Übungsaufgaben zur Vorlesung
Mathematische Grundlagen der Computerlinguistik II

Universität München, CIS, SS 2007

H.Leiß

Abgabetermin: Di, 3.7.2007

Aufgabe 17 (Gentzen-Kalkül für die Aussagenlogik)

(a) Konstruiere mit den Regeln des Gentzen-Kalküls für die Aussagenlogik die Sequenz $\Gamma \rhd \varphi$ für die Annahmen $\Gamma = \{A, \neg B\}$ und die Aussage

$$\varphi := (A \to B) \to \neg C$$

(b) Versuche, für die leere Annahmemenge Γ und die Aussage

$$\varphi := (A \lor C) \to (B \lor \neg A \lor C)$$

die Sequenz $\Gamma \rhd \varphi$ mit den Regeln des Gentzen-Kalküls zu konstruieren (vom Ziel $\Gamma \rhd \varphi$ rückwärts!). Dabei tritt eine Sequenz auf, die weder Axiom noch Untersequenz einer Regel ist. Konstruiere aus ihr eine Belegung $g : \mathit{Var} \to \mathbb{B}$ mit $[\![\varphi]\!]_g = 0$.

Bemerkung: Normalerweise definiert man die Lambda-Terme durch

$$t, s \;:=\; x \mid c \mid (t \cdot s) \mid \lambda x t \qquad \text{(Variable, Konstante, Anwendung, Abstaktion)}$$

mit einem 2-stelligen Funktionszeichen $\cdot$ für die „Anwendung". Dann ist in z.B. $\lambda x(t \cdot s)$ die Klammerung klar und die Abstraktion λx wirkt auf den unmittelbar folgenden Term, hier also $(t \cdot s)$. Mit $t \cdot s \cdot r$ ist $(t \cdot s) \cdot r$ gemeint. Wenn man für die Anwendung von t auf s die Schreibweise $t(s)$ benutzt, wie Dowty/Wall/Peters, schreibt man $(\lambda x.t(s)(r))$ statt $\lambda x((t \cdot s) \cdot r)$, um es von $(\lambda x.t(s))(r)$ und $(\lambda x.t)(s)(r)$ unterscheiden zu können. Ich benutze hier das klarere $(t \cdot s)$.

Die *Ersetzung (der freien Vorkommen) von x in t durch s* wird über den Aufbau von t definert:

$$\begin{aligned}
y[x/s] &:= \begin{cases} s, & \text{falls } y \equiv x, \text{ (d.h. beides sind dieselbe Variable)} \\ y, & \text{falls } y \not\equiv x \end{cases} \\
c[x/s] &:= c \\
(r \cdot t)[x/s] &:= (r[x/s] \cdot t[x/s]) \\
\lambda y\, t\,[x/s] &:= \begin{cases} \lambda y\, t, & \text{falls } y \equiv x, \\ \lambda y\,(t[x/s]), & \text{sonst, falls } y \notin \mathit{frei}(s), \\ \lambda z\,(t[y/z][x/s]), & \text{sonst, mit } z \notin \mathit{frei}(\lambda y\, t \cdot s) \end{cases}
\end{aligned}$$

Aufgabe 18 (a) Was ist jeweils die Menge der freien Variablen der folgenden Lambda-Terme (ohne Typen)? Zum Einüben sind sie in beiden Schreibweisen angegeben, ohne Außenklammern:

(i) $\lambda x.y(x)(x)$ bzw. $\lambda x((y \cdot x) \cdot x)$

(ii) $x(\lambda y.y(x))$ bzw. $x \cdot \lambda y(y \cdot x)$

(iii) $x(\lambda x.y(x))$ bzw. $x \cdot \lambda x(y \cdot x)$

(iv) $y(\lambda x.z(x)(x))$ bzw. $y \cdot \lambda x((z \cdot x) \cdot x)$

(v) $y((\lambda x.z(x))(x))$ bzw. $y \cdot (\lambda x(z \cdot x) \cdot x)$

(vi) $(\lambda y.x(y))((\lambda x.y(x))(y))$ bzw. $\lambda y(x \cdot y) \cdot (\lambda x(y \cdot x) \cdot y)$

Hierbei seien x, y, z paarweise verschiedene Variable (ohne Typen).

(b) Für welche der Terme aus (i) können Sie (aus dem Grundtyp e mit $\rightarrow$ aufgebaute) Typen für die Variablen angeben, so daß es getypte λ-Terme werden, und was ist dann der Typ des gesamten Terms?

(Alle durch dasselbe λv gebundenen Vorkommen von v müssen denselben Typ haben.)

Aufgabe 19 Vereinfache die folgenden Lambdaterme (mit Konstanten f, c, d passenden Typs), bis es keine weitere Vereinfachung (außer α-Reduktionen = Umbenennungen gebundener Variablen) mehr gibt (wobei x^e, y^e, z^e paarweise verschiedene Variable seien):

(a) $(\lambda x^e.(\lambda y^e.f(y)(x)))(z^e)$

(b) $(\lambda x^e.(\lambda y^e.f(y)(x)))(x^e)$

(c) $(\lambda x^e.(\lambda y^e.f(y)(x)))(y^e)$

(d) $(\lambda x^{e\rightarrow t}.f(x(c)))(\lambda y^e.d(y))$

(e) $(\lambda x^\sigma \lambda y^{\sigma\rightarrow(\sigma\rightarrow\tau)}.y(x)(x))(\lambda z^\tau.f(z))$ für $\sigma = (\tau \rightarrow \varrho)$

(f) $(\lambda x^\sigma \lambda y^{\sigma\rightarrow(\sigma\rightarrow\tau)}.y(x)(x))(\lambda z^\tau.f(z))(\lambda x^\sigma.(\lambda z^\sigma.c(x)))$ für $\sigma = (\tau \rightarrow \varrho)$