Permutative Konversionen und Schnitt-Typen

Ralph Matthes[a], LMU München

Arbeitstagung Bern-München
am 6. und 7. April 2000
am Institut für Informatik und Angewandte Mathematik
der Universität Bern

[a] `matthes@informatik.uni-muenchen.de`,
Homepage: `http://www.tcs.informatik.uni-muenchen.de/~matthes/`

Der Abstract des zugehörigen Artikels:

An intersection type assignment system for the extension ΛJ of the untyped λ-calculus, introduced by Joachimski and Matthes, is given and proven to characterize the strongly normalizing terms of ΛJ. Since ΛJ's generalized applications naturally allow permutative/commuting conversions, this the first analysis of a term rewrite system with permutative conversions by help of intersection types. Two proofs are given for the fact that the typable terms are strongly normalizing: One by the computability predicates method à la Tait and one showing directly that strongly normalizing typable terms are closed under (generalized) application and substitution. It is also shown that a straightforward extension of the type assignment for lambda calculus fails to capture the strongly normalizing terms.

Inhalt:

- Vorstellung von ΛJ, dem Lambda-Kalkül mit verallgemeinerter Anwendungsregel
- Typisierung mittels Schnitt-Typen
- Alle stark normalisierenden Terme werden erfaßt.
- Gilt nicht für ein naiveres Typsystem.
- Die typisierbaren Terme sind stark normalisierend.
  - Die Methode à la Tait
  - Die Methode à la Joachimski/Matthes

Der Kalkül ΛJ

Ausgangspunkt: Gentzens Einführungsregel

$$\frac{\Gamma \vdash A \qquad \Gamma, B \vdash C}{\Gamma, A \to B \vdash C}\,(L\!\to)$$

Dazu einen Termkalkül mit Typisierung (für schnittfreie Herleitungen):

$$\frac{\Gamma \vdash s : A \qquad \Gamma, z : B \vdash t : C}{\Gamma, x : A \to B \vdash x(s, z.t) : C}\,(L\!\to)$$

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma, x : A \vdash r : B}{\Gamma \vdash \lambda x r : A \to B}$$

Entsprechend im natürlichen Schließen:

$$\frac{}{x^A} \qquad \frac{s^A \qquad t^C}{(x^{A\to B}(s, z^B.t))^C}\,(L\to) \qquad \frac{r^B}{(\lambda x^A r)^{A\to B}}$$

Idee der verallgemeinerten $\to$-Eliminationen bei von Plato (dort für Herleitungsbäume ausgesprochen):
Statt x ein beliebiger Term $r^{A\to B}$ in der $\to$-Beseitigungsregel, also als Beseitigungsregel

$$\frac{r^{A\to B} \qquad s^A \qquad t^C}{(r(s, z^B.t))^C}$$

Mithin wird der Gentzen-Kalkül LJ in seiner schnittfreien Fassung gegen Substitution abgeschlossen.

Idee von J/M: Betrachte gleich den ungetypten Kalkül: x, λxr, $r(s, z.t)$. Das ergibt den Abschluß des rechnerischen (d.h. ungetypten) Gehalts des schnittfreien LJ unter Substitution—ΛJ.

Notation: S für Gebilde der Form $(s, z.t)$.

Terme der Form $r(s, z.t)$ mit $r \neq x$ sind nicht normal! (schon im Gentzen-Kalkül nicht schnittfrei) Sie sind reduzierbar via:

(β) $(\lambda x r)(s, z.t) \mapsto t[z := r[x := s]]$

(π) $r(s, z.t)S \mapsto r(s, z.tS)$ mit z nicht frei in S

Diese Transformationen erhalten logische Korrektheit (im Sinne des natürlichen Schließens).

$rs := r(s, z.z)$ ergibt eine Einbettung des ungetypten Lambda-Kalküls.

Umgekehrt sind (β) und (π) korrekt bzgl. $r(s, z.t) := t[z := rs]$.

Das motiviert noch (κ) $r(s, z.t) \mapsto t$ für z nicht frei in t.

Induktive Charakterisierung der stark normalisierenden Terme von ΛJ (J/M 1999):

$$\frac{}{x \in \mathsf{SN}} \qquad \frac{s, t \in \mathsf{SN}}{x(s, z.t) \in \mathsf{SN}}\,(\mathrm{V})$$

$$\frac{r \in \mathsf{SN}}{\lambda x r \in \mathsf{SN}}\,(\lambda) \qquad \frac{x(s, z.tS)\vec{S} \in \mathsf{SN}}{x(s, z.t)S\vec{S} \in \mathsf{SN}}\,(\pi)$$

$$\frac{t[z := r[x := s]]\vec{S} \in \mathsf{SN} \qquad r, s \in \mathsf{SN}}{(\lambda x r)(s, z.t)\vec{S} \in \mathsf{SN}}\,(\beta)$$

$\vec{S}$ bedeutet endlich viele verallgemeinerte Anwendungen.

Das Resultat ist davon unabhängig, ob κ-Reduktion eingeschlossen wird.

Typisierung mit Schnitt-Typen

$$\frac{}{\Gamma, x : \rho \vdash x : \rho}\,(\mathrm{V}) \qquad \frac{\Gamma, x : \rho \vdash r : \sigma}{\Gamma \vdash \lambda x r : \rho \to \sigma}\,(\lambda)$$

$$\frac{\Gamma \vdash r : \rho_1 \qquad \Gamma \vdash r : \rho_2}{\Gamma \vdash r : \rho_1 \cap \rho_2}\,(\cap\text{-I}) \qquad \frac{\Gamma \vdash r : \rho_1 \cap \rho_2 \qquad i \in \{1, 2\}}{\Gamma \vdash r : \rho_i}\,(\cap\text{-E})$$

$$\frac{\Gamma \vdash r : \rho \qquad \Gamma \vdash s : \rho_1 \qquad \Gamma, z : \rho_2 \vdash t : \tau \qquad z \in \mathrm{FV}(t) \Rightarrow \rho = \rho_1 \to \rho_2}{\Gamma \vdash r(s, z.t) : \tau}$$

Die letzte Regel (app) ist äquivalent zu zwei Regeln:

$$\frac{\Gamma \vdash r : \rho_1 \to \rho_2 \qquad \Gamma \vdash s : \rho_1 \qquad \Gamma, z : \rho_2 \vdash t : \tau}{\Gamma \vdash r(s, z.t) : \tau}\,(\mathrm{app}_1)$$

Die erste Regel ist wie erwartet. Neu ist die zweite Regel:

$$\frac{\Gamma \vdash r : \rho \qquad \Gamma \vdash s : \sigma \qquad \Gamma \vdash t : \tau \qquad z \notin \mathrm{FV}(t)}{\Gamma \vdash r(s, z.t) : \tau}\,(\mathrm{app}_2)$$

Ohne sie werden nicht alle stark normalisierenden Terme erfaßt (siehe später).

Eigenschaften: Abschwächung, Verstärkung, Vertauschung und Substitution. Letzteres wie folgt:

(a) Wenn $\Gamma, x : \rho \vdash r : \tau$ und $\Gamma \vdash s : \rho$, dann $\Gamma \vdash r[x := s] : \tau$.

(b) Wenn $\Gamma \vdash r[x := s] : \tau$, $x \notin \mathrm{FV}(s) \cup \Gamma$ und $(x \notin \mathrm{FV}(r) \Rightarrow \Gamma \vdash s : \rho_0)$, dann gibt es einen Typ ρ mit $\Gamma, x : \rho \vdash r : \tau$ und $\Gamma \vdash s : \rho$.

Bem.: (a) wird man immer erwarten, (b) ist typisch für Systeme mit Schnitt-Typen.

Primtypen: Typen nicht von der Form $\rho \cap \sigma$. Gesamtheit sei $\mathcal{PT}$. Jeder Typ hat die Form $\rho_1 \cap \ldots \cap \rho_n$ mit Primtypen $\rho_1, \ldots, \rho_n$ (Primfaktoren).

Damit gilt folgendes **Inversionslemma** (analog Krivine):
Sei $\Gamma \vdash r : \rho$ mit $\rho \in \mathcal{PT}$.

1. Falls $r = x$, dann $x : \cap\vec{\rho} \in \Gamma$ mit $\rho \in \vec{\rho}$.
2. Falls $r = \lambda xt$, dann $\rho = \rho_1 \to \rho_2$ und $\Gamma, x : \rho_1 \vdash t : \rho_2$.
3. Falls $r = r'(s, z.t)$, dann $\Gamma \vdash r' : \sigma$, $\Gamma \vdash s : \sigma_1$, $\Gamma, z : \sigma_2 \vdash t : \rho$ und $(z \in \mathrm{FV}(t) \Rightarrow \sigma = \sigma_1 \to \sigma_2)$.

Damit zeigt man **Subject Reduction**: Falls $\Gamma \vdash r : \rho$ und $r \to r'$, dann $\Gamma \vdash r' : \rho$. (Beweis erfordert Analyse aller Redukte. κ-Reduktion funktioniert problemlos. Permutation erfordert keine Unterscheidung vierer Fälle, da die Regel (app) so kompakt geschrieben ist.)

Vollständigkeit der Typisierung Gezeigt wird, daß die typisierbaren Terme alle definierenden Abschlußeigenschaften von SN haben. Damit ist jeder Term in SN typisierbar.

Zuerst beweist man:

- Falls $\Gamma \vdash t[z := r[x := s]] : \tau$, $\Gamma, x : \sigma' \vdash r : \rho$ und $\Gamma \vdash s : \sigma$, dann $\Gamma \vdash (\lambda xr)(s, z.t) : \tau$.
- Falls $\Gamma \vdash r(s, z.tS) : \tau$, dann $\Gamma \vdash r(s, z.t)S : \tau$.

Dann geht der Beweis recht problemlos mit viel Abschwächung und Substitution (mit (b) in meiner starken Fassung).

Unvollständigkeit ohne (app_2)

$t := (\lambda x.x(x, z.z))(\lambda x.x(x, z.z), y.z) \in \mathsf{SN}$ ist nicht typisierbar im eingeschränkten System.

Das geht mit einem geänderten Inversionslemma, dessen 3. Fall lautet: Falls $r = r'(s, z.t)$, dann $\Gamma \vdash r' : \sigma_1 \to \sigma_2$, $\Gamma \vdash s : \sigma_1$ und $\Gamma, z : \sigma_2 \vdash t : \rho$.

Wäre $\Gamma \vdash t : \tau$, dann ohne Einschränkung $\tau \in \mathcal{PT}$. Also $\Gamma \vdash \lambda x.x(x, z.z) : \rho \to \sigma$ und $\Gamma \vdash \lambda x.x(x, z.z) : \rho$. Trivialerweise $\Gamma, y : \sigma \vdash y : \sigma$. Also $\Gamma \vdash (\lambda x.x(x, z.z))(\lambda x.x(x, z.z), y.y) : \sigma$.

Wegen Korrektheit (erst recht der eingeschränkten Regel) folgt $(\lambda x.x(x, z.z))(\lambda x.x(x, z.z), y.y) \in \mathsf{SN}$, was offenbar falsch ist: Der Term ist die ΛJ-Version von $(\lambda x.xx)(\lambda x.xx)$.

Korrektheit wurde noch nicht gezeigt.

Beweis ohne Korrektheit braucht folgende Aussage: Sei $\vec{\rho} \subset \mathcal{PT}$. Unmöglich sind:

$$\Gamma \vdash \lambda x.x(x, z.z) : \cap\vec{\rho} \to \sigma \text{ und } \Gamma \vdash \lambda x.x(x, z.z) : \cap\vec{\rho}.$$

Beweis durch Induktion über die Anzahl der verschiedenen Typen in $\vec{\rho}$. (Dies geht natürlich sogar für das volle Typisierungssystem.)

Korrektheit Jeder typisierbare Term liegt in SN (ist also stark normalisierend).

Zwei Beweise: Tait-Methode, Methode von Joachimski/Matthes.

Die Tait-Methode [1975] klappt problemlos.

Etwas präziser: Man definiert saturierte Mengen von Termen wie folgt: Eine Menge $\mathcal{M}$ ist saturiert, wenn

(SN) $\mathcal{M} \subseteq \mathsf{SN}$.

(V) $x \in \mathcal{M}$, und falls $s \in \mathsf{SN}$ und $\boxed{t \in \mathcal{M}}$, dann $x(s, z.t) \in \mathcal{M}$.

(π) Falls $x(s, z.tS)\vec{S} \in \mathcal{M}$, dann $x(s, z.t)S\vec{S} \in \mathcal{M}$.

(β) Falls $t[z := r[x := s]]\vec{S} \in \mathcal{M}$ und $r, s \in \mathsf{SN}$, dann $(\lambda xr)(s, z.t)\vec{S} \in \mathcal{M}$.

Bem.: $t \in \mathcal{M}$ ist essentiell, um permutative Konversionen behandeln zu können.

Die Regel (app_2) wird behandelt mit:
Sei $\mathcal{M}$ saturiert. Dann gilt folgende abgeleitete κ-Expansions-Regel:
Wenn $r, s \in \mathsf{SN}$, $t \in \mathcal{M}$ und $z \notin \mathrm{FV}(t)$, dann $r(s, z.t) \in \mathcal{M}$.

Es geht ohne logische Relationen/Berechenbarkeitsprädikate.

[Die Methode von Joachimski/Matthes—nun auch für Schnitt-Typen.]

Wir brauchen die abgeleitete π-Expansions-Regel (J/M 1999):
Falls $r(s, z.tS)\vec{S} \in \mathsf{SN}$, dann $r(s, z.t)S\vec{S} \in \mathsf{SN}$.

Lemma. *Für alle Typen ρ und alle Terme $r \in \mathsf{SN}$ gilt:*

(i) *Falls $\Gamma \vdash r : \rho$, $\Gamma \vdash s : \rho_1$, $\Gamma, z : \rho_2 \vdash t : \tau$ und $s, t \in \mathsf{SN}$ und $(z \in \mathrm{FV}(t) \Rightarrow \rho = \rho_1 \to \rho_2)$, dann $r(s, z.t) \in \mathsf{SN}$.*

(ii) *Falls $\Gamma, x : \rho \vdash r : \tau$, $\Gamma \vdash s : \rho$ und $s \in \mathsf{SN}$, dann $r[x := s] \in \mathsf{SN}$.*

Beweis: Simultan durch Induktion über ρ und Nebeninduktion über $r \in \mathsf{SN}$. Jeweils ist (i) vor (ii) zu zeigen. Die Fallunterscheidung wird nach der Ableitung von $r \in \mathsf{SN}$ vorgenommen.

Wir brauchen intensiv Subject Reduction und die Substitutions-Aussagen sowie ständig Abschwächung und Inversion. Die Haupt-Induktionsvoraussetzung wird nur in den Fällen (ii)(V) (hier die von (i)) und (i)(λ) (hier die von (ii)) gebraucht. (ii) greift auf (i) zurück.

Bem.: Korrektheit folgt sofort aus dem Lemma, Teil (i).

Fragen:

- Geht das auch mit strikten Schnitt-Typen? Ich habe ein System und einen Vollständigkeitsbeweis, aber keinen der starken Normalisierung (weil auch Taits Methode nicht funktioniert).
- Geht der zweite Korrektheitsbeweis durch für ungetyptes Gödelsches T mit Schnitt-Typen (also Rekursion anstatt permutativer Reduktionen)? Dabei wäre spannend, wie mit der (infinitären) ω-Regel im Kontext von Schnitt-Typen umzugehen ist.

Finis