ÜBER DIE REKURSIVITÄT EINIGER ÜBERSETZUNGS-TRANSFORMATIONEN (I. MITTEILUNG)

von

RÓZSA PÉTER

1. Sowohl in der exakten Formulierung der einzelnen Zweige der Mathematik als auch in der Programmierung von Rechenautomaten spielt die Angabe der verwendeten Sprachen und die Art der Übersetzung solcher Sprachen auf einander eine wichtige Rolle. Ich behaupte, dass bei exakter Formulierung die betreffenden Übersetzungs-Transformationen auf einer »Wortemenge« mit geeignet gewähltem »Alphabet« primitiv-rekursiv sind.

In vorliegender I. Mitteilung beschränke ich mich auf das folgende Teilproblem:

Man trachtet (besonders bei den praktischen Anwendungen) eine exakte Sprache möglichst ökonomisch anzugeben. In einer solchen Sprache ist der »Ausdruck« immer ein zentraler Begriff (algebraischer, logischer, oder im allgemeinsten Sinn betrachteter abstrakter Ausdruck). Zum Aufzeichnen eines Ausdrucks werden üblich Anfangsklammern und Endklammern benutzt. LUKASIEWICZ hat eine eindeutige klammernfreie Bezeichnung eingeführt, auf Kosten einer wesentlichen Änderung der Reihenfolge der Zeichen eines Ausdrucks.[1] KALMÁR[2] hat bewiesen, dass man im Fall von höchstens zwei Operationen bei Behaltung der ursprünglichen Reihenfolge die Endklammern ersparen kann, und dass sein Bezeichnungssystem sogar dann eindeutig bleibt, wenn gewisse Zeichen auch mehrdeutig verwendet werden (wie z. B. $\sim$ manchmal als Negationszeichen, manchmal als Äquivalenzzeichen verwendet wird; ein Anfangsklammer kann mit dem LUKASIEWICZschen Zeichen C der Implikation verwechselt werden; K kann sowohl als Zeichen der Konjunktion als auch als Zeichen einer Konstante vorkommen).

Das Bezeichnungssystem mit der üblichen Klammernverwendung in den Ausdrücken soll mit (S_1) bezeichnet werden, und das LUKASIEWICZsche klammernfreie System mit (S_2). Die Ausdrücke der Systeme (S_1) und (S_2) können ein-eindeutig auf einander übersetzt werden. In vorliegender Arbeit beweise ich, dass die betreffenden Übersetzungs-Transformationen auf einer geeigneten »Wortemenge« primitiv-rekursiv sind. (Dasselbe gilt auch für Übersetzungen zwischen Ausdrücke von Systemen mit anderen Klammernkonventionen; auf den Fall des KALMÁRschen Halbklammernsystems, und anderer Systeme, deren

[1] Siehe z.B. L. KALMÁR, Another proof of the Markov-Post theorem, *Acta Math. Acad. Sci. Hungaricae* **3** (1952) S. 1—27.

[2] Mündliche Mitteilung von L. KALMÁR, vorgetragen in Juni 1960 am Kolloquium »Sprachen und Algorithmen« in Berlin.

Behandlung etwas umständlicher ist, komme ich noch zurück.) Ich beschränke mich dabei auf Ausdrücke mit 1- und 2-stelligen Operationen (es kann ja z. B. jede arithmetische Operation auf Addition und Multiplikation nebst einstelligen Operationen, jede logische Operation auf Negation und etwa Konjunktion zurückgeführt werden); der Gedankengang könnte aber leicht auch auf Systeme mit mehrstelligen Operationen übertragen werden.

2. Nun folgt die exakte Angabe der betrachteten Systeme.

Für beide Systeme ist eine abzählbare Menge $\mathfrak{V}$ der Zeichen $v_1, v_2, \ldots$ für Variablen (worunter auch die Konstanten zu zählen sind; hier ist eine Unterscheidung dieser Begriffe belanglos), ferner eine abzählbare Menge $\mathfrak{E}$ der Zeichen $\Delta_1, \Delta_2, \ldots$ für einstellige Operationen, endlich eine abzählbare Menge $\mathfrak{Z}$ der Zeichen $\Theta_1, \Theta_2, \ldots$ für zweistellige Operationen vorhanden. In (S_1) kommen ausserdem noch die beiden Klammernzeichen vor. $\mathfrak{V}, \mathfrak{E}$ und $\mathfrak{Z}$ sind paarweise fremd, und enthalten auch die Klammernzeichen nicht.

Das leere Wort Λ und jedes Element von $\mathfrak{V}$ gilt in beiden Systemen als Ausdruck. Ausserdem gelten für $i = 1, 2, \ldots$, falls $\mathfrak{a}_1$ und $\mathfrak{a}_2$ bereits Ausdrücke sind,

$$\text{in } (S_1) \quad \Delta_i\, \mathfrak{a}_1 \quad \text{und} \quad (\mathfrak{a}_1\, \Theta_i\, \mathfrak{a}_2)\,,$$

$$\text{in } (S_2) \quad \Delta_i\, \mathfrak{a}_1 \quad \text{und} \quad \Theta_i\, \mathfrak{a}_1\, \mathfrak{a}_2$$

als Ausdrücke.

So enthält ein Ausdruck von (S_1) genau soviele Anfangsklammern als Endklammern. Und ein nicht zu $\mathfrak{V}$ gehöriger Ausdruck $\mathfrak{a}$ von (S_1) beginnt entweder mit einem Δ_i-Zeichen (worauf ein Ausdruck folgt), oder mit einer Anfangsklammer. Im letzteren Fall ist er der Form $\mathfrak{a} = (\mathfrak{a}_1\, \Theta_i\, \mathfrak{a}_2)$, wo $\mathfrak{a}_1$ und $\mathfrak{a}_2$ eindeutig bestimmte Ausdrücke sind, und Θ_i ein eindeutig bestimmtes Element von $\mathfrak{Z}$ ist. Betrachtet man nämlich die Zeichenreihe, wodurch $\mathfrak{a}$ aufgeschrieben wird, so können zwar darin an verschiedenen Stellen Elemente von $\mathfrak{Z}$ auftreten, aber unser Θ_i ist das einzige unter diesen, vor welchem genau um 1 mehr Anfangsklammern als Endklammern stehen.

Ein nicht zu $\mathfrak{V}$ gehöriger Ausdruck $\mathfrak{a}$ von (S_2) beginnt entweder mit einem Δ_i-Zeichen (worauf ein Ausdruck folgt), oder mit einem Θ_i-Zeichen. Im letzteren Fall ist er der Form $\mathfrak{a} = \Theta_i\, \mathfrak{a}_1\, \mathfrak{a}_2$, wo $\mathfrak{a}_1$ und $\mathfrak{a}_2$ eindeutig bestimmte Ausdrücke sind (siehe Fussnote 1)).

Es ergeben sich daraus unmittelbar ein-eindeutige Übersetzungen der Ausdrücke der Systeme (S_1) und (S_2) aufeinander: wird für $j_1, j_2 = 1, 2;\, j_1 \neq j_2$ ein Ausdruck von (S_{j_1}) mit (eventuell mit Striche versehenem) $\mathfrak{a}$ und der ihm entsprechende Ausdruck von (S_{j_2}) mit $t_{j_1 j_2}(\mathfrak{a})$ bezeichnet, so gilt für $i = 1, 2, \ldots$

(1) falls

$$\mathfrak{a} \in \mathfrak{V}\,,$$

dann

$$t_{j_1 j_2}(\mathfrak{a}) = \mathfrak{a}\,;$$

(2) falls

$$\mathfrak{a} = \Delta_i\, \mathfrak{a}'\,,$$

dann

$$t_{j_1 j_2}(\mathfrak{a}) = \Delta_i\, t_{j_1 j_2}(\mathfrak{a}')\,;$$

(3) falls

$$\mathfrak{a} = (\mathfrak{a}'\, \Theta_i\, \mathfrak{a}'')\,,$$

dann

$$t_{12}(\mathfrak{a}) = \Theta_i\, t_{12}(\mathfrak{a}')\, t_{12}(\mathfrak{a}'')\,;$$

(4) falls

$$\mathfrak{a} = \Theta_i\, \mathfrak{a}'\, \mathfrak{a}'',$$

dann

$$t_{21}(\mathfrak{a}) = (t_{21}(\mathfrak{a}')\, \Theta_i\, t_{21}(\mathfrak{a}''))\,.$$

3. Das »Alphabet« $\mathfrak{A}$ soll als »Buchstaben« die Zeichen des Systems (S_1) (also die Elemente von $\mathfrak{B}$, $\mathfrak{C}$ und $\mathfrak{Z}$, ferner die Klammernzeichen, welche als Buchstaben des Alphabets fett gedruckt werden) enthalten. Die auf diesem Alphabet beruhende Wortemenge $\mathfrak{M}$ enthält jene Zeichenreihen (»Worte«), die von dem mit Λ bezeichneten leeren Wort ausgehend durch Anknüpfen der Buchstaben von $\mathfrak{A}$ entstehen. Das ist also eine »zahlenartig aufbaubare Menge«: die natürlichen Zahlen werden ja von 0 ausgehend durch »Nachfolgerbildung« d. h. durch Addieren von 1 aufgebaut; in der Wortemenge werden aber abzählbar viele »Nachfolgerfunktionen« angewandt. Ich habe[3] die Begriffe der auf zahlenartig aufbaubaren Mengen definierten verschiedenartigen rekursiven Funktionen eingeführt und ihre Zusammenhänge untersucht; hier zähle ich die bezüglichen Tatsachen für Wortemengen auf, auf die ich mich berufen werde.

1. Eine auf der Wortemenge $\mathfrak{M}$ definierte Funktion $f(x)$ heisst primitiv-rekursiv in $\mathfrak{M}$, falls sie von Λ und von den Anknüpfungsfunktionen xa_i (wo $a_i \in \mathfrak{A}$) ausgehend, durch endlich viele Substitutionen und Anwendungen des folgenden Schemas der primitiven Rekursion definiert werden kanu (wobei ${}^{-1}\Lambda = \Lambda$, und ${}^{-1}x$ für $x = a_1 a_2 \dots a_n$ das Wort $a_2 \dots a_n$ bedeutet):

$$\text{und für } a \in \mathfrak{A} \qquad \left.\begin{aligned} f(\Lambda) &= c \\ f(xa) &= g_a\big(x, f(x), f({}^{-1}(xa))\big) \end{aligned}\right\}, \tag{D}$$

wobei $c \in \mathfrak{A}$ und g_a für jedes a eine bereits definierte primitiv-rekursive Funktion ist. Es können dabei auch beliebig viele Parameter auftreten.

2. Mit $o(x)$ wird die »Ordnung« des Wortes x bezeichnet, wobei $o(\Lambda) = \Lambda$, und $o(x) = n$ für $x = a_1 a_2 \dots a_n$ ist.

3. Die natürlichen Zahlen

$$0,\ 1,\ 2,\ 3, \dots$$

können etwa mit

$$\Lambda,\ \mathbf{(},\ \mathbf{((},\ \mathbf{(((}, \dots$$

identifiziert werden.

4. Ähnlich wie ${}^{-1}x$ kann auch x^{-1} definiert werden: $\Lambda^{-1} = \Lambda$, und x^{-1} bedeutet für $x = a_1 \dots a_{n-1} a_n$ das Wort $a_1 \dots a_{n-1}$.

5. Falls i eine natürliche Zahl ist, dann bedeutet für $o(x) \geq i$

$$e_i(x) \quad \text{bzw.} \quad l_i(x)$$

[3] R. PÉTER, Über die Verallgemeinerung der Theorie der rekursiven Funktionen für abstrakte Mengen geeigneter Struktur als Definitionsbereiche, *Acta Math. Acad. Sci. Hungaricae* (eingegangen am 30-ten September 1959).

das aus den ersten i bzw. aus den letzten i Buchstaben von x bestehende Wort. (Für andere i oder x können diese Funktionen etwa als Λ definiert werden. Ähnliches gilt auch für die Folgenden.)

6. Falls i eine natürliche Zahl ist, dann bedeutet für $o(x) \geqq i$

$$^{-i}x \quad \text{bzw.} \quad x^{-i}$$

den durch Weglassen von $e_i(x)$ vom Anfang bzw. von $l_i(x)$ vom Ende von x zurückbleibenden Rest von x.

7. Die aufgezählten Funktionen (auch die Buchstaben des Alphabets als Konstanten) sind alle primitiv-rekursiv in $\mathfrak{M}$; auch alle zahlentheoretische primitiv-rekursive Funktionen lassen sich zu primitiv-rekursive Funktionen in $\mathfrak{M}$ ergänzen[4] (diese ausgedehnte Funktionen werde ich ebenso bezeichnen, wie die ursprünglichen zahlentheoretischen Funktionen). Ferner sieht man leicht, dass auch die Identitätsfunktion x und die zweistellige Anknüpfungsfunktion xy primitiv-rekursiv in $\mathfrak{M}$ sind.

8. In (D) wird $f(xa)$ mit Hilfe von $f(x)$ und $f(^{-1}(xa))$ definiert. x und $^{-1}(xa)$ gelten dabei als »unmittelbare Vorgänger« (d. h. Vorgänger nächstkleinerer Ordnung) von x. Als sämtliche Vorgänger eines Wortes werden ausser Λ seine zusammenhängende Bestandteile (»Abschnitte«) betrachtet; für diese wird auch eine zweckmässige Anordnung angegeben. Z. B. sind die Vorigänger des Wortes $x = a_1 a_2 a_3 a_4$ in dieser Reihenfolge:

$$\Lambda,\ a_1,\ a_2,\ a_1 a_2,\ a_3,\ a_2 a_3,\ a_1 a_2 a_3,\ a_4,\ a_3 a_4,\ a_2 a_3 a_4,\ a_1 a_2 a_3 a_4 .$$

Diese sind, abgesehen von x selbst, echte Vorgänger von x; die unmittelbaren Vorgänger von x sind hier $a_1 a_2 a_3$ und $a_2 a_3 a_4$.

$$y \leqq x \quad \text{bzw.} \quad y \prec x$$

gelten als kurze Bezeichnungen für die Beziehungen (die nach der bald folgenden Definition primitiv-rekursiv sind), dass y ein Vorgänger bzw. ein echter Vorgänger von x ist.

Da die natürliche Zahl n mit

$$\underbrace{((\ldots(}_{n\text{-mal}}$$

dentifiziert wurde, ist für natürliche Zahlen $x \prec y$ mit $x < y$ gleichbedeutend.

9. Einer Wortefolge $x_0, x_1, \ldots, x_n$ kann ein Wort $c_n(x_0, x_1, \ldots, x_n)$ als für festes n in $\mathfrak{M}$ primitiv-rekursive Funktion derart zugeordnet werden, dass für $i = 0, 1, \ldots, n$ mit in $\mathfrak{M}$ primitiv-rekursiven $k_y(x)$ und long (x)

$$x_i = k_i(c_n(x_0, \ldots, x_n))$$

und

$$n = \operatorname{long}(c_n(x_0, \ldots, x_n))$$

gilt.

10. Sind nun sämtliche Vorgänger eines Wortes x in der oben genannten Reihenfolge:

$$\bar{x}_0, \bar{x}_1, \ldots, \bar{x}_{s(x)},$$

[4] Bezüglich der allgemeinen Kentnisse über zahlentheoretische rekursive Funktionen berufe ich mich auf mein Buch: R. PÉTER, Rekursive Funktionen, Budapest, Akademischer Verlag, 2-te Auflage (1957).

dann sieht man leicht, dass $s(x)$ in $\mathfrak{M}$ primitiv-rekursiv ist, und dass mit in $\mathfrak{M}$ primitiv-rekursivem $v_y(x)$ für $i = 0, 1, \ldots, s(x)$

$$\overline{x}_i = v_i(x)$$

gilt. Ist ferner für ein $x = a_1 a_2 \ldots a_n$

$$\overline{x}_i = a_{i_1+1} a_{i_1+2} \ldots a_{i_1+i_2},$$

so sieht man leicht, dass

$$i = \binom{i_1 + i_2}{2} + i_2,$$

und daher i eine zahlentheoretische primitiv-rekursive Funktion von i_1 und i_2 ist.

Nun kann die »Wertverlaufsfunktion« $\varphi(x)$ einer Funktion $f(x)$ durch

$$\varphi(x) = c_{s(x)}(f(\overline{x}_0), f(\overline{x}_1), \ldots, f(\overline{x}_{s(x)}))$$

definiert werden, woraus sich für $i \leq s(x)$

$$f(\overline{x}_i) = k_i(\varphi(x)),$$

ferner

$$s(x) = \mathrm{long}\,(\varphi(x))$$

ergibt.

11. Wird in der Definition (D) $f(x)$ und $f(^{-1}(xa))$ durch $\varphi(x)$ bzw. $\varphi(^{-1}(xa))$ ersetzt, so wird aus (D) eine Wertverlaufsrekursion (D*); und ich habe bewiesen, dass auch die Anwendung von (D*) nicht von der Klasse der primitiv-rekursiven Funktionen von $\mathfrak{M}$ hinausführt.

12. Die charakteristische Funktion $b(x_1, \ldots, x_n)$ einer Wortbeziehung $B(x_1, \ldots, x_n)$ kann etwa durch

$$b(x_1, \ldots, x_n) = \begin{cases} \Lambda, & \text{wenn } B(x_1, \ldots, x_n) \\ (& \text{sonst} \end{cases}$$

definiert werden; und $B(x_1, \ldots, x_n)$ wird dann in $\mathfrak{M}$ primitiv-rekursiv genannt, wenn $b(x_1, \ldots, x_n)$ eine in $\mathfrak{M}$ primitive-rekursive Funktion ist. Es ist z. B. die charakteristische Funktion $z(x)$ der Beziehung »x ist der Form (((... («, und daher auch die Beziehung »eine natürliche Zahl zu sein« primitiv-rekursiv in $\mathfrak{M}$.

13. Aussagenlogische Verknüpfungen überführen primitiv-rekursive Beziehungen wieder in solche; ferner sind samt der Funktion $f(x)$ und der Beziehung $B(x, y)$ auch die Beziehungen

$$(Ey)\,[y \leq f(x) \;\&\; B(x, y)] \quad \text{und} \quad (y)\,[y \leq f(x) \rightarrow B(x, y)]$$

und auch die Funktion

$$\mu_y\,[y \leq f(x) \;\&\; B(x, y)]$$

— welche falls es Worte $y \leq f(x)$ mit $B(x, y)$ gibt, das erste solche y in der festgesetzten Reihenfolge der Vorgänger von $f(x)$, sonst aber Λ als Wert annimmt — primitiv-rekursiv in $\mathfrak{M}$. (Ausser den ausgeschriebenen Variablen können immer auch beliebig viele Parameter auftreten.)

14. Endlich werde ich noch benutzen, dass falls $B_i(x_1, \ldots, x_n)$ für $i = 1, 2, \ldots, r$ sich gegenseitig ausschliessende primitiv-rekursive Beziehungen

in $\mathfrak{M}$ sind, dann auch die aus in $\mathfrak{M}$ primitiv-rekursiven Funktionen $f_i(x_1, \ldots, x_n)$ $(i = 1, 2, \ldots, r, r+1)$ folgenderweise »zusammengeflickte« Funktion:

$$f(x_1, \ldots, x_n) = \begin{cases} f_1(x_1, \ldots, x_n), & \text{falls} \quad B_1(x_1, \ldots, x_n) \\ \ldots\ldots\ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots\ldots\ldots \\ f_r(x_1, \ldots, x_n), & \text{falls} \quad B_r(x_1, \ldots, x_n) \\ f_{r+1}(x_1, \ldots, x_n) & \text{sonst} \end{cases}$$

primitiv-rekursiv in $\mathfrak{M}$ ist.

4. Setzen wir von nun an der Reihe nach

$$f_v(x),\ f_\Delta(x),\ f_\Theta(x),\ f_{\boldsymbol{(}}(x),\ f_{\boldsymbol{)}}(x)$$

für die charakteristische Funktionen der Beziehungen

$$x \in \mathfrak{V},\ x \in \mathfrak{E},\ x \in \mathfrak{Z},\ x = \boldsymbol{(},\ x = \boldsymbol{)}\,.$$

Es ist klar, dass diese Funktionen primitiv-rekursiv in $\mathfrak{M}$ sind; z. B. kann $f_\Delta(x)$ durch folgende primitive Rekursion definiert werden:

$$f_\Delta(\Lambda) = \boldsymbol{(}$$

$$f_\Delta(xv_i) = f_\Delta(x\,\Theta_i) = f_\Delta(x\,\boldsymbol{(}\,) = f_\Delta(x\,\boldsymbol{)}) = \boldsymbol{(}$$

$$f_\Delta(x\,\Delta_i) = \begin{cases} \Lambda, & \text{falls} \quad x = \Lambda \\ \boldsymbol{(} & \text{sonst} \end{cases}$$

5. Seien

$$s_1(x) \text{ und } s_2(x)$$

die charakteristischen Funktionen der Beziehungen, dass x ein Ausdruck des Systems (S_1) bzw. (S_2)) ist. Ihre exakten Definitionen lauten:

$$s_1(\Lambda) = \Lambda$$

und für $a \in \mathfrak{A}$

$$s_1(xa) = \begin{cases} \Lambda, \text{ falls } f_v(xa) = \Lambda \vee (f_\Delta(e_1(x)) = \Lambda \,\&\, s_1({}^{-1}(xa)) = \Lambda) \vee \\ \quad \vee (Ey_1)(Ey_2)(Ey)\,[y_1, y_2, y \leqq x \,\&\, s_1(y_1) = s_1(y_2) = f_\Theta(y) = \Lambda \,\& \\ \quad \& \, y_1 \neq \Lambda \,\&\, y_2 \neq \Lambda \,\&\, xa = \boldsymbol{(}y_1\, y y_2\boldsymbol{)}] \\ \boldsymbol{(} \text{ sonst}; \end{cases}$$

ferner

$$s_2(\Lambda) = \Lambda$$

und für $a \in \mathfrak{A}$

$$s_2(xa) = \begin{cases} \Lambda, \text{ falls } f_v(xa) = \Lambda \vee (f_\Delta(e_1(x)) = \Lambda \,\&\, s_2({}^{-1}(xa)) = \Lambda) \vee \\ \quad \vee (f_\Theta(e_1(x)) = \Lambda \,\&\, (Ey_1)(Ey_2)\,[y_1, y_2 \leqq {}^{-1}(xa) \,\& \\ \quad \& \, s_2(y_1) = s_2(y_2) = \Lambda \,\&\, y_1 \neq \Lambda \,\&\, y_2 \neq \Lambda \,\&\, {}^{-1}(xa) = y_1\, y_2]) \\ \boldsymbol{(} \text{ sonst}. \end{cases}$$

Es handelt sich hier einfach um Wertverlaufsrekursionen. Das zeige ich für die Definition von $s_1(x)$; für $s_2(x)$ geht es ähnlich.

Die Existenz von $y_1, y_2, y \leqq x$ gewisser Eigenschaft ist gleichbedeutend mit der Existenz solcher natürlichen Zahlen $i_1, i_2, j \leqq s(x)$, für welche die Vorgänger $\bar{x}_{i_1}, \bar{x}_{i_2}, \bar{x}_j$ von x die gewünschte Eigenschaft besitzen. Diese sind der Reihe nach gleich

$$v_{i_1}(x),\ v_{i_2}(x),\ v_j(x)\,.$$

Ferner besteht ${}^{-1}(xa)$ aus genau sovielen Buchstaben wie x, daher ist die Anzahl $s({}^{-1}(xa))$ der Vorgänger von ${}^{-1}(xa)$ auch $s(x)$. Wird die Wertverlaufsfunktion von $s_1(x)$ mit $\sigma_1(x)$ bezeichnet, so gilt daher

$$s_1({}^{-1}(xa)) = k_{s(x)}\big(\sigma_1({}^{-1}(xa))\big)\,;$$

ferner ist

$$s_1(y_1) = s_1(\bar{x}_{i_1}) = k_{i_1}(\sigma_1(x))\,,$$

$$s_1(y_2) = s_1(\bar{x}_{i_2}) = k_{i_2}(\sigma_1(x))\,.$$

Daher kann die Definition von $s_1(x)$ folgendermassen umformuliert werden:

$$s_1(\Lambda) = \Lambda$$

und für $a \in \mathfrak{A}$

$$s_1(xa) = \begin{cases} \Lambda, \text{ falls } f_v(xa) = \Lambda \vee \big(f_\Delta(e_1(x)) = \Lambda \,\&\, k_{s(x)}\big(\sigma_1({}^{-1}(xa))\big) = \Lambda\big) \vee \\ \vee (E_{i_1})(E_{i_2})(Ej)\,[i_1, i_2, j \leqq s(x) \,\&\, z(i_1) = z(i_2) = z(j) = \Lambda \,\& \\ \&\, k_{i_1}(\sigma_1(x)) = k_{i_2}(\sigma_1(x)) = f_\Theta(v_j(x)) = \Lambda \,\& \\ \&\, v_{i_1}(x) \neq \Lambda \,\&\, v_{i_2}(x) \neq \Lambda \,\&\, xa = (v_{i_1}(x)\, v_j(x)\, v_{i_2}(x))] \\ (\ \text{sonst}, \end{cases}$$

das ist aber eine Wertverlaufsrekursion.

6. Jetzt können wir endlich die Übersetzungstransformationen untersuchen.

Für $i, j = 1, 2,\ i \neq j$ soll $t_{ij}(x)$, falls x ein Ausdruck des Systems (S_i) ist, die Übersetzung dieses Ausdrucks auf die Sprache des Systems (S_j) bedeuten: für andere Worte x sei $t_{ij}(x) = \Lambda$.

Zur Definition von $t_{12}(x)$ betrachten wir näher einen mit **(** beginnenden Ausdruck xa des Systems (S_1). Ein solcher ist der Form

$$xa = \mathbf{(} y_1\, \Theta_i\, y_2 \mathbf{)}\,,$$

wobei y_1 und y_2 nicht-leere Ausdrücke des Systems (S_1), und beide Vorgänger von x sind. Es gilt

$$y_2 = {}^{-\langle o(y_1)+2\rangle}x$$

und

$$y_1 = \mu_y\big[y \leqq x \,\&\, s_1(y) = \Lambda \,\&\, e_{x(y)}({}^{-1}x) = y \,\&\, f_\Theta\big(l_1(e_{o(y)+2}(x))\big) = \Lambda \,\&$$

$$\&\, s_1({}^{-\langle o(y)+2\rangle}x) = \Lambda\big]\,.$$

In den Weiteren sollen y_1 und y_2 immer diese in $\mathfrak{M}$ primitiv-rekursiven Funktionen von x bezeichnen.

Dann kann $t_{12}(x)$ folgenderweise definiert werden:

$$t_{12}(\Lambda) = \Lambda$$

und für $a \in \mathfrak{A}$

$$t_{12}(xa) = \begin{cases} a, \text{ falls } x = \Lambda \mathbin{\&} f_v(a) = \Lambda \\ e_1(x)\, t_{12}({}^{-1}(xa)), \text{ falls } s_1(xa) = \Lambda \mathbin{\&} f_\Delta(e_1(x)) = \Lambda \\ l_1(e_{o(y_1)+2}(x))\, t_{12}(y_1)\, t_{12}(y_2), \\ \quad \text{falls } s_1(xa) = \Lambda \mathbin{\&} e_1(x) = (\\ \Lambda \text{ sonst.} \end{cases}$$

Wegen $y_1, y_2 \leqq x$ zeigt man leicht, dass es sich hier um einen Spezialfall der Wertverlaufsrekursion (D*) handelt.

Wenn nämlich

$$x = a_1 a_2 \ldots a_{s(x)}$$

ist, so ist

$$y_1 = a_2 \ldots a_{o(y_1)+1}$$

und

$$y_2 = a_{o(y_1)+3} \ldots a_{s(x)};$$

diese sind aber Vorgänger $\bar{x}_{j_1}$ bzw. $\bar{x}_{j_2}$ von x, wobei

$$j_1 = \binom{o(y_1)+1}{2} + o(y_1) \quad \text{und} \quad j_2 = \binom{s(x)}{2} + \big(s(x) \dot{-} (o(y_1)+2)\big)$$

in $\mathfrak{M}$ primitiv-rekursive Funktionen von x sind. So gilt mit in $\mathfrak{M}$ primitiv-rekursiven $g_1(x)$ und $g_2(x)$

$$y_1 = \bar{x}_{g_1(x)}$$

und

$$y_2 = \bar{x}_{g_2(x)}.$$

Es soll nun $\tau_{12}(x)$ die Wertverlaufsfunktion von $t_{12}(x)$ bezeichnen.

Wie bereits in Nr. 5 bemerkt wurde, ist

$$s({}^{-1}(xa)) = s(x),$$

also gilt

$$t_{12}({}^{-1}(xa)) = k_{s(x)}\big(\tau_{12}({}^{-1}(xa))\big).$$

Ferner gilt nach den Vorherigen

$$t_{12}(y_1) = k_{g_1(x)}(\tau_{12}(x)) \quad \text{und} \quad t_{12}(y_2) = k_{g_2(x)}(\tau_{12}(x)).$$

Werden diese in die Definition der Funktion $t_{12}(x)$ eingesetzt, so erhält man

$$t_{12}(\Lambda) = \Lambda$$

und für $a \in \mathfrak{A}$

$$t_{12}(xa) = \begin{cases} a, \text{ falls } x = \Lambda \mathbin{\&} f_v(a) = \Lambda \\ e_1(x)\, k_{s(x)}\big(\tau_{12}({}^{-1}(xa))\big), \text{ falls } s_1(xa) = \Lambda \mathbin{\&} f_\Delta(e_1(x)) = \Lambda \\ l_1(e_{o(y_1)+2}(x))\, k_{g_1(x)}(\tau_{12}(x))\, k_{g_2(x)}(\tau_{12}(x)), \\ \quad \text{falls } s_1(xa) = \Lambda \mathbin{\&} e_1(x) = (\\ \Lambda \text{ sonst,} \end{cases}$$

und das ist tatsächlich ein Spezialfall der Wertverlaufsrekursion (D*).

7. Ganz ähnlich geht auch die Untersuchung von $t_{21}(x)$.

Ein mit einem Element Θ_i der Menge $\mathfrak{Z}$ beginnender Ausdruck xa von (S_2) ist der Form

$$xa = \Theta_i z_1 z_2 ,$$

wobei z_1, z_2 nicht-leere Ausdrücke von (S_2), und beide echte Vorgänger von ${}^{-1}(xa)$ sind. Es gilt

$$z_2 = {}^{-(o(z_1)+1}(xa)$$

und

$$z_1 = \mu_z [z \leq x \,\&\, s_2(z) = \Lambda \,\&\, e_{o(z)}({}^{-1}x) = z \,\&\, s_2({}^{-(o(z)+1}(xa)) = \Lambda] .$$

Mit diesen lautet die Definition von $t_{21}(x)$:

$$t_{21}(\Lambda) = \Lambda$$

und für $a \in \mathfrak{A}$

$$t_{21}(xa) = \begin{cases} a, \text{ falls } x = \Lambda \,\&\, f_v(a) = \Lambda \\ e_1(x)\, t_{21}({}^{-1}(xa)), \text{ falls } s_2(xa) = \Lambda \,\&\, f_\Delta(e_1(x)) = \Lambda \\ (t_{21}(z_1)\, e_1(x)\, t_{21}(z_2)) , \\ \quad \text{falls } s_2(xa) = \Lambda \,\&\, f_\Theta(e_1(x)) = \Lambda \\ \Lambda \text{ sonst,} \end{cases}$$

Wegen $z_1, z_2 \leq {}^{-1}(xa)$ beweist man dem Verfahren in Nr. 6 ähnlich, dass auch dies ein Spezialfall der Wertverlaufsrekursion (D*) ist.

(Eingegangen: 11. September, 1961.)

О РЕКУРСИВНОСТИ ПЕРЕВОДНЫХ ТРАНСФОРМАЦИЙ (I. СООБЩЕНИЕ)

R. PÉTER

Резюме

Как в экзактной формулировке отдельных отрасль математики, так и в программировании арифмографов, важную роль выполняет вопрос, какой язык применяется и какие языки каким образом переводятся друг на друга. Я ставил себе целью доказывать, что указанные переводные трансформации являются в альфавитном отношении способно выбранном «множестве слов» примитивными рекурсиями.

В этом первом сообщении я ограничиваюсь на следующую частную проблему:

Мы стараемся, особенно в практике применений, задавать какой-то экзактный язык насколько только возможно экономично. В таком языке «выражение» (алгебраическое, логическое или в самом общепринятом значении абстрактное выражение) является всегда центральным понятием.

В написании какого-нибудь выражения обыкновенно применяются в начинающие и закрывающие скобки. Лукашевиц заводил существенное изменение в порядок стоящих при выражении знаков однозначное бесскобочное обозначение. Выражение скобочной (S_1) и бесскобочной (S_2) системы обозначения можно взаимно и однозначно переводить друг на друга. В статье доказываю, что соответствующие переводные трансформации являются примитивными рекурсивными на одном подходящем множестве слов. (На системы, коренившиеся в других скобочных конвенциях имеют действие подобные правила; к таким, например к «полускобочной» системе обозначения Калмара я следующий раз вернусь.