Arch. math. Logik 17 (1976), 113—114

# DEFINIERBARE FUNKTIONEN IM $\lambda$-KALKÜL MIT TYPEN*

Von Helmut Schwichtenberg

Bekanntlich sind im typenfreien $\lambda$-Kalkül genau die rekursiven Funktionen definierbar[1]. Der dabei verwendete Begriff der Definierbarkeit einer Funktion ist auch für den $\lambda$-Kalkül mit Typen sinnvoll. Wir beantworten hier die Frage[2] nach den dann definierbaren Funktionen.

Typen sind 0 und mit $\sigma$, $\tau$ auch $(\sigma \to \tau)$. Terme (bezeichnet mit $r^\tau, s^\tau, t^\tau$) werden aus Variablen mit Typen durch Anwendung und $\lambda$-Abstraktion gebildet. Typenindizes, die sich aus dem Zusammenhang ergeben oder die unwesentlich sind, lassen wir häufig weg. Terme, die sich nur durch gebundene Umbenennung unterscheiden, werden identifiziert. Die Relation $t =| t'$ ($t$ ist reduzierbar auf $t'$) wird induktiv definiert durch

(i) $(\lambda x t)s =| t_x[s]$.
(ii) Wenn $t =| t'$ und $s =| s'$, so $ts =| t's'$.
(iii) Wenn $t =| t'$, so $\lambda x t =| \lambda x t'$.
(iv) $t =| t$.
(v) Wenn $t =| t'$ und $t' =| t''$, so $t =| t''$.

Ein Term heißt in Normalform, wenn er keine Teilterme der Gestalt $(\lambda x t)s$ besitzt. Bekanntlich gibt es zu jedem Term $t$ eine eindeutig bestimmte Normalform $t'$ mit $t =| t'$. Zwei Terme heißen gleich, wenn sie dieselbe Normalform besitzen. Man kann jetzt natürliche Zahlen einführen als Terme $\bar{n} :\equiv \lambda \alpha \cdot \alpha^n$ vom Typ $v := (0 \to 0) \to (0 \to 0)$; dabei ist $\alpha^n$ die $n$-te Iterierte von $\alpha$, also $\equiv \lambda x \cdot \alpha(\alpha \ldots (\alpha x) \ldots)$. Jeder abgeschlossene Term vom Typ $v$ in Normalform ist ein $\bar{n}$. Also definiert jeder abgeschlossene Term $t$ vom Typ $v \to (v \to \ldots (v \to v) \ldots)$ eine zahlentheoretische Funktion $f$, die durch $t\bar{n}_1 \ldots \bar{n}_k = \overline{f(n_1, \ldots, n_k)}$ bestimmt ist. Zum Beispiel wird[3]

$n + m$ definiert durch $\lambda F G \alpha \cdot (F\alpha) \circ (G\alpha)$

$n \cdot m$ definiert durch $\lambda F G \cdot F \circ G$

$k$ (konstante Funktion) definiert durch $\lambda F \alpha \cdot \alpha^k$

$d(n, m, i) = \begin{cases} n & \text{falls } i = 0 \\ m & \text{falls } i \neq 0 \end{cases}$ definiert durch $\lambda F G H \alpha x \cdot H(\lambda y \cdot G \alpha x)(F \alpha x)$

[$F, G, H$ Variable vom Typ $v$, $t^{\tau \to \tau} \circ s^{\tau \to \tau} :\equiv \lambda x^\tau \cdot t(sx)$]. Die Menge der so definierbaren Funktionen ist offenbar abgeschlossen gegen Einsetzungen. Also sind alle

---

* Eingegangen am 15. 11. 1973.

[1] Siehe etwa [2, § 31]. [2] Siehe [4, § 5], [3, § 8.9], [1, § 13D4]. [3] Nach [4, § 5].

Polynome definierbar, und allgemeiner alle Funktionen, die aus Polynomen durch Fallunterscheidung nach Verschwinden oder Nicht-Verschwinden von Argumenten definierbar sind; im 2-stelligen Fall sind dies die Funktionen

$$f(n,m)=\begin{cases} k & \text{falls } n=0 \text{ und } m=0 \\ P_1(m) & \quad = \qquad\quad \neq \\ P_2(n) & \quad \neq \qquad\quad = \\ P_3(n,m) & \quad \neq \qquad\quad \neq \end{cases}$$

mit Polynomen $P_1$, $P_2$, $P_3$. Im folgenden soll gezeigt werden, daß dies schon alle definierbaren Funktionen sind.

$t$ definiere also eine etwa 2-stellige Funktion. Man bringe $tFG\alpha$ auf Normalform. Jeder Teilterm (der Normalform von $tFG\alpha$) hat einen der Typen 0, $0\to 0$ oder $v$. Jeder Teilterm vom Typ $v$ ist identisch mit $F$ oder $G$. Mögliche Teilterme vom Typ $0\to 0$ sind:

(1) $\alpha$.

(2) Mit $s$ auch $Fs$, $Gs$.

(3) Mit $s_1, \ldots, s_q$ gebildet nach (1), (2) auch $\lambda y \cdot s_1(\ldots s_q(z)\ldots)$, wobei $z$ auch identisch mit $y$ sein kann.

$s$ steht im folgenden für Teilterme vom Typ $0\to 0$. – Für $F$, $G$ setze man $\bar{n}, \bar{m}$, wobei zunächst $n, m \geqq 1$ angenommen sei. Durch Induktion über $s$ zeigt man: Jedes $s' :\equiv s_{F,G}[\bar{n}, \bar{m}]$ ist gleich einem $\alpha^{P(n,m)}$ oder gleich einer konstanten Funktion $\lambda y \cdot \alpha^{P(n,m)} z$ ($P(n,m)$ Polynom). Beweis: Zu (2).

$$(\lambda\beta\beta^n)\alpha^{P(n,m)} = \alpha^{P(n,m)\cdot n}$$

$$(\lambda\beta\beta^n)(\lambda y \cdot \alpha^{P(n,m)} z) = \lambda y \cdot \alpha^{P(n,m)} z \qquad (\text{da } n \geqq 1).$$

Zu (3). Fall 1: Keines der $s_i'$ ist konstant.

$$\lambda y \cdot s_1'(\ldots s_q'(z)\ldots) = \lambda y \cdot \alpha^{P_1(n,m)+\cdots+P_q(n,m)} z.$$

Fall 2: Es gibt ein erstes konstantes $s_i'$, etwa $s_i' = \lambda\tilde{y} \cdot \alpha^{P_i(n,m)} \tilde{z}$.

$$\lambda y \cdot s_1'(\ldots s_q'(z)\ldots) = \lambda y \cdot \alpha^{P_1(n,m)+\cdots+P_i(n,m)} z.$$

Da die Normalform von $tFG\alpha$ keine freie Variable vom Typ 0 enthält, ist sie nach Ersetzung von $F$, $G$ durch $\bar{n}, \bar{m}$ ($n, m \geqq 1$) gleich einem $\alpha^{P(n,m)}$.

Ist etwa $n=0$, $m \geqq 1$, so kann man alle äußersten Teilterme der Form $Fs$ ersetzen durch $\lambda xx$ und erhält wie eben ein $\alpha^{P(m)}$.

## LITERATUR


[1] Curry, H. B., Hindley, J. R., Seldin, J. P.: Combinatory logic, Bd. II. Amsterdam 1972.

[2] Hermes, H.: Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit. Berlin 1961.

[3] Hindley, J. R., Lercher, B., Seldin, J. P.: Introduction to combinatory logic. Cambridge 1972.

[4] Schütte, K.: Theorie der Funktionale endlicher Typen. Hektographiert, München 1968.