten läßt (Lemma 1.15)). Eine korrekte

[2] Da es keine gebundenen Variablen gibt, brauchen wir auch nicht zwischen Termen und Termbäumen zu unterscheiden.

Formalisierung von $\mathrm{Ax_W}$ muß einen expliziten Substitutionsoperator verwenden, d.h. Substitution als termformende Operation.

Um diese Schwierigkeit zu vermeiden, aber denoch $\mathrm{Ax_W}$ und $\mathrm{Ax_{SK}}$ vergleichen zu können, verwenden wir auch eine Version von $\mathrm{Ax_{SK}}$ ohne die Substitutionsregel:

$$\mathrm{Ax_{SK'}} = \{\mathbf{Refl}, \mathbf{Sym}, \mathbf{Trans}, \mathbf{Weak}, \mathbf{App}, \mathbf{Swap}, \mathrm{K}^=, \mathrm{S}^=\}$$

3.2 Vergleich mit dem schwachen λ-Kalkül

Wir definieren zunächst eine Übersetzung $-^\S : \mathrm{Tm}^{\mathrm{SK}} \to \mathrm{Tm}$, die Termen der kombinatorischen Logik Λ-Terme zuordnet:

$$\begin{array}{rcl} (\mathrm{K}_{\sigma\tau})^\S & = & \lambda x:\sigma.\lambda y:\tau.x \\ (\mathrm{S}_{\sigma\tau\rho})^\S & = & \lambda x:\sigma\to\tau\to\rho.\lambda y:\sigma\to\tau.\lambda z:\sigma.(xz)yz \\ (MN)^\S & = & M^\S N^\S \end{array}$$

Lemma 3.1 (Korrektheit von $-^\S$)

1.

$$\frac{\Gamma \vdash^{\mathrm{SK}} M:\sigma}{\Gamma \vdash M^\S : \sigma}$$

2.

$$\frac{\mathcal{E} \vdash_{\mathrm{SK}} \Gamma \rhd M = N : \sigma}{\mathcal{E}^\S \vdash_{\mathrm{W}} \Gamma \rhd M^\S = N^\S : \sigma}$$

wobei wir mit $\mathcal{E}^\S$ die Anwendung von $-^{\mathrm{S}}$ auf alle Terme in $\mathcal{E}$ denotieren.

Beweis: Durch Induktion über $\vdash^{\mathrm{SK}}$, bzw. über die Struktur von Herleitungen mit $\vdash_{\mathrm{SK}}$. Insbesondere können $\mathrm{K}^=$ und $\mathrm{S}^=$ nur mit β und **App** hergeleitet werden. □

Bevor wir eine Übersetzung in die andere Richtung definieren, führen wir einen Abstraktionsoperator ein, der λ-Abstraktion in Kombinatoren übersetzt. Gegeben $\Gamma, x:\sigma \vdash^{\mathrm{SK}} M:\tau$ definieren wir $\Gamma \vdash \lambda^* x:\sigma.M : \sigma\to\tau$ durch Induktion über die Struktur der Herleitung:

var Wenn $M = x$ dann

$$\lambda^* x:\sigma.x = I_\sigma,$$

sonst

$$\lambda^* x:\sigma.y = \mathrm{K}_{\sigma\tau} y.$$

K, S In beiden Fällen definieren wir

$$\lambda^* x:\sigma.M = \mathrm{K}_{\sigma\tau} M.$$

app Sei $\Gamma, x:\sigma \vdash^{\mathrm{SK}} M_1 : \tau' \to \tau$ und $\Gamma, x:\sigma \vdash^{\mathrm{SK}} M_2 : \tau'$, dann ist

$$\lambda^* x:\sigma.M_1 M_2 = \mathrm{S}_{\sigma\tau'\tau}(\lambda^* x:\sigma.M_1)(\lambda^* x:\sigma.M_2)$$

Frage: Warum definieren wir λ^* nicht zunächst als Operation über Termen?

Wir können eine β-Regel für λ^* zeigen:

Lemma 3.2 *Es sei $\Gamma, x:\sigma \vdash^{\mathrm{SK}} M:\tau$ und $\Gamma \vdash^{\mathrm{SK}} N:\sigma$ dann gilt*

$$\vdash_{\mathrm{SK'}} \Gamma \rhd (\lambda^* x:\sigma.M)N = M[x:=N]:\tau$$

Beweis: Durch Induktion über die Herleitung von $\Gamma, x : \sigma \vdash^{\mathrm{SK}} M : \tau$. $\Box$

Aufgabe 3.2 *Zeige, daß die ξ-Regel nicht für λ^* gilt, d.h. finde Terme, so daß gilt*

$$\vdash_{\mathrm{SK}} \Gamma, x : \sigma \rhd M = N : \tau$$

aber

$$\not\vdash_{\mathrm{SK}} \Gamma(\lambda^* x : \sigma.M) = (\lambda^* x : \sigma.N) : \tau$$

Wir definieren die Operation $-^{\#}$, die λ-Terme in Terme der kombinatorischen Logik übersetzt: Gegeben $\Gamma \vdash M : \sigma$ erhalten wir $\Gamma \vdash^{\mathrm{SK}} M^{\#} : \sigma$. Die Definition erfolgt durch Induktion über die Herleitung von $\Gamma \vdash M : \sigma$:

var $x^{\#} = x$

app $(M_1 M_2)^{\#} = M_1^{\#} M_2^{\#}$

lam $(\lambda x : \sigma.M)^{\#} = \lambda^* x : \sigma.M^{\#}$

Lemma 3.3

$$\frac{\mathcal{E} \vdash_{\mathrm{SK}} \Gamma \rhd M = N : \sigma}{\mathcal{E}^{\#} \vdash \Gamma \rhd M^{\#} = N^{\#} : \sigma}$$

Beweis: Durch Induktion über die Herleitung. Der einzige schwierige Fall ist β und dies wird durch Lemma 3.2 abgedeckt. $\Box$

Wir haben gezeigt, daß die Übersetzungen zwischen kombinatorischer Logik und dem schwachen λ-Kalkül Typen und Gleichheit erhalten. Bilden diese Homomorphismen einen Isomorphimsus (d.h. die beiden Kalküle sind äquivilent) oder zumindestens einen Retrakt (ein Kalkül ist eine Verfeinerung des anderen)?

Aufgabe 3.3 *Versuche die folgenden Behauptungen zu etablieren oder zu widerlegen:*

1. $$\frac{\Gamma \vdash M : \sigma}{\vdash_{\mathrm{W}} (M^{\S})^{\#} = M : \sigma}$$

2. $$\frac{\Gamma \vdash^{\mathrm{SK}} M : \sigma}{\vdash_{\mathrm{SK}} (M^{\#})^{\S} = M : \sigma}$$

Falls 1. und 2. gelten sprechen wir von einem Isomorphismus, falls nur eines von beiden gilt von einem Retrakt.

3.3 Reduktionsfreie Normalisierung

In [BS91] wird ein Beweis für die Normalisierung des einfach getypten λ-Kalküls gegeben, der auf einem Inversen der Auswertung in einem Modell beruht. Dies führt zu einer effizienten Implementation insbesondere in LISP-dialekten mit `eval`. In [CD93] wird eine ähnliche Konstruktion für kombinatorische Logik mit primitiver Rekursion analysiert. Die Autoren beobachten insbesondere eine Verbindung zu einer Konstruktion aus der Kategorientheorie (dem *glueing*). In [AHS95] wird diese Idee weiterentwickelt und ein rein kategorieller Normalisierungsbeweis für den einfach getypten λ-Kalkül angegeben. Dies wurde in der Zwischenzeit für eine Kombinator-Version für System F [AHS96a] und für das volle System F [AHS96b] erweitert.

Im folgenden konstruieren wir eine Normalisierungsfunktion für geschlossene Terme der kombinatorischen Logik, wie in [CD93] beschrieben. Wir bezeichnen $\mathrm{Tm}(\sigma) = \{M \mid \vdash^{\mathrm{SK}} M : \sigma\}$ und wir werden explizit die Menge der Normalformen

$\mathrm{Nf}(\sigma) \subseteq \mathrm{Tm}(\sigma)$ definieren. Aus unserer Konstruktion erhalten wir eine Familie von Funktionen $\mathrm{nf}^\sigma \in \mathrm{Tm}(\sigma) \to \mathrm{Nf}(\sigma)$, so daß gilt

$$\frac{\vdash_{\mathrm{SK}} M = N : \sigma}{\mathrm{nf}^\sigma(M) = \mathrm{nf}^\sigma(N)} \qquad \vdash_{\mathrm{SK}} M = \mathrm{nf}^\sigma(M)$$

Daher gilt $\vdash_{\mathrm{SK}} M = N : \sigma \iff \mathrm{nf}^\sigma(M) = \mathrm{nf}^\sigma(N)$, insbesondere läßt sich zeigen, daß $\vdash_{\mathrm{SK}} - = - : \sigma$ entscheidbar ist, wenn die Funktion nf^σ berechenbar ist[3]. Dies kann dadurch sichergestellt werden, indem die gesamte Konstruktion in einer konstruktiven Mengenlehre, wie z.B. Martin Löf's Intuitionistische Typtheorie durchgeführt wird. Wir werden hier jedoch die Konstruktion in normaler (naiver) Mengenlehre durchführen und duch Inspektion der Konstruktion sehen, daß sich die Funktion nf in der Tat berechnen läßt. Insbesondere werden wir eine ML-Implementation von nf gewinnen.

Wir definieren zunächst $\mathrm{Nf}(\sigma)$ als die Teilmenge von $\mathrm{Tm}(\sigma)$, die durch die folgende Grammatik erzeugt wird:

$$\mathrm{Nf} :: \mathrm{K}_{\sigma\tau} \mid \mathrm{K}_{\sigma\tau}\mathrm{Nf} \mid \mathrm{S}_{\sigma\tau\rho} \mid \mathrm{S}_{\sigma\tau\rho}\mathrm{Nf} \mid \mathrm{S}_{\sigma\tau\rho}\mathrm{Nf}\ \mathrm{Nf}$$

Diese Wahl der Normalformen ist identisch zu der Menge der Normalformen, die wir aus dem Termersetzungsystm erhalten, das durch das Richten der Gleichungen entsteht.

Wir definieren nun eine Semantik für Typen $[\![\sigma]\!]$ zusammen mit einer Funktion $\mathrm{q}_\sigma \in [\![\sigma]\!] \to \mathrm{Nf}(\sigma)$. Dies geschieht durch Induktion über die Struktur von σ:

$\sigma = o$

$$\begin{aligned} [\![o]\!] &= \mathrm{Nf}(o) \\ \mathrm{q}_o(M) &= M \end{aligned}$$

$\sigma = \sigma_1 \to \sigma_2$

$$\begin{aligned} [\![\sigma_1 \to \sigma_2]\!] &= \left\{ \begin{array}{ll} & (M, f) \in \mathrm{Nf}(\sigma_1 \to \sigma_2) \times ([\![\sigma_1]\!] \to [\![\sigma_2]\!]) \\ \mid & \forall x \in [\![\sigma]\!].\, \vdash_{\mathrm{SK}} M\mathrm{q}_\sigma(x) = \mathrm{q}_\tau(fx) : \tau \end{array} \right\} \\ \mathrm{q}_{\sigma_1 \to \sigma_2}(M, f) &= M \end{aligned}$$

Die wesentliche Idee dieser Konstruktion wird in der Definition der Sematik des Funktionenraums sichtbar: Die Semantik ist eine Normalform und eine mengentheoretische Funktion und durch die Bedingung wird sichergestelt, daß die Normalform tatsächlich die Funktion *repräsentiert*.

Wir müssen nun zeigen, daß sich diese Semantik auf Terme erweitern läßt - wir definieren zunächst $[\![M]\!]$:

K

$$[\![\mathrm{K}_{\sigma\tau}]\!] = (\mathrm{K}_{\sigma\tau}, x \in [\![\sigma]\!] \mapsto (K_{\sigma\tau}(\mathrm{q}_\sigma x), y \in [\![\tau]\!] \mapsto x))$$

S *Übung*

app Es sei $M \in \mathrm{Tm}(\sigma \to \tau)$ und $N \in \mathrm{Tm}(\sigma)$ dann erhalten wir

$$[\![MN]\!] = f[\![M]\!] \qquad \text{wobei } (-, f) = [\![M]\!]$$

[3] Da wir $\mathrm{Nf}(\sigma)$ als Teilmenge der Terme definiert haben, gehen wir stillschwiegend davon aus, daß sich die Gleichheit auf Nf entscheiden läßt.

Lemma 3.4 (Typkorrektheit)

$$\frac{M \in \mathrm{Tm}(\sigma)}{[\![M]\!] \in [\![\sigma]\!]}$$

Beweis: Beweis durch Induktion über die Herleitung $\vdash^{\mathrm{SK}} M : \sigma$.

K Es gilt zu zeigen, daß $[\![K_{\sigma\tau}]\!] = (M, f) \in [\![\sigma \to \tau \to \sigma]\!]$. Wir müssen also verifizieren:

1.
$$M = \mathrm{K}_{\sigma\tau} \in \mathrm{Nf}(\sigma \to \tau \to \sigma)$$
Folgt aus der Definition von Nf.

2.
$$\forall x \in [\![\sigma]\!]. \vdash_{\mathrm{SK}} \mathrm{K}_{\sigma\tau}(\mathrm{q}_\sigma x) = \mathrm{q}_{\sigma\to\tau}(fx).$$
Folgt aus der Definition von $[\![\mathrm{K}_{\sigma\tau}]\!]$, denn $\mathrm{q}_{\sigma\to\tau}(fx) = \mathrm{K}_{\sigma\tau}(\mathrm{q}_\sigma x)$.

3.
$$f \in [\![\sigma]\!] \to [\![\sigma \to \tau]\!]$$
Wir nehmen an $x \in [\![\sigma]\!]$. Zu zeigen
$$(K_{\sigma\tau}(\mathrm{q}_\sigma x), y \in [\![\tau]\!] \mapsto x)) \in [\![\tau \to \sigma]\!],$$
d.h.

(a)
$$K_{\sigma\tau}(\mathrm{q}_\sigma x) \in \mathrm{Nf}(\tau \to \sigma)$$
Folgt aus der Definition von Nf.

(b)
$$\forall y \in [\![\tau]\!]. \vdash_{\mathrm{SK}} \mathrm{K}_{\sigma\tau}(\mathrm{q}_\sigma x)(\mathrm{q}_\tau y) = (\mathrm{q}_\tau y) : \tau$$
Dies folgt aus $\mathrm{K}^=$.

(c)
$$y \in [\![\tau]\!] \mapsto x \in [\![\tau]\!] \to [\![\sigma]\!]$$
Dies folgt aus der Annahme $x \in [\![\sigma]\!]$

S *Übung*

app Folgt direkt aus den Voraussetzungen.

□

Lemma 3.5 (Korrektheit für Gleichungen)

$$\frac{\vdash_{\mathrm{SK}} M = N : \sigma}{[\![M]\!] = [\![N]\!]}$$

Beweis: Durch Induktion über die Herleitung von $\vdash_{\mathrm{SK}} M = N : \sigma$. □

Aufgabe 3.4 *Ergänze den Fall für $[\![\mathrm{S}_{\sigma\tau\rho}]\!]$, und vervollständige die Beweise der Lemmas 3.4 und 3.5.*

Wir müssen jetzt nur noch zeigen, daß q das Inverse der Auswertung ist:

Lemma 3.6 *Es sei $M \in \mathrm{Tm}(\sigma)$ dann gilt*

$$\vdash_{\mathrm{SK}} \mathrm{q}_\sigma([\![M]\!]) = M : \sigma$$

Beweis: Durch Induktion über die Ableitung von $\vdash^{\mathrm{SK}} M : \sigma$. Die Fälle K und S sind jetzt einfach, also wenden wir uns **app** zu: Wir nehmen an $M_1 \in \mathrm{Tm}(\sigma_1 \to \sigma_2), M_2 \in \mathrm{Tm}(\sigma_1)$. Wir nehmen an, $[\![M_1]\!] = (-, f) \in [\![\sigma_1 \to \sigma_2]\!]$ und wir können folgendes herleiten:

$$\begin{array}{llll} \vdash_{\mathrm{SK}} & \mathrm{q}_\tau([\![M_1 M_2]\!]) & = & \mathrm{q}_\tau(f[\![M_2]\!]) \\ & & = & M_1(\mathrm{q}_\sigma[\![M_2]\!]) \quad \text{Bedingung für } [\![\sigma_1 \to \sigma_2]\!] \\ & & = & M_1 M_2 \qquad\qquad \text{Induktionsannahme für } M_2. \end{array}$$

□

Wir definieren nun

$$\mathrm{nf}^\sigma(M) = \mathrm{q}_\sigma([\![M]\!])$$

und es gilt:

Satz 3.7

$$\frac{\vdash_{\mathrm{SK}} M = N : \sigma}{\mathrm{nf}^\sigma(M) = \mathrm{nf}^\sigma(N)} \qquad \frac{}{\vdash_{\mathrm{SK}} M = \mathrm{nf}^\sigma(M)}$$

Beweis: Folgt direkt aus den Lemmas 3.5 und 3.6. □

Wir können nun aus der Konstruktion folgendes ML Programm gewinnen:

```
datatype tm = s | k | ap of tm*tm;

datatype vl = v_arr of tm*(vl->vl);

fun term_part (v_arr (M,_)) = M;
fun val_ap (v_arr (_,f),x) = f x;

fun eval s =
     v_arr (s,fn x=>
              v_arr (ap(s,term_part x),
                     fn y =>
                     v_arr (ap (ap (s,term_part x),term_part y),
                                fn z => val_ap (val_ap (x,z),
                                                val_ap (y,z)))))
  | eval k = v_arr (k,fn x => v_arr (ap(k,term_part x),
                                     fn y => x))
  | eval (ap(x,y)) = val_ap (eval x,eval y);

fun norm t = term_part (eval t);
```

Aufgabe 3.5 *Erweitere den Beweis von Satz 3.7. um natürliche Zahlen und primitive Rekursion, d.h. wir führen einen Typen* $\mathrm{N} \in \mathrm{Ty}$ *ein und folgende Konstanten:* $0, \mathrm{s}, \mathrm{R}^\sigma \in \mathrm{Tm}^{\mathrm{SK}}$ *mit den Typisierungen:*

$$\vdash^{\mathrm{SK}} 0 : \mathrm{N}$$
$$\vdash^{\mathrm{SK}} \mathrm{s} : \mathrm{N} \to \mathrm{N}$$
$$\vdash^{\mathrm{SK}} \mathrm{R}^\sigma : \sigma \to (\mathrm{N} \to \sigma \to \sigma) \to \mathrm{N} \to \sigma$$

und den Gleichungen:

$$\begin{array}{llcll} \vdash_{\mathrm{SK}} & \mathrm{R}^\sigma M_1 M_2 0 & = & M_1 & : \sigma \\ \vdash_{\mathrm{SK}} & \mathrm{R}^\sigma M_1 M_2 (\mathrm{s}\ N) & = & M_2 N(\ R^\sigma M_1 M_2 N) & : \sigma \end{array}$$

Erweitere die Semantik, zeige die neuen Fälle und leite daraus Entscheidbarkeit der Gleichheit und Injektivität für s*:*

$$\frac{\vdash_{\mathrm{SK}} \mathrm{s}\ N = \mathrm{s}\ N' : \mathrm{N}}{\vdash_{\mathrm{SK}} N = N' : \mathrm{N}}$$

Hinweis: *Die Konstruktion wird in [CD93] beschrieben; es ist jedoch lehrreich es alleine zu versuchen.*

Literatur

[AHS95] Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. Categorical reconstruction of a reduction free normalization proof. In David Pitt, David E. Rydeheard, and Peter Johnstone, editors, *Category Theory and Computer Science*, LNCS 953, pages 182–199, 1995.

[AHS96a] Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. Reduction-free normalisation for a polymorphic system. In *11th Annual IEEE Symposium on Logic in Computer Science*, pages 98–106, 1996.

[AHS96b] Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. Reduction-free normalization for system f. submitted for publication, 1996.

[BS91] Ulrich Berger and Helmut Schwichtenberg. An inverse of the evaluation functional for typed λ-calculus. In *Proceedings of the Sixth Annual Symposium on Logic in Computer Science*, pages 202 – 211, 1991.

[CD93] Thierry Coquand and Peter Dybjer. Normalization algorithms, mechanical normalization and intuitionistic model constructions, 1993. To appear in: Mathematical Structures in Computer Science.

[Sel96] Peter Selinger. The lambda calculus is algebraic. Expository, December 1996.