Arch. math. Logik 12 (1969), 85—97.

REKURSIONSZAHLEN UND DIE GRZEGORCZYK-HIERARCHIE*

Von HELMUT SCHWICHTENBERG in Münster/Westfalen

Grzegorczyk untersucht in [2] eine aufsteigende Folge („Hierarchie") $\mathfrak{E}_0, \mathfrak{E}_1, \mathfrak{E}_2, \ldots$ von Klassen primitiv-rekursiver Funktionen. Diese Klassen $\mathfrak{E}_n$ sind mit Hilfe einer (etwas willkürlich) vorgegebenen Folge von immer stärker wachsenden primitiv-rekursiven Funktionen f_n definiert, und zwar auf folgende Weise: $\mathfrak{E}_n$ ist die kleinste Klasse, die f_n und die üblichen Ausgangsfunktionen enthält und abgeschlossen ist gegen Einsetzungen und „beschränkte Rekursionen" (das sind solche primitiven Rekursionen, bei denen die definierte Funktion durch eine schon vorher erzeugte Funktion beschränkt ist). Grzegorczyk beweist dann unter anderem:

1) $\mathfrak{E}_n \subset \mathfrak{E}_{n+1}$ für $n \geqq 0$ [1].
2) $\bigcup_n \mathfrak{E}_n$ ist die Klasse $\mathfrak{P}$ der primitiv-rekursiven Funktionen.
3) $\mathfrak{E}_3$ ist die Klasse $\mathfrak{E}$ der elementaren Funktionen.
4) Für $n \geqq 0$ gibt es in $\mathfrak{E}_{n+1}$ eine Funktion, die alle einstelligen Funktionen aus $\mathfrak{E}_n$ schließlich majorisiert.
5) Für $n \geqq 3$ gibt es in $\mathfrak{E}_{n+1}$ eine „universelle Funktion" für die einstelligen Funktionen aus $\mathfrak{E}_n$.

Ein anderer, natürlicher erscheinender Ansatz zur Klassifikation der primitiv-rekursiven Funktionen wird von Heinermann in [3] und von Axt in [1] untersucht. Die von ihnen eingeführten Klassen $\mathfrak{R}_n$ (bzw. K_n) sollen genau die primitiv-rekursiven Funktionen enthalten, die sich aus den Ausgangsfunktionen durch Einsetzungen und primitive Rekursionen so definieren lassen, daß dabei höchstens n primitive Rekursionen „übereinander liegen". Wir werden hier zeigen, daß für $n \geqq 3$ $\mathfrak{R}_n = \mathfrak{E}_{n+1}$ ist[2]. Dies ist eine leichte Verbesserung von Ergebnissen von Meyer (angekündigt in [4]) und D. M. Ritchie (angekündigt in [5]), die diese Übereinstimmung für $n \geqq 6$ bzw. für $n \geqq 4$ erhalten haben. Ein Beweis für $n \geqq 5$ wird von Rödding in [7] gegeben. Außerdem betrachten wir noch Klassen $\mathfrak{R}_n^{\mathrm{sim}}$, die mit simultanen primitiven Rekursionen anstelle von primitiven Rekursionen wie die $\mathfrak{R}_n$ definiert sind. Für diese Klassen ergibt sich $\mathfrak{R}_n^{\mathrm{sim}} = \mathfrak{E}_{n+1}$ schon für $n \geqq 2$; insbesondere ist also $\mathfrak{R}_2^{\mathrm{sim}} = \mathfrak{E}$ [3].

* Eingegangen am 8. 11. 1967.

[1] Mit „$\subset$" bezeichnen wir die echte Inklusion.

[2] Dem Verfasser ist nicht bekannt, ob auch $\mathfrak{R}_2 = \mathfrak{E}_3$ ist. $\mathfrak{R}_0$ und $\mathfrak{R}_1$ bestehen dagegen aus einfach gebauten, leicht zu beschreibenden Funktionen; vgl. Heinermann [3].

[3] $\mathfrak{R}_0^{\mathrm{sim}}$ und $\mathfrak{R}_1^{\mathrm{sim}}$ lassen sich noch direkt übersehen; z. B. liegt die $\mathfrak{E}_0$-Funktion $\lambda x[\sqrt{x}]$ nicht in $\mathfrak{R}_1^{\mathrm{sim}}$. Wir gehen auf diese Klassen nicht näher ein.

1. Die Grzegorczyk-Hierarchie. In [6] zeigt R. W. Ritchie, daß man die Grzegorczyk-Klassen $\mathfrak{E}_n$ auch erhält, wenn man von den folgenden „Ackermannschen" Funktionen A_n ausgeht:

$$A_0(x, y) = x + 1$$

$$A_{n+1}(x, 0) = \begin{cases} x & \text{für} \quad n = 0 \\ 0 & \text{für} \quad n = 1 \\ 1 & \text{sonst} \end{cases}$$

$$A_{n+1}(x, y + 1) = A_n(x, A_{n+1}(x, y))$$

Es ist dann $A_1(x, y) = x + y$, $A_2(x, y) = x \cdot y$, $A_3(x, y) = x^y$. Diese Ritchiesche Definition der Grzegorczyk-Klassen wollen wir hier zugrunde legen. Also

Definition: $\mathfrak{E}_n (n \geqq 0)$ sei die kleinste Klasse zahlentheoretischer Funktionen, die

1) die Ausgangsfunktionen $\lambda x\, x + 1$, $\lambda x_1 \ldots x_m x_i (1 \leqq i \leqq m)$, $\lambda x_1 \ldots x_m\, a$ $(m \geqq 0, a \geqq 0)$ und A_n enthält,
2) abgeschlossen ist gegen Einsetzungen, also mit g (r-stellig, $r \geqq 1$) und $h_1, \ldots, h_r$ (m-stellig, $m \geqq 1$) stets auch die durch
$$f(\mathfrak{x}) = g(h_1(\mathfrak{x}), \ldots, h_r(\mathfrak{x}))$$
definierte Funktion f enthält, und
3) abgeschlossen ist gegen beschränkte Rekursionen, also mit g (m-stellig, $m \geqq 0$), h ($m + 2$-stellig) und j ($m + 1$-stellig) stets auch die durch
$$f(\mathfrak{x}, 0) = g(\mathfrak{x})$$
$$f(\mathfrak{x}, y + 1) = h(\mathfrak{x}, y, f(\mathfrak{x}, y))$$
definierte Funktion f enthält, falls sie durch j beschränkt ist, d. h. falls gilt
$$f(\mathfrak{x}, y) \leqq j(\mathfrak{x}, y) .$$

Es sollen jetzt aus dieser Definition einige später benötigte Eigenschaften der Klassen $\mathfrak{E}_n$ abgeleitet werden. Dabei ist es nützlich, genauere Kenntnisse über die Funktionen A_n zu haben:

Lemma:

(1) $A_n(x, 1) = x$ für $n \geqq 2, x \geqq 0$.
(2) $A_n(1, y) = 1$ für $n \geqq 3, y \geqq 0$.
(3) $A_n(x, y) > y$ für $n \geqq 3, x \geqq 2, y \geqq 0$.
(4) $A_n(x, y + 1) > A_n(x, y)$ für $n \geqq 1, x \geqq 2, y \geqq 0$.
(5) $A_n(x + 1, y) \geqq A_n(x, y)$ für $n \geqq 0, x \geqq 0, y \geqq 0$.
(6) $A_{n+1}(x, y) \geqq A_n(x, y)$ für $n \geqq 2, x \geqq 2, y \geqq 0$.

Die einfachen Induktionsbeweise wollen wir hier übergehen.

Lemma:

(a) $A_n(A_n(x, k), l) \leqq A_n(x, k \cdot l)$ für $n \geqq 2, x \geqq 2, k \geqq 0, l \geqq 0$.

(b) $A_{n-1}(A_n(x, k), A_n(x, l)) \leqq A_n(x, k + l)$ für $n \geqq 2, x \geqq 2, k \geqq 0, l \geqq 0$.

Wir beweisen die Konjunktion von (a) und (b) durch Induktion über n. Für $n = 2$ sind (a) und (b) offenbar richtig. Schluß von n auf $n + 1$: Zunächst beweisen wir (b), und zwar durch Induktion über k. Für $k = 0$ und $k = 1$ ist die Behauptung richtig. Schluß von k auf $k + 1$:

$$\begin{aligned} A_n(A_{n+1}(x, k+1), A_{n+1}(x, l)) &= A_n\big(A_n(x, A_{n+1}(x, k)), A_{n+1}(x, l)\big) \\ &\leqq A_n(x, A_{n+1}(x, k) \cdot A_{n+1}(x, l)) \\ &\leqq A_n\big(x, A_n(A_{n+1}(x, k), A_{n+1}(x, l))\big) \\ &\leqq A_n\big(x, A_{n+1}(x, k + l)\big) \\ &= A_{n+1}(x, k + 1 + l) . \end{aligned}$$

Mit Hilfe von (b) läßt sich jetzt (a) durch Induktion über l beweisen. Für $l = 0$ ist die Behauptung richtig. Schluß von l auf $l + 1$:

$$\begin{aligned} A_{n+1}(A_{n+1}(x, k), l+1) &= A_n\big(A_{n+1}(x, k), A_{n+1}(A_{n+1}(x, k), l)\big) \\ &\leqq A_n(A_{n+1}(x, k), A_{n+1}(x, k \cdot l)) \\ &\leqq A_{n+1}(x, k(l + 1)) . \end{aligned}$$

Mit diesen Hilfsmitteln erhält man jetzt leicht die folgende Abschätzung der Funktionen aus $\mathfrak{E}_n$:

Satz: Zu jedem $f \in \mathfrak{E}_n (n \geqq 0)$ gibt es ein k, so daß mit $\hat{\mathfrak{x}} := \max(\mathfrak{x}, 2)$ gilt

$$f(\mathfrak{x}) \leqq A_{n+1}(\hat{\mathfrak{x}}, k) .$$

Den Beweis führen wir durch Induktion über den Aufbau von $\mathfrak{E}_n$. Für die Ausgangsfunktionen $\lambda x\, x + 1$, $\lambda \mathfrak{x}\, x_i$, $\lambda \mathfrak{x}\, a$ ist die Behauptung in den Fällen $n = 0$ und $n = 1$ offenbar richtig; für $n \geqq 2$ folgt sie dann aus (6). Für A_n ist der Fall $n = 0$ ebenfalls trivial; für $n \geqq 1$ ist $A_n(x_1, x_2) \leqq A_n(\hat{\mathfrak{x}}, x_2) \leqq A_n(\hat{\mathfrak{x}}, \hat{\mathfrak{x}}) = A_n(\hat{\mathfrak{x}}, A_{n+1}(\hat{\mathfrak{x}}, 1)) = A_{n+1}(\hat{\mathfrak{x}}, 2)$. Die Operation der beschränkten Rekursion führt selbstverständlich von Funktionen mit der behaupteten Eigenschaft zu ebensolchen. Daß dies auch für die Einsetzung gilt, läßt sich mit Hilfe von (a) leicht beweisen:

Es sei

$$\begin{aligned} f(\mathfrak{x}) &= g(h_1(\mathfrak{x}), \ldots, h_r(\mathfrak{x})) , \\ h_i(\mathfrak{x}) &\leqq A_{n+1}(\mathfrak{x}, k_i) \qquad \text{für } 1 \leqq i \leqq r , \\ g(\mathfrak{x}) &\leqq A_{n+1}(\mathfrak{x}, k) . \end{aligned}$$

Wegen (4) kann man annehmen, daß k und alle $k_i \geqq 2$ sind. Man erhält

$$\begin{aligned} f(\mathfrak{x}) &\leqq A_{n+1}\big(\max(h_1(\mathfrak{x}), \ldots, h_r(\mathfrak{x}), 2), k\big) \\ &\leqq A_{n+1}\big(\max(A_{n+1}(\hat{\mathfrak{x}}, k_1), \ldots, A_{n+1}(\hat{\mathfrak{x}}, k_r)), k\big) \\ &= \max\big(A_{n+1}(A_{n+1}(\hat{\mathfrak{x}}, k_1), k), \ldots, A_{n+1}(A_{n+1}(\hat{\mathfrak{x}}, k_r), k)\big) \\ &\leqq \max(A_{n+1}(\hat{\mathfrak{x}}, k_1 k), \ldots, A_{n+1}(\hat{\mathfrak{x}}, k_r k)) \\ &= A_{n+1}(\hat{\mathfrak{x}}, \max(k_1, \ldots, k_r) \cdot k) . \end{aligned}$$

Korollar: $\mathfrak{E}_n \subset \mathfrak{E}_{n+1}$ für $n \geqq 0$.

Beweis: Man zeigt leicht $A_n(x, y) \leqq A_{n+1}(x+1, y+1)$ für $n \geqq 0$, $x \geqq 0$, $y \geqq 0$ und damit $A_i \in \mathfrak{E}_n$ für $0 \leqq i \leqq n$ (durch Induktion über i). Also ist $\mathfrak{E}_n \subseteq \mathfrak{E}_{n+1}$. Daß diese Inklusion echt ist, ergibt sich mit einem Diagonalschluß aus dem vorigen Satz: Zunächst ist $\lambda x\, A_{n+1}(x, x) + 1 \in \mathfrak{E}_{n+1}$. Läge diese Funktion in $\mathfrak{E}_n$, so gäbe es ein $k \geqq 2$ mit $A_{n+1}(x, x) + 1 \leqq A_{n+1}(x, k)$ für alle $x \geqq 2$. Setzt man $x = k$, so erhält man einen Widerspruch.

Schließlich beweisen wir noch zwei später benötigte Lemmata über Abschlußeigenschaften der Grzegorczyk-Klassen.

Lemma: Für $n \geqq 2$ ist $\mathfrak{E}_n$ abgeschlossen gegen simultane beschränkte Rekursionen. Das heißt:

Aus	$f_i(\mathfrak{x}, 0) = g_i(\mathfrak{x})$	für $1 \leqq i \leqq r$,
	$f_i(\mathfrak{x}, y+1) = h_i(\mathfrak{x}, y, f_1(\mathfrak{x}, y), \ldots, f_r(\mathfrak{x}, y))$	für $1 \leqq i \leqq r$,
	$f_i(\mathfrak{x}, y) \leqq j_i(\mathfrak{x}, y)$	für $1 \leqq i \leqq r$,
	$g_i, h_i, j_i \in \mathfrak{E}_n$	für $1 \leqq i \leqq r$,
folgt	$f_i \in \mathfrak{E}_n$	für $1 \leqq i \leqq r$.

Beweis: Zunächst gibt es in $\mathfrak{E}_2$ Funktionen $\sigma_1, \sigma_{r1}, \ldots, \sigma_{rr}$ mit $\sigma_{ri}(\sigma_r(z_1, \ldots, z_r)) = z_i$ für $1 \leqq i \leqq r$ und $\sigma_r(z_1, \ldots, z_r) \leqq \sigma_r(\bar{z}_1, \ldots, \bar{z}_r)$ falls $z_1 \leqq \bar{z}_1, \ldots, z_r \leqq \bar{z}_r$ (s. etwa Grzegorczyk [2], S. 4f. und S. 31f.). Wir setzen

$$f(\mathfrak{x}, y) := \sigma_r(f_1(\mathfrak{x}, y), \ldots, f_r(\mathfrak{x}, y)) .$$

Entsprechend seien g, h und j definiert. Dann sind $g, h, j \in \mathfrak{E}_n$ und es gilt

$$\begin{aligned} &f(\mathfrak{x}, 0) = g(\mathfrak{x}) \\ &f(\mathfrak{x}, y+1) = h\big(\mathfrak{x}, y, \sigma_{r1}(f(\mathfrak{x}, y)), \ldots, \sigma_{rr}(f(\mathfrak{x}, y))\big) \\ &f(\mathfrak{x}, y) \leqq j(\mathfrak{x}, y) . \end{aligned}$$

Also ist $f \in \mathfrak{E}_n$. Mit $f_i(\mathfrak{x}, y) = \sigma_{ri}(f(\mathfrak{x}, y))$ folgt die Behauptung.

Lemma: Für $n \geqq 2$ ist $\mathfrak{E}_n$ abgeschlossen gegen „beschränkte Summationen". Das heißt: Aus $f(\mathfrak{x}, y) = \sum_{i<y} g(\mathfrak{x}, i)$ und $g \in \mathfrak{E}_n$ folgt $f \in \mathfrak{E}_n$.

Beweis: Offenbar genügt es zu zeigen, daß f von einer $\mathfrak{E}_n$-Funktion majorisiert wird. Zunächst gibt es ein k mit

$$g(\mathfrak{x}, i) \leqq A_{n+1}(\max(\mathfrak{x}, i, 2), k) .$$

Also: $$f(\mathfrak{x}, y) \leqq y \cdot A_{n+1}(\max(\mathfrak{x}, y, 2), k) .$$

Nun ist $\lambda x\, A_{n+1}(x, k) \in \mathfrak{E}_n$, wie man durch Induktion über k leicht beweist. Mit $\lambda xy \max(x, y) \in \mathfrak{E}_1$ (Beweis: $\max(x, y) = x + (y \dot{-} x)$) folgt

$$\lambda \mathfrak{x} y\, y \cdot A_{n+1}(\max(\mathfrak{x}, y, 2), k) \in \mathfrak{E}_n.$$

2. Rekursionszahlen. In [3] definiert Heinermann die folgenden Klassen $\mathfrak{R}_n$ primitiv-rekursiver Funktionen:

Definition: $f \in \mathfrak{R}_n (n \geqq 0)$ genau dann, wenn

1) f eine der Ausgangsfunktionen $\lambda x\, x + 1$, $\lambda x_1 \ldots x_m x_i (1 \leqq i \leqq m)$, $\lambda x_1 \ldots x_m\, a$ $(m \geqq 0, a \geqq 0)$ ist, oder
2) f durch eine Einsetzung aus bereits erhaltenen Funktionen aus $\mathfrak{R}_n$ definiert ist, oder
3) $n \geqq 1$ ist und f durch eine primitive Rekursion aus bereits erhaltenen Funktionen aus $\mathfrak{R}_{n-1}$ definiert ist.

$\mathfrak{R}_n$ enthält also genau die primitiv-rekursiven Funktionen, zu denen es eine Definition aus den Ausgangsfunktionen durch Einsetzungen und primitive Rekursionen gibt, in der die Maximalzahl „übereinander liegender" primitiver Rekursionen oder kurz die „Rekursionszahl" höchstens den Wert n hat. Die in [1] von Axt eingeführten Klassen K_n sind mit den $\mathfrak{R}_n$ identisch.

Ganz ähnlich definieren wir noch Klassen $\mathfrak{R}_n^{\mathrm{sim}} \subseteq \mathfrak{P}$: Man ersetze in der vorstehenden Definition die Wörter „primitive Rekursion" durch „simultane primitive Rekursion" und „$\mathfrak{R}_n$" durch „$\mathfrak{R}_n^{\mathrm{sim}}$". Unter einer simultanen primitiven Rekursion verstehen wir eine Rekursion der Form

$$\begin{aligned} f_i(\mathfrak{x}, 0) &= g_i(\mathfrak{x}) && \text{für } 1 \leqq i \leqq r \\ f_i(\mathfrak{x}, y+1) &= h_i(\mathfrak{x}, y, f_1(\mathfrak{x}, y), \ldots, f_r(\mathfrak{x}, y)) && \text{für } 1 \leqq i \leqq r . \end{aligned}$$

Später wird es darauf ankommen, von gewissen Funktionen nachzuweisen, daß sie kleine Rekursionszahlen haben. Für diese Beweise sollen jetzt einige Hilfsmittel bereitgestellt werden. Wir beginnen mit einer aus [3] übernommenen Liste von Funktionen aus $\mathfrak{R}_1$, $\mathfrak{R}_2$ und $\mathfrak{R}_3$. Links steht jeweils die Bezeichnung des Funktionswerts an der Stelle x bzw. x, y und rechts eine Definition, aus der sich ablesen läßt, daß die betreffende Funktion in $\mathfrak{R}_1$, $\mathfrak{R}_2$ bzw. in $\mathfrak{R}_3$ liegt:

$\mathfrak{R}_1$:	$x + y$	$x + 0 = x$
		$x + (y + 1) = (x + y) + 1$
	$x \dot{-} 1$	$0 \dot{-} 1 = 0$
		$(x + 1) \dot{-} 1 = x$
	$x \cdot \overline{\mathrm{sg}}(y)$	$x \cdot \overline{\mathrm{sg}}(0) = x$
		$x \cdot \overline{\mathrm{sg}}(y + 1) = 0$

$\mathfrak{R}_2$: $x \cdot y$ $\quad x \cdot 0 = 0$

$x \cdot (y+1) = x \cdot y + x$

2^x $\quad 2^0 = 1$

$2^{x+1} = 2^x + 2^x$

$x \dot{-} y$ $\quad x \dot{-} 0 = x$

$x \dot{-} (y+1) = (x \dot{-} y) \dot{-} 1$

$\max(x, y)$ $\quad \max(x, y) = (x \dot{-} y) + y$

$\frac{1}{2} x(x+1)$ $\quad \frac{1}{2} x(x+1) = 1 + 2 + \cdots + x$

$r(x, y)$ $\quad$ Rest bei der Division von x durch y

Mit $h(0, y) = y \dot{-} 1$

$h(x+1, y) = (h(x, y) \dot{-} 1) + (y \dot{-} 1) \cdot \overline{\mathrm{sg}}(h(x, y))$

ist $r(x, y) = (y \dot{-} 1) \dot{-} h(x, y)$.

$\left[\frac{x}{2}\right]$ $\quad \left[\frac{x}{2}\right] = r\left(\frac{1}{2} x(x+1), x\right) + r\left(\frac{1}{2}(x \dot{-} 1)\, x, x \dot{-} 1\right)$

$\mathrm{maxpot}_2(x)$ $\quad$ Größte Zweierpotenz $\leqq x$

Mit $g(0) = 0$

$g(x+1) = (g(x) \dot{-} 1) + (x+1) \cdot \overline{\mathrm{sg}}(g(x) \dot{-} 1)$

ist $\mathrm{maxpot}_2(x) = \left[\frac{1}{2}(g(x) + x)\right]$.

$\mathfrak{R}_3$: $\exp_2(x)$ $\quad \exp_2(x) = \sum_{y \leqq x} y \cdot \overline{\mathrm{sg}}((x \dot{-} 2^y) + (2^y \dot{-} x))$

$= \begin{cases} y & \text{falls es ein } y \text{ gibt mit } x = 2^y \\ 0 & \text{sonst} \end{cases}$

Weiter beweisen wir einige Lemmata über die zu den $\mathfrak{R}_n$ gehörenden Relationenklassen $\mathrm{Rel}(\mathfrak{R}_n)$:

Definition: $\mathrm{Rel}(\mathfrak{R}_n)$ sei die Klasse der Relationen, die sich in der Form $\lambda x_1 \ldots x_m\, f(x_1, \ldots, x_m) = 0$ $(m \geqq 0)$ mit $f \in \mathfrak{R}_n$ darstellen lassen.

Lemma: Für $n \geqq 1$ ist $\mathrm{Rel}(\mathfrak{R}_n)$ abgeschlossen gegen die Operationen der Aussagenlogik.

Beweis: $\quad \neg\, [f(\mathfrak{x}) = 0] \leftrightarrow \overline{\mathrm{sg}}(f(\mathfrak{x})) = 0$

$[f(\mathfrak{x}) = 0] \wedge [g(\mathfrak{x}) = 0] \leftrightarrow f(\mathfrak{x}) + g(\mathfrak{x}) = 0$

Lemma: $\quad \lambda x\, x = y \in \mathrm{Rel}(\mathfrak{R}_1),\ \lambda x y\, x = y \in \mathrm{Rel}(\mathfrak{R}_2)$.

Beweis: $\quad x = y \leftrightarrow x \dot{-} y = 0 \wedge (x+1) \dot{-} y \neq 0$.

Lemma: Für $n \geqq 1$ ist $\mathfrak{R}_n$ abgeschlossen gegen Definitionen durch Fallunterscheidung. Das heißt:

Aus $$f(\mathfrak{x}) = \begin{cases} g_1(\mathfrak{x}) & \text{falls} \quad Q_1(\mathfrak{x}) \\ \dots & \\ g_k(\mathfrak{x}) & \text{falls} \quad Q_k(\mathfrak{x}), \end{cases}$$

auf jedes $\mathfrak{x}$ trifft genau ein Q_k zu,

$g_1, \dots, g_k \in \mathfrak{R}_n$, $Q_1, \dots, Q_k \in \mathrm{Rel}(\mathfrak{R}_n)$

folgt $f \in \mathfrak{R}_n$.

Beweis: Wegen $Q_1, \dots, Q_k \in \mathrm{Rel}(\mathfrak{R}_n)$ gibt es Funktionen $h_1, \dots, h_k \in \mathfrak{R}_n$, so daß für $1 \leqq i \leqq k$ gilt $Q_i(\mathfrak{x}) \leftrightarrow h_i(\mathfrak{x}) = 0$. Die Behauptung folgt aus

$$f(\mathfrak{x}) = g_1(\mathfrak{x}) \cdot \overline{\mathrm{sg}}(h_1(\mathfrak{x})) + \dots + g_k(\mathfrak{x}) \cdot \overline{\mathrm{sg}}(h_k(\mathfrak{x})) .$$

Schließlich definieren wir noch Funktionen $\tau_r, \tau_{r1}, \dots, \tau_{rr}$ $(r \geqq 1)$ mit den folgenden, später benötigten Eigenschaften:

1) $\tau_{ri}(\tau_r(z_1, \dots, z_r)) = z_i \quad \text{für} \quad 1 \leqq i \leqq r$.

2) $\tau_r \in \mathfrak{R}_2$, $\tau_{r1}, \dots, \tau_{rr} \in \mathfrak{R}_3$.

3) $\lambda z\, \tau_{ri}(z) = c \in \mathrm{Rel}(\mathfrak{R}_2) \quad \text{für} \quad 1 \leqq i \leqq r$.

4) Es gibt Funktionen $a_{ri}, s_{ri} \in \mathfrak{R}_2$ $(1 \leqq i \leqq r)$ mit

$$a_{ri}(\tau_r(z_1, \dots, z_i, \dots, z_r)) = \tau_r(z_1, \dots, z_i + 1, \dots, z_r) ,$$
$$s_{ri}(\tau_r(z_1, \dots, z_i, \dots, z_r)) = \tau_r(z_1, \dots, z_i \dot{-} 1, \dots, z_r) .$$

Sei $$\tau_r(z_1, \dots, z_r) := 2^{z_1' + \dots + z_r'} + 2^{z_2' + \dots + z_r'} + \dots + 2^{z_r'} + 2^0 \quad z_i' := z_i + 1,$$
$$t_i(z) := \mathrm{maxpot}_2\left(z \dot{-} \sum_{j<i} t_j(z)\right) \qquad \text{für} \quad 1 \leqq i \leqq r+1,$$
$$\tau_{ri}(z) := \left(\exp_2(t_i(z)) \dot{-} \exp_2(t_{i+1}(z))\right) \dot{-} 1 \qquad \text{für} \quad 1 \leqq i \leqq r .$$

Offenbar ist $t_i(\tau_r(z_1, \dots, z_r))$ der i-te Term der oben ausgeschriebenen Summe; damit ergibt sich 1). 2) ist nach dem Vorherigen trivial, und 3), 4) folgen mit $t_i \in \mathfrak{R}_2$ aus

$$\tau_{ri}(z) = c \leftrightarrow (t_i(z) = 2^{c+1} t_{i+1}(z) \wedge t_i(z) \geqq 2) \vee (c = 0 \wedge t_i(z) \leqq 1)$$
$$a_{ri}(z) := 2\, t_1(z) + \dots + 2\, t_i(z) + t_{i+1}(z) + \dots + t_{r+1}(z)$$
$$s_{ri}(z) := \begin{cases} \left[\frac{t_1(z)}{2}\right] + \dots + \left[\frac{t_i(z)}{2}\right] + t_{i+1}(z) + \dots + t_{r+1}(z) & \text{falls} \quad \tau_{ri}(z) \neq 0 \\ z & \text{sonst.} \end{cases}$$

3. Die unbeschränkte Register-Maschine (URM). Shepherdson und Sturgis untersuchen in [8] eine gewisse idealisierte Rechenmaschine, die sie unbeschränkte Register-Maschine (URM) nennen. Gegenüber Turingmaschinen hat die URM

den Vorteil, daß sich der Ablauf von Rechnungen auf ihr durch relativ einfache Funktionen beschreiben läßt; insbesondere haben diese Funktionen kleine Rekursionszahlen. Davon werden wir später Gebrauch machen. In diesem Abschnitt geben wir eine kurze Beschreibung der URM und führen einige Begriffe und Bezeichnungen ein.

Die URM verfügt über unbeschränkt viele Register $R_1, R_2, \ldots$, von denen jedes eine beliebig große natürliche Zahl $0, 1, 2, \ldots$ aufnehmen kann. Sie arbeitet nach „Programmen". Unter einem Programm verstehen wir hier eine endliche nummerierte Liste aus Elementarbefehlen der folgenden Form:

$A_i c$	„Addiere 1 zum Inhalt a_i von R_i. Gehe über zum c-ten Befehl".
$S_i c$	„Subtrahiere 1 vom Inhalt a_i von R_i, falls $a_i \neq 0$. Sonst lasse a_i unverändert. Gehe über zum c-ten Befehl".
$E_i c_0 c_1$	„Prüfe den Inhalt a_i von R_i. Ist $a_i = 0 (\neq 0)$, so gehe über zum $c_0 (c_1)$-ten Befehl". (E erinnert an „Entscheiden").
S	„Stoppe".

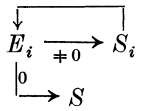
Gibt man der mit einem Programm versehenen URM irgendwelche Anfangsinhalte $a_1, a_2, \ldots$ ihrer Register vor, so rechnet sie in einer durch das Programm eindeutig bestimmten Weise, beginnend mit dem ersten Befehl. Es kann sein, daß sie nach endlich vielen Schritten stoppt, sie kann aber auch unbegrenzt weiterlaufen. Man beachte, daß jedes Programm nur mit endlich vielen Registern arbeitet, nämlich mit denen, auf die sich die Elementarbefehle beziehen. Die Inhalte der anderen Register haben auf die Rechnung keinen Einfluß und sie werden im Verlauf der Rechnung nicht verändert.

Eine m-stellige zahlentheoretische Funktion f heißt URM-berechenbar, wenn es ein Programm P gibt, das folgendes leistet: Setzt man die mit P programmierte URM an auf die Registerinhalte $x_1, \ldots, x_m, 0, 0, \ldots$, so stoppt sie nach endlich vielen Schritten, und zwar mit den Registerinhalten $x_1, \ldots, x_m, f(x_1, \ldots, x_m), 0, 0, \ldots$ Mit den üblichen Methoden kann man zeigen, daß genau die rekursiven Funktionen URM-berechenbar sind (vgl. Shepherdson/Sturgis [8]).

Jeder abbrechenden Rechnung auf der URM kann man eine Schrittzahl zuordnen, indem man jede Ausführung eines Elementarbefehls (außer S) als einen Schritt ansieht. Bei der Berechnung einer Funktion f hängt diese Schrittzahl i. a. vom Argumentetupel ab. Man erhält also eine Schrittzahlfunktion s_f mit derselben Stellenzahl wie f. Dieses s_f ist durch f nicht eindeutig bestimmt, sondern erst durch das gewählte Programm zur Berechnung von f. Das Zeichen „s_f" wird hier als Variable für derartige Schrittzahlfunktionen verwendet. Außer exakten Schrittzahlfunktionen treten später noch Majoranten solcher Funktionen auf, also Funktionen $\bar{s}_f$, zu denen es ein s_f gibt mit $s_f(\mathfrak{x}) \leqq \bar{s}_f(\mathfrak{x})$ für alle $\mathfrak{x}$. Als Variable für solche Majoranten von Schrittzahlfunktionen verwenden wir das Zeichen „$\bar{s}_f$".

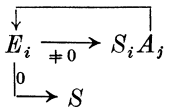
Wir geben jetzt noch einige einfache, später verwendete Programme an:

(1) L_i „Lösche den Inhalt des Registers R_i“:



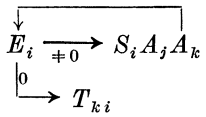
Es ist klar, wie man aus einem solchen Diagramm ein exaktes Programm gewinnt. Ist a_i der Inhalt von R_i, so ist die benötigte Schrittzahl $2a_i + 1$.

(2) T_{ij} „Transportiere den Inhalt von R_i nach R_j“:



Schrittzahl: $3\,a_i + 1$

(3) $K_{ij;k}$ „Kopiere den Inhalt von R_i in R_j unter Verwendung von R_k als Hilfsregister“:



Schrittzahl: $7\,a_i + 2$ (falls R_k zu Beginn der Rechnung leer ist).

4. Vergleich der Grzegorczyk-Klassen mit den Klassen $\mathfrak{R}_n$ und $\mathfrak{R}_n^{\text{sim}}$. Nach diesen Vorbereitungen können wir jetzt beweisen, daß die Klassen $\mathfrak{R}_n$ bzw. $\mathfrak{R}_n^{\text{sim}}$ mit den Grzegorczyk-Klassen $\mathfrak{E}_{n+1}$ schließlich zusammenfallen. Die Beweisidee ist sehr einfach (vgl. Meyer [4]): Sowohl die Funktionen aus $\mathfrak{R}_n$ bzw. $\mathfrak{R}_n^{\text{sim}}$ als auch die aus $\mathfrak{E}_{n+1}$ lassen sich für genügend große n dadurch charakterisieren, daß sie auf der URM mit einer in $\mathfrak{E}_{n+1}$ majorisierbaren Schrittzahlfunktion berechnet werden können.

Satz:
$$\mathfrak{R}_n^{\text{sim}} = \mathfrak{E}_{n+1} \quad \text{für} \quad n \geqq 2\,,$$
$$\mathfrak{R}_n \;\; = \mathfrak{E}_{n+1} \quad \text{für} \quad n \geqq 3\,.$$

Den Beweis gliedern wir in drei Teile:

Hilfssatz 1:
$$\mathfrak{R}_0^{\text{sim}} \subseteq \mathfrak{E}_0$$
$$\mathfrak{R}_1^{\text{sim}} \subseteq \mathfrak{E}_1$$
$$\mathfrak{R}_n^{\text{sim}} \subseteq \mathfrak{E}_{n+1} \quad \text{für} \quad n \geqq 2\,.$$

Hilfssatz 2: Für $n \geqq 2$ gibt es zu jedem $f \in \mathfrak{E}_n$ ein $s_f \in \mathfrak{E}_n$.

Hilfssatz 3: a) Ist $n \geqq 2$ und gibt es ein $\bar{s}_f \in \mathfrak{E}_{n+1}$, so ist $f \in \mathfrak{R}_n^{\text{sim}}$.

b) Ist $n \geqq 3$ und gibt es ein $\bar{s}_f \in \mathfrak{E}_{n+1}$, so ist $f \in \mathfrak{R}_n$.

Offenbar folgt aus diesen Hilfssätzen der Satz.

Beweis zu Hilfssatz 1: $\mathfrak{R}_0^{\mathrm{sim}} \subseteq \mathfrak{E}_0$ ist trivial. Zum Beweis von $\mathfrak{R}_1^{\mathrm{sim}} \subseteq \mathfrak{E}_1$ beachte man zunächst, daß $\mathfrak{R}_0^{\mathrm{sim}}$ genau alle Funktionen der Form $\lambda x_1 \ldots x_m x_i + k$ $(1 \leqq i \leqq m)$ und alle konstanten Funktionen enthält. Daraus ergibt sich leicht, daß eine simultane primitive Rekursion, angewandt auf Funktionen aus $\mathfrak{R}_0^{\mathrm{sim}}$, stets Funktionen aus $\mathfrak{E}_1$ liefert. Also ist $\mathfrak{R}_1^{\mathrm{sim}} \subseteq \mathfrak{E}_1$. (Jedoch ist $\mathfrak{R}_2^{\mathrm{sim}} \not\subseteq \mathfrak{E}_2$. Denn zu jedem einstelligen $f \in \mathfrak{E}_2$ gibt es ein k mit $f(x) \leqq x^k$ für $x \geqq 2$. Zu $\lambda x 2^x \in \mathfrak{R}_2^{\mathrm{sim}}$ gibt es aber kein solches k). Die allgemeine Behauptung beweisen wir durch Induktion über n. Sei also $n \geqq 2$ und $\mathfrak{R}_{n-1}^{\mathrm{sim}} \subseteq \mathfrak{E}_n$. Zu zeigen ist $\mathfrak{R}_n^{\mathrm{sim}} \subseteq \mathfrak{E}_{n+1}$. Offenbar genügt es, folgendes zu beweisen: Aus

$$f_i(\mathfrak{x}, 0) = g_i(\mathfrak{x}) \qquad \text{für} \quad 1 \leqq i \leqq r,$$
$$f_i(\mathfrak{x}, y+1) = h_i(\mathfrak{x}, y, f_1(\mathfrak{x}, y), \ldots, f_r(\mathfrak{x}, y)) \quad \text{für} \quad 1 \leqq i \leqq r$$

und $g_1, \ldots, g_r, h_1, \ldots, h_r \in \mathfrak{E}_n$ folgt $f_1, \ldots, f_r \in \mathfrak{E}_{n+1}$. Da $\mathfrak{E}_{n+1}$ abgeschlossen ist gegen simultane beschränkte Rekursionen, genügt es zu zeigen, daß $f_1, \ldots, f_r$ in $\mathfrak{E}_{n+1}$ beschränkt sind. Zunächst gibt es ein $k \geqq 2$ mit

$$h_i(\mathfrak{x}, y, \mathfrak{z}) \leqq A_{n+1}(\max(\mathfrak{x}, y, \mathfrak{z}, 2), k) \quad \text{für} \quad 1 \leqq i \leqq r.$$

Wir beweisen durch Induktion über y

$$f_i(\mathfrak{x}, y) \leqq A_{n+1}\big(\max(\mathfrak{x}, g_1(\mathfrak{x}), \ldots, g_r(\mathfrak{x}), 2), k^y\big) \quad \text{für} \quad 1 \leqq i \leqq r.$$

Für $y = 0$ ist das richtig. Schluß von y auf $y+1$:

$$\begin{aligned} f_i(\mathfrak{x}, y+1) &\leqq A_{n+1}\big(\max(\mathfrak{x}, y, f_1(\mathfrak{x}, y), \ldots, f_r(\mathfrak{x}, y), 2), k\big) \\ &\leqq A_{n+1}\big(A_{n+1}\big(\max(\mathfrak{x}, g_1(\mathfrak{x}), \ldots, g_r(\mathfrak{x}), 2), k^y\big), k\big) \\ &\leqq A_{n+1}\big(\max(\mathfrak{x}, g_1(\mathfrak{x}), \ldots, g_r(\mathfrak{x}), 2), k^{y+1}\big). \end{aligned}$$

Beweis zu Hilfssatz 2: Wir führen den Beweis durch Induktion über den Aufbau von $\mathfrak{E}_n$. Für jedes $n \geqq 2$ ist also zu zeigen:

Lemma 1: Die Ausgangsfunktionen von $\mathfrak{E}_n$ sind mit Schrittzahlfunktionen in $\mathfrak{E}_n$ auf der URM berechenbar.

Lemma 2: Ist f durch eine Einsetzung aus Funktionen $g, h_1, \ldots, h_r \in \mathfrak{E}_n$ definiert, und sind $g, h_1, \ldots, h_r$ mit Schrittzahlfunktionen in $\mathfrak{E}_n$ URM-berechenbar, so gilt dies auch für f.

Lemma 3: Ist f durch eine beschränkte Rekursion aus Funktionen $g, h, j \in \mathfrak{E}_n$ definiert, und sind g, h mit Schrittzahlfunktionen in $\mathfrak{E}_n$ URM-berechenbar, so gilt dies auch für f.

Beweis zu Lemma 1: Die Ausgangsfunktionen $\lambda x\, x+1$, $\lambda \mathfrak{x}\, x_i$, $\lambda \mathfrak{x}\, a$ und $A_1 = \lambda x y\, x + y$ sind offenbar mit linearen Schrittzahlfunktionen auf der URM berechenbar. Zu zeigen bleibt, daß für jedes $n \geqq 2$ auch A_n die verlangte Eigenschaft hat. Dies folgt aber durch Induktion über n aus Lemma 3.

Vorbemerkung zum Beweis der Lemmata 2 und 3: P sei ein Programm zur Berechnung einer m-stelligen Funktion f. Offenbar kann man durch Änderung der Indizes der in P vorkommenden Elementarbefehle erreichen, daß die Berechnung

nicht mehr bezüglich der Register $R_1, \ldots, R_m, R_{m+1}$, sondern bezüglich irgendwelcher paarweise verschiedener Register $R_{i_1}, \ldots, R_{i_m}, R_{i_{m+1}}$ geleistet wird. Auch die eventuell verwendeten Hilfsregister können geeignet neu gewählt werden. Das in dieser Weise geänderte Programm bezeichnen wir mit $P_{i_1 \ldots i_m i_{m+1}}$. Es sei darauf hingewiesen, daß diese Bezeichnungsweise nicht eindeutig ist, da die Hilfsregister nicht berücksichtigt werden. Man denke sie sich jedesmal geeignet gewählt.

Beweis zu Lemma 2: Wir behandeln den Fall $f(x) = g(h_1(x), h_2(x))$; die allgemeine Behauptung beweist man entsprechend. P, P', P'' seien Programme zur Berechnung von $g, h_1, h_2 \in \mathfrak{E}_n$ mit Schrittzahlfunktionen $s_g, s_{h_1}, s_{h_2} \in \mathfrak{E}_n$. $f(x)$ kann dann nach dem folgenden Diagramm berechnet werden:

$$P''_{12} P'_{13} P_{234} L_2 L_3 T_{42} .$$

Schrittzahl: $s_{h_1}(x) + s_{h_2}(x) + s_g(h_1(x), h_2(x)) + L(h_1(x), h_2(x), f(x))$.
L steht dabei für eine nicht näher interessierende lineare Funktion. Auf Grund der Voraussetzungen liegt diese Schrittzahlfunktion in $\mathfrak{E}_n$.

Beweis zu Lemma 3: Zur Vereinfachung nehmen wir an, daß f zweistellig ist; den allgemeinen Fall beweist man entsprechend. P, P' seien Programme zur Berechnung von $g, h \in \mathfrak{E}_n$ mit Schrittzahlfunktionen $s_g, s_h \in \mathfrak{E}_n$. $f(x, y)$ läßt sich dann wie folgt berechnen:

$$T_{24} P_{13} \overset{\downarrow}{E_4} \xrightarrow{\neq 0} P'_{1235} L_3 T_{53} A_2 S_4$$
$$\overset{|0}{\longrightarrow} S$$

Schrittzahl: $s_g(x) + L'(x) + \sum\limits_{i<y} \big(s_h(x, i, f(x, i)) + L''(f(x, i), f(x, i+1))\big)$.
L' und L'' sind dabei wieder lineare Funktionen. Da nach den Voraussetzungen auch $f \in \mathfrak{E}_n$ ist, und da wegen $n \geqq 2$ $\mathfrak{E}_n$ abgeschlossen ist gegen beschränkte Summationen, liegt auch diese Schrittzahlfunktion in $\mathfrak{E}_n$.

Beweis zu Hilfssatz 3: Zunächst zeigen wir, daß für $n \geqq 2$ aus der Existenz eines $\bar{s}_f$ in $\mathfrak{E}_{n+1}$ die Existenz eines $\bar{s}_f$ in $\mathfrak{R}_n$ folgt. Es gilt nämlich:

Lemma: Für $n \geqq 2$ gibt es zu jeder Funktion $g \in \mathfrak{E}_{n+1}$ eine sie majorisierende Funktion $h \in \mathfrak{R}_n$.

Beweis: Man betrachte die Funktionen

$$\begin{aligned} &\overline{A}_3(x, y) = 2^{xy} \\ &\overline{A}_{n+1}(x, 0) = 1 && \text{für } n \geqq 3 \\ &\overline{A}_{n+1}(x, y+1) = \overline{A}_n(x, \overline{A}_{n+1}(x, y)) && \text{für } n \geqq 3 . \end{aligned}$$

Wegen $\lambda xy\, x \cdot y$, $\lambda x 2^x \in \mathfrak{R}_2$ ist $\overline{A}_3 \in \mathfrak{R}_2$ und damit $\overline{A}_{n+1} \in \mathfrak{R}_n$ für $n \geqq 2$. Durch Induktion beweist man leicht $A_{n+1}(x, y) \leqq \overline{A}_{n+1}(x, y)$ für $n \geqq 2$, $x \geqq 0$, $y \geqq 0$. Es sei nun $n \geqq 2$ und $g \in \mathfrak{E}_{n+1}$. Dann gibt es ein k mit

$$\begin{aligned} g(\mathfrak{x}) &\leqq A_{n+2}(\hat{\mathfrak{x}}, k) \\ &\leqq \overline{A}_{n+2}(\hat{\mathfrak{x}}, k) . \end{aligned}$$

Mit $\lambda x\, \overline{A}_{n+2}(x, k) \in \mathfrak{R}_n$ (Beweis: Induktion über k) und $\lambda xy \max(x, y) \in \mathfrak{R}_2$ folgt die Behauptung.

Es sei jetzt eine Funktion f vorgelegt durch ein Programm P, nach dem sie auf der URM berechnet werden kann. Die Schrittzahlfunktion s_f werde durch ein $\bar{s}_f \in \mathfrak{R}_n$ majorisiert. Zu zeigen ist a) $f \in \mathfrak{R}_n^{\text{sim}}$, falls $n \geqq 2$, und b) $f \in \mathfrak{R}_n$, falls $n \geqq 3$. Zunächst beachte man, daß sich das Programm nur auf endlich viele Register beziehen kann. Sie mögen alle unter $R_1, \ldots, R_r$ vorkommen. Die „Konfiguration" der ganzen Apparatur vor Ausführung eines Rechenschritts ist also vollständig beschrieben durch Angabe der Inhalte $a_1, \ldots, a_r$ der Register $R_1, \ldots, R_r$ und der Nummer c der Programmzeile, die als nächste in Aktion treten wird. Genauer sollen unter Konfigurationen der mit P programmierten URM $r+1$-Tupel $(a_1, \ldots, a_r, c)$ verstanden werden, wobei c alle Zeilennummern von P und $a_1, \ldots, a_r$ alle natürlichen Zahlen durchlaufen. Läßt man nun die URM nach dem Programm P einen Funktionswert $f(\mathfrak{x})$ berechnen, so entspricht dieser Rechnung eine endliche Folge von Konfigurationen $\mathfrak{k}_1, \ldots, \mathfrak{k}_s$. Zur Vereinfachung denke man sich diese Folge konstant fortgesetzt, also $\mathfrak{k}_s = \mathfrak{k}_{s+1} = \mathfrak{k}_{s+2} = \cdots$. Die Komponenten der y-ten Konfiguration bezeichnen wir der Reihe nach mit $R_1(\mathfrak{x}, y), \ldots, R_r(\mathfrak{x}, y)$, $Z(\mathfrak{x}, y)$.

Zum Beweis von a) genügt es zu zeigen, daß die Funktionen $R_1, \ldots, R_r, Z$ in $\mathfrak{R}_2^{\text{sim}}$ liegen. Denn ist m die Stellenzahl von f, so gilt nach Definition der URM-Berechenbarkeit

$$f(\mathfrak{x}) = R_{m+1}(\mathfrak{x}, \bar{s}_f(\mathfrak{x}))\,.$$

Wir zeigen, daß sich $R_1, \ldots, R_r, Z$ durch eine simultane primitive Rekursion aus $\mathfrak{R}_1$-Funktionen definieren lassen. Zunächst ist $R_i(\mathfrak{x}, 0) = x_i$ für $1 \leqq i \leqq m$, $R_i(\mathfrak{x}, 0) = 0$ für $m+1 \leqq i \leqq r$ und $Z(\mathfrak{x}, 0) = 1$. Weiter gilt für $1 \leqq i \leqq r$

$$R_i(\mathfrak{x}, y+1) = \begin{cases} R_i(\mathfrak{x}, y) + 1 & \text{falls } Z(\mathfrak{x}, y) \text{ Nummer eines } A_i\text{-Befehls ist} \\ R_i(\mathfrak{x}, y) \dot{-} 1 & \text{falls } Z(\mathfrak{x}, y) \text{ Nummer eines } S_i\text{-Befehls ist} \\ R_i(\mathfrak{x}, y) & \text{sonst.} \end{cases}$$

$Z(\mathfrak{x}, y+1)$ läßt sich wie folgt aus $R_1(\mathfrak{x}, y), \ldots, R_r(\mathfrak{x}, y), Z(\mathfrak{x}, y)$ definieren: Man unterscheide die Fälle $Z(\mathfrak{x}, y) = 1, \ldots, Z(\mathfrak{x}, y) = z$, wobei $1, \ldots, z$ die Nummern der Befehle im Programm P sind. Ist c Nummer eines Prüfbefehls, etwa E_i, so unterscheide man zusätzlich $Z(\mathfrak{x}, y) = c \wedge R_i(\mathfrak{x}, y) = 0$ und $Z(\mathfrak{x}, y) = c \wedge R_i(\mathfrak{x}, y) \neq 0$. In jedem dieser Fälle sei dann $Z(\mathfrak{x}, y+1)$ die aus dem Programm abzulesende Nummer des jeweils nächsten Befehls. Mit den in Abschnitt 2 bereitgestellten Lemmata ergibt sich daraus $R_1, \ldots, R_r, Z \in \mathfrak{R}_2^{\text{sim}}$. Damit ist a) bewiesen.

Zum Beweis von b) fassen wir die Funktionen $R_1, \ldots, R_r, Z$ mit Hilfe von τ_{r+1} (s. Abschnitt 2) zu einer Funktion K zusammen:

$$K(\mathfrak{x}, y) := \tau_{r+1}(R_1(\mathfrak{x}, y), \ldots, R_r(\mathfrak{x}, y), Z(\mathfrak{x}, y))$$

$f(\mathfrak{x})$ läßt sich dann darstellen als

$$f(\mathfrak{x}) = \tau_{r+1\, m+1}\big(K(\mathfrak{x}, \bar{s}_f(\mathfrak{x}))\big)\,.$$

Wegen $\tau_{r+1\,m+1} \in \mathfrak{R}_3$ genügt es also, $K \in \mathfrak{R}_3$ zu beweisen. Gezeigt wird, daß K durch eine primitive Rekursion aus $\mathfrak{R}_2$-Funktionen definiert werden kann. Zunächst ist $K(\mathfrak{x}, 0) = \tau_{r+1}(x_1, \ldots, x_m, 0, \ldots, 0, 1)$. Zur Definition von $K(\mathfrak{x}, y+1)$ aus $K(\mathfrak{x}, y)$ verwenden wir dieselbe Fallunterscheidung wie bei der Definition von $Z(\mathfrak{x}, y+1)$ aus $R_1(\mathfrak{x}, y), \ldots, R_r(\mathfrak{x}, y), Z(\mathfrak{x}, y)$. Diese Fälle können in der Form $Q_j(K(\mathfrak{x}, y))$, $1 \leqq j \leqq p$, formuliert werden, wobei die Q_j aus Prädikaten $\lambda x\, \tau_{r+1\,i}(x) = c \in \mathrm{Rel}(\mathfrak{R}_2)$ mit den Operationen der Aussagenlogik aufgebaut sind. In jedem dieser Fälle erhält man $K(\mathfrak{x}, y+1)$ aus $K(\mathfrak{x}, y)$ durch Vergrößern oder Verkleinern der „Komponenten" um Konstante. Die entsprechenden Funktionen $h_1, \ldots, h_p$ können also aus den Funktionen $a_{r+1\,i}, s_{r+1\,i} \in \mathfrak{R}_2$, $1 \leqq i \leqq r+1$, (s. Abschnitt 2) zusammengesetzt werden. Insgesamt ergibt sich

$$K(\mathfrak{x}, y+1) = \begin{cases} h_1(K(\mathfrak{x}, y)) & \text{falls} \quad Q_1(K(\mathfrak{x}, y)) \\ \cdots & \\ h_p(K(\mathfrak{x}, y)) & \text{falls} \quad Q_p(K(\mathfrak{x}, y)) \end{cases}$$

mit $h_1, \ldots, h_p \in \mathfrak{R}_2$ und $Q_1, \ldots, Q_p \in \mathrm{Rel}(\mathfrak{R}_2)$. Damit ist auch b) bewiesen.

LITERATUR

[1] Axt, P., Iteration of primitive recursion. Z. math. Logik Grundlagen d. Math. *11* (1965), 253—255.

[2] Grzegorczyk, A., Some classes of recursive functions. Rozprawy Matematyczne IV, Warszawa 1953.

[3] Heinermann, W., Untersuchungen über die Rekursionszahlen rekursiver Funktionen. Dissertation, Münster 1961.

[4] Meyer, A. R., Depth of nesting and the Grzegorczyk hierarchy. Notices of the Amer. Math. Soc. *12* (1965), 342.

[5] Ritchie, D. M., Complexity classification of primitive recursive functions by their machine programs. Notices of the Amer. Math. Soc. *12* (1965), 343.

[6] Ritchie, R. W., Classes of recursive functions based on Ackermann's function. Pacific J. Math. *15* (1965), 1027—1044.

[7] Rödding, D., Klassen rekursiver Funktionen. S. 159—222 in: Proceedings of the Summer School in Logic, Leeds, 1967. Berlin (Springer) 1968.

[8] Shepherdson, J. C., and H. E. Sturgis, Computability of recursive functions. J. Ass. Computing Machinery, *10* (1963), 217—255.