Plaser

INSTITUT FÜR NUMERISCHE MATHEMATIK UND
ELEKTRONISCHE INFORMATIONSVERARBEITUNG
UNIVERSITÄT INNSBRUCK

Associating functions and the operator of conditioned iteration +)

Bruno Buchberger

Bericht Nr. 71 - 1 Juni 1971

+) This is an english version of the russian original, forthcoming as a report of the Joint Institute for Nuclear Research, Dubna.

A - 6020 Innsbruck, Innrain 2

SUMMARY

Introducing the notion of "associating functions" (section 1) and the "operator of conditioned iteration" (definition 2.1) we give a new representation theorem for the recursive functions. In section 3 the result is applied to the semantics of programming languages.

(04) We define functions Φ_m (m> o), that turn out to be universal.

$$\Phi_m(x,x_1,\ldots,x_m)=\mathrm{eval}(< x,< x_1,\ldots,x_m> ,\pi>)_{\bar{o},\bar{4},\bar{o}}, \text{ where } \pi = << \bar{o},\bar{o},\bar{o}>> ,$$

$$\mathrm{eval}(\eta) = \Omega^{P}(\eta),$$

$$\Omega(\eta) = (\eta_{\bar{o},\bar{o}} = \bar{1} \rightarrow \mathrm{subst}(\eta_{\bar{2}}, g(\eta_{\bar{1},\bar{o}})),$$

$$\eta_{\bar{o},\bar{o}} = \bar{2} \rightarrow \mathrm{subst}(\eta_{\bar{2}}, \eta_{\bar{1},\eta_{\bar{o},\bar{1}}}),$$

$$\eta_{\bar{o},\bar{o}} = \bar{3} \rightarrow \mathrm{subst}(\eta_{\bar{2}}, \eta_{\bar{o},\bar{1}}),$$

$$\eta_{\bar{o},\bar{o}} = \bar{4} \rightarrow \mathrm{subst}(\eta_{\bar{2}}, \alpha(\eta_{\bar{1},\bar{o}}, \eta_{\bar{1},\bar{1}})),$$

$$\eta_{\bar{o},\bar{o}} = \bar{5} \rightarrow \mathrm{subst}(\eta_{\bar{2}}, \sigma(\eta_{\bar{1},\bar{o}}, \eta_{\bar{1},\bar{1}}, \eta_{\bar{1},\bar{2}})),$$

$$\eta_{\bar{o},\bar{o}} = \bar{6} \rightarrow (\eta_{\bar{o},\bar{1}} = \eta_{\bar{o},\bar{2}} \rightarrow [\eta, \bar{o}, \eta_{\bar{o},\bar{3}}, \bar{1}, \eta_{\bar{o},\bar{4}}],$$

$$\quad \text{else} \rightarrow [\eta, \bar{o}, \eta_{\bar{o},\bar{4},\eta_{\bar{o},\bar{2}}}, \bar{2}, \eta]),$$

$$\eta_{\bar{o},\bar{o}} = \bar{7} \rightarrow (\eta_{\bar{1},\bar{o},\bar{o},\bar{o}} = \bar{o} \rightarrow \mathrm{subst}(\eta_{\bar{2}}, \eta_{\bar{1},\bar{o}}),$$

$$\quad \text{else} \rightarrow [\eta, \bar{o}, \eta_{\bar{o},\bar{1}}, \bar{2}, [\eta, \bar{o}/\bar{o}, \bar{8}]]),$$

$$\eta_{\bar{o},\bar{o}} = \bar{8} \rightarrow (\eta_{\bar{o},\bar{4},\bar{o},\bar{o},\bar{o}} = \bar{o} \rightarrow \mathrm{subst}(\eta_{\bar{2}}, \eta_{\bar{o},\bar{4},\bar{o}}),$$

$$\quad \text{else} \rightarrow [\eta, \bar{o}, \eta_{\bar{o},\bar{1}}, \bar{1}, \eta_{\bar{o},\bar{4}}, \bar{2}, [\eta, \bar{o}/\bar{2}, \bar{o}]]),$$

$$\eta_{\bar{o},\bar{o}} = \bar{9} \to \quad (\eta_{\bar{o},\bar{2}} = \eta_{\bar{o},\bar{3}} \quad \to [\eta,\bar{o},\eta_{\bar{o},\bar{4}},\bar{1},\eta_{\bar{o},\bar{5}}],$$

$$\text{else} \quad \to \quad [\eta\,,\bar{o}/\bar{5}/\eta_{\bar{o},\bar{1}},\eta_{\bar{1},\eta_{\bar{c},\bar{2}}},\bar{o}/\bar{1},\nu\,(\eta_{\bar{o},\bar{1}}),\bar{o}/\bar{2},\nu\,(\eta_{\bar{o},\bar{2}})]\,),$$

$$\text{else} \to \quad \bar{o}),$$

$$\text{subst}(d,r) = \quad [d,\bar{o}/\bar{4}/d_{\bar{o},\bar{2}},r,\bar{o}/\bar{2},\nu\,(d_{\bar{o},\bar{2}})\,].$$

We denote $\eta_{\bar{2}^o} = \eta$, $\eta_{\bar{2}^k} = (\eta_{\bar{2}^{k-1}})_{\bar{2}}$, and define

D e f i n i t i o n 2.5: $\eta \sim \eta'$, if

$$(Ek)((i<k)\;(\eta_{\bar{2}^i,\bar{o},\bar{j}} = \eta'_{\bar{2}^i,\bar{o},\bar{j}} \qquad \text{for } j = o,1,2,3$$

$$\eta_{\bar{2}^i,\bar{o},\bar{4},a} = \eta'_{\bar{2}^i,\bar{o},\bar{4},a} \qquad \text{for } a \in A,$$

$$\eta_{\bar{2}^i,\bar{o},\bar{5},a} = \eta'_{\bar{2}^i,\bar{o},\bar{5},a} \qquad \text{for } a \in A,$$

$$\eta_{\bar{2}^i,\bar{1},a} = \eta'_{\bar{2}^i,\bar{1},a} \qquad \text{for } a \in A) \quad \wedge \quad (\eta_{\bar{2}^k} = \eta'_{\bar{2}^k})\,).$$

We need the following lemmas:

L e m m a 2.8: The relation $\eta \sim \eta'$ is an equivalence relation.

L e m m a 2.9: $\eta \sim \eta' \Longrightarrow \Omega(\eta) \sim \Omega(\eta')$.

L e m m a 2.10: $\eta \sim \eta' \Longrightarrow \text{eval}(\eta) \sim \text{eval}(\eta')$.

P r o o f s: Lemma 2.8 is obvious. For proving Lemma 2.9 one has to evaluate all components of $\Omega\,(\eta)$ and $\Omega(\eta')$ which are used in Definition 2.5

(using Lemma 1.8 and 1.9) and to compare them. This procedure is cumbersome, but logically easy. We omit details. Lemma 2.10 is immediate consequence of Lemma 2.8 and 2.9 and the definition of the operator P.

We show now that Φ_m has the properties of a universal function. Analysing the structure of Φ_m and using Lemma 1.1, 1.2, 1.5, 1.7, and 2.7, we obtain the first property necessary for universality.

L e m m a 2.11: $\Phi_m \in B^1(G)_{m+1}$ (for $m > o$).

Let ${}^{\Phi}F_m = \{f \mid f(x_1,\dots,x_m) = \Phi_m(x,x_1,\dots,x_m)$ for $x \in A\}$, ${}^{\Phi}F = \bigcup_m {}^{\Phi}F_m$.

It is clear, that ${}^{\Phi}F_m \subseteq B(G)_m$ (using Lemma 2.11). For showing

$B(G)_m \subseteq {}^{\Phi}F_m$ $(m > o)$ we prove $B(G) \subseteq {}^{\Phi}F$, even $B(G) \subseteq {}^{\Phi}\bar{F}$, where

$${}^{\Phi}\bar{F} = \bigcup_m {}^{\Phi}\bar{F}_m \quad \text{and}$$

$${}^{\Phi}\bar{F}_m = \{ f \mid f(x_1,\dots,x_m) = \Phi_m(x,x_1,\dots,x_m)$$
$$(d)(\mathrm{eval}(\langle x, \langle x_1,\dots,x_m\rangle, d\rangle) \sim \mathrm{eval}(\mathrm{subst}(d, f(x_1,\dots,x_m))))$$
$$\text{for } x \in A \}.$$

$B(G) \subseteq {}^{\Phi}\bar{F}$ follows from the assertions

(04.1) $G \subseteq {}^{\Phi}\bar{F}$

(04.2) ${}^{\Phi}\bar{F}$ is closed under substitution

(04.3) ${}^{\Phi}\bar{F}$ is closed under application of the operator P,

which we shall show in the sequel.

(04.1) We show $g \in {}^{\Phi}\bar{F}$. Let $\eta^d = \langle\langle \bar{1}\rangle, \langle x_1\rangle, d\rangle$, where d is arbitrary. We have $\eta^d_{\bar{o},\bar{o}} = \bar{1}$, $\eta^d_{\bar{1},\bar{o}} = x_1$, $\eta^d_{\bar{2}} = d$ (Lemma 1.6) and

$\eta^{d}_{\bar{o},\bar{o}} \neq \bar{o}$ (Lemma 1.3). Therefore,

(2.1) $eval(\eta^{d}) = eval(\Omega(\eta^{d})) = eval(subst(d,g(x_1)))$. Thus,

(2.2) $\Phi_1(\langle\bar{1}\rangle, x_1) = eval(\langle\langle \bar{1}\rangle,\langle x_1\rangle, \pi\rangle)_{\bar{o},\bar{4},\bar{o}} = eval(\eta^{\pi})_{\bar{o},\bar{4},\bar{o}} =$

$$eval(subst(\pi,g(x_1)))_{\bar{o},\bar{4},\bar{o}} \overset{(Lemma\ 1.9)^{+)}}{=} subst(\pi,g(x_1))_{\bar{o},\bar{4},\bar{o}} \overset{(Lemma\ 1.8)}{=} g(x_1).$$

From (2.1) and (2.2) it is evident that $g \in {}^{\Phi}\bar{F}$.

Analogously, one shows u^n_i, c^n_y, α, $\sigma \in {}^{\Phi}\bar{F}$. One only has to substitute the elements $\langle\bar{2},\overline{i-1}\rangle$, $\langle\bar{3},y\rangle$, $\langle\bar{4}\rangle$ and $\langle\bar{5}\rangle$ instead of $\langle\bar{1}\rangle$ into the expression $eval(\eta^{d})$ and the corresponding function Φ_m and to evaluate these terms.

(04.2) Let $h \in {}^{\Phi}\bar{F}_r$ and $f_1,\ldots,f_r \in {}^{\Phi}\bar{F}_n$ $(r \geq 1)$, i.e.

(2.3) $h(y_1,\ldots,y_r) = \Phi_r(h^*,y_1,\ldots,y_r)$,

(2.4) $f_i(x_1,\ldots,x_n) = \Phi_n(f_i^*,x_1,\ldots,x_n)$ $(i = 1,\ldots,r)$,

(2.5) $eval(\langle h^*, \langle y_1,\ldots,y_r\rangle, d\rangle) \sim eval(subst(d,h(y_1,\ldots,y_r)))$,

(2.6) $eval(\langle f_i, \langle x_1,\ldots,x_n\rangle, d\rangle) \sim eval(subst(d,f_i(x_1,\ldots,x_n)))$ $(i = 1,\ldots,r)$

for g^*, $f_1^*,\ldots,f_r^* \in A$.

We put $\eta^{d} = \langle\langle\bar{6},\bar{r},\bar{o},h^*, \langle f_1^*,\ldots,f_r^*\rangle\rangle, \langle x_1,\ldots,x_n\rangle, d\rangle$ and evaluate

---

+) $\overset{(Lemma\ 1.9)}{=}$ means: this equation can be derived using Lemma 1.9.

$$\mathrm{eval}(\eta^d) = \mathrm{eval}(\Omega(\eta^d)) \overset{(\bar r \neq \bar o)}{=} \mathrm{eval}(\,[\eta^d, \bar o, \eta^d_{\bar o,\bar 4,\, \eta_{\bar o, \bar 2}}, \bar 2, \eta^d]) \sim$$

$$\overset{(\text{Lemma } 2.10)}{\sim} \mathrm{eval}(\langle f_1^*, \langle x_1,\dots,x_n\rangle, \eta^d\rangle) \overset{(2.6)}{\sim} \mathrm{eval}(\mathrm{subst}(\eta^d, f_1(x_1,\dots,x_n))) \sim$$

$$\overset{(\text{Lemma } 2.10)}{\sim} \mathrm{eval}(\langle\langle \bar 6, \bar r, \bar 1, h^*, \langle f_1(x_1,\dots,x_n), f_2^*,\dots,f_r^*\rangle\rangle, \langle x_1,\dots,x_n\rangle, d\rangle).$$

The following steps don't require new methods. We, therefore, omit details. Using several times the definition of the function $\Omega(\eta)$, Lemma 2.10, the relations (2.3),...,(2.6), induction on r, and, finally, Lemma 2.8 we obtain the desired relations

(2.7) $\mathrm{eval}(\eta^d) \sim \mathrm{eval}(\mathrm{subst}(d, h(f_1(x_1,\dots,x_n),\dots,f_r(x_1,\dots,x_n))))$, and

(2.8) $\Phi_n(\langle \bar 6, \bar r, \bar o, h^*, \langle f_1^*,\dots,f_r^*\rangle\rangle, x_1,\dots,x_n) =$

$= h(f_1(x_1,\dots,x_n),\dots,f_r(x_1,\dots,x_n)).$

(04.3) Let $h \in {}^{\Phi}\bar F_1$, i.e.

(2.9) $h(x_1) = \Phi_1(h^*, x_1)$,

(2.10) $\mathrm{eval}(\langle h^*, \langle x_1\rangle, d\rangle) \sim \mathrm{eval}(\mathrm{subst}(d, h(x_1)))$ for $h^* \in A$.

Similarly to (04.2), using the "branches" $\eta_{\bar o,\bar o} = \bar 7$ and $\eta_{\bar o,\bar o} = \bar 8$ of the function Ω and induction on the number of iterations of the function h, we obtain the desired relation

(2.11) $\mathrm{eval}(\langle\langle \bar 7, h^*, \bar o\rangle, \langle x_1\rangle, d\rangle) = \mathrm{eval}(\mathrm{subst}(d, h^P(x_1)))$, and

(2.12) $\Phi_1(\langle \bar 7, h^*, \bar o\rangle, x_1) = h^P(x_1)$

(05) For $m,n > o$ we define

$$S^m_n(x,x_1,\dots,x_m) = \langle \bar{9},\bar{m},\bar{o},\bar{n},x,\langle x_1,\dots,x_m\rangle\rangle$$

Obviously $S^m_n \in B(G)_{m+1}$. The S-m-n-Theorem is shown by evaluating the expression

$$eval(\langle\langle \bar{9},\bar{m},\bar{o},\bar{n},x,\langle x_1,\dots x_m\rangle\rangle, \langle y_1,\dots,y_n\rangle, \pi\rangle_{\bar{o},\bar{4},\bar{o}}.$$

For this one has to carry out induction on n, using the "branch" $n_{\bar{o},\bar{o}} = \bar{9}$ of the function Ω.

Thus, the first assertion of Theorem 2.2 is proven. The second assertion follows from Lemma 2.11 and the fact that every function $f \in B(G)_n$ can be represented in the form

$$f(x_1,\dots,x_n) = \Phi_n(x,x_1,\dots,x_n) \quad \text{(for all } n > o),$$

which was just established.

3. An application to the semantics of programming languages

We start from the concept that the semantics of a programming language L can be given by a partial recursive function $\Phi_L(p,d)$ of two arguments, which for two numbers $p,d \in N$, coding a programm p^* and data d^* of L, computes the number $r = \Phi_L(p,d) \in N$, coding the result of the application of the program p^* to the data d^*. It seems that this precise substitute of the intuitive notion "semantics of a programming language" has arisen recently (see, for instance, Uspenskij [9], [10], Blum [1] and the work of the IBM Laboratory, Vienna, e.g. [4], [5]).

Using Theorem 2.1 we now can decompose such an "interpreting function"

$\Phi_L(p,d)$ into three functions $\gamma_L(p,d)$, $\bar{\Phi}_L(\eta)$ and $\rho_L(\eta)$, such that

$$\Phi_L(p,d) = \rho_L(\bar{\Phi}_L^P(\gamma_L(p,d))), \text{ where}$$

γ_L, $\bar{\Phi}_L$ and ρ_L can be obtained from α, σ, s, c_o^1 and the functions of U^N by a finite number of substitutions.

The function $\gamma_L(p,d)$ can be conceived as a function, which for every program p and data d defines an "initial state" $\eta_o = \gamma_L(p,d)$ of an automaton. For this a fixed number of operations, namely essentially "storage operations" α, σ, are necessary. $\bar{\Phi}_L(\eta)$ can be conceived as the transition function of the automaton, which does the actual computational work. The number of iterations of the function $\bar{\Phi}_L$ depends on the initial state η_o. The number of operations during one "step" (i.e. one application of the function $\bar{\Phi}_L(\eta)$), however, is again fixed. Also during computation by the automaton all operations, essentially, are "storage operations" α, σ. Finally, by applying ρ_L, the result of the computation is read from the final state η_f (for which $(\eta_f)_{\bar{o},\bar{o}} = \bar{o}$). This is again done by a fixed number of operations.

There arises the following problem: give necessary and sufficient conditions for ρ_L, $\bar{\Phi}_L$ and γ_L, in order that the function

$$\Phi_L(p,d) = \rho_L(\bar{\Phi}_L^P(\gamma_L(p,d)))$$

defines an effective Gödel numbering for F_1^{rek} (for the notion of effective Gödel numbering see Rogers [7], Uspenskij [10] p.294 and Strong [8]). Of course, the most interesting would be conditions for $\bar{\Phi}_L$. This problem, until now, is not solved. A solution would give interesting insight into the structure of programming languages and computer concepts.

N.N. Govorun, S.S. Lavrov, N.M. Nagornij, Yelena Nogina and G.A. Ososkov helped me during different stages of the work. V.Dushskij and E.Kurenniemi spent much time for discussing with me the subject of the present note. To all of them I want to express my sincere gratitude.

L i t e r a t u r e :

[1] E.K. Blum, Towards a theory of semantics and compilers for programming languages, J. of Computers and System Sciences, Vol. 3, Nr. 3, 1969.

[2] M. Davis, Computability and Unsolvability, McGraw Hill, 1958.

[3] H.M. Friedmann, Axiomatic recursive function theory, Standford University, 1969, unpublished.

[4] P. Lauer, Formal defintion of ALGOL 60. TR 25.088, IBM Laboratory, Vienna.

[5] P. Lucas, P. Lauer, H. Stigleitner, Method and notation for the formal definition of programming languages, TR 25.087, 1968, IBM Laboratory Vienna.

[6] A. I. Malzev, Algoritmy i rekursivnye funktsii, Izd. Nauka, Moskva,1965.

[7] H. Rogers, Jr., Gödel numberings of partial recursive functions. J. of Symbolic Logic, Vol. 23, Nr. 3, 1958.

[8] H.R. Strong, Algebraically generalized recursive function theory, IBM J. of Research and Development, Nov. 1958.

[9] V.A. Uspenskij, Vytshislimye operatsii i ponyatie programmy. Yspekhi matem. nauk 12, nr, 1, 1957.

[10] V.A. Uspenskij, Lektsii o vytshislimykh funktsiyakh, Gos. izd. fiz-mat. lit., Moskva, 1960.

[11] E.G. Wagner, Uniformly reflexive structures: on the nature of Gödelization and relative computability, Transactions of the Amer. Math. Soc., Vol. 144, Oct. 1969.

1. Associating functions

We use the following notation. N is the set of non-negative integer numbers. Let A be an arbitrary set. By F^A we denote the set of all partial functions with arbitrarily many arguments, the function values and arguments being elements of A. Let F be a set of functions, then F_n is the set of all n-ary functions of F, e.g. $F^A = \bigcup_n F_n^A$. For $n \geq 1$, $1 \leq i \leq n$ we define

(1.1) $u_i^n(x_1, \ldots, x_n) = x_i$,

and for $n \geq 1$, $y \in A$

(1.2) $c_y^n(x_1, \ldots, x_n) = y$.

U^A (C^A) is the set of all u_i^n (c_y^n) of F^A. The notions "substitution of partial functions of F^A", "minimalization operator μ" and "primitive recursion" are used in the ordinary sense (cf. Davis [2]). $\neg$, $\wedge$, (x) and (Ex) denote negation, conjunction, universal and existential quantification of x, respectively.

D e f i n i t i o n 1.1: The functions $\alpha \in F_2^A$ and $\sigma \in F_3^A$ are called associating functions (AF) for A, if

(A1) $(E\,\varepsilon \in A)(a \in A)(\alpha(a, \varepsilon) = \varepsilon)$, and

(A2) $(b,s,a,c \in A)\left(\alpha(b, \sigma(s,a,c)) = \begin{cases} c, & \text{if } b=a \\ \alpha(b,s), & \text{otherwise} \end{cases}\right)$.

This definition reflects the following intuitive concept. $c = \alpha(a,s)$ reads: c is the element ("content") corresponding to the element ("address") a in "storage" s, or "c is the a-th component of s". $s' = \sigma(s,a,c)$ reads: s' is a storage, whose a-th component is c, all the other components being the same as in s. (A2) guarantees this property, whereas (A1) guarantees the existence of a "void" storage ε, whose content is known, namely "void", for all addresses a.

In the following we consider a fixed domain A with given AF α, σ. We require the existence of at least two distinct elements in A. We agree that the variables i, j, k, l, m, n, t always range over N. These symbols and the numerals 0, 1, 2, ... used as indices denote normal indexing, whereas all remaining symbols at index position denote the application of the function α, e.g. s_a abbreviates $\alpha(a,s)$, and for $n \geq 2$ $s_{a_1, \dots, a_n} = \alpha(a_n, s_{a_1, \dots, a_{n-1}})$. Obviously,

L e m m a 1.1: The (n+1)-ary functions $s_{a_1, \dots, a_n}$ $(n \geq 1)$ may be obtained from α and the functions of U^A by a finite number of substitutions.

We fix $\delta \neq \varepsilon$ and define $\bar{o} = \sigma(\varepsilon, \delta, \delta)$, $\nu(x) = \sigma(\varepsilon, \varepsilon, x)$, and $\overline{n+1} = \nu(\bar{n})$ for $n \geq 0$. It is easy to check the following lemmas.

L e m m a 1.2: The function ν may be obtained from σ, c^1_ε, u^1_1 by a finite number of substitutions.

L e m m a 1.3: $\bar{n} \neq \varepsilon$ for $n \geq 0$ and $\bar{m} \neq \bar{n}$ for $m < n$.

Sketch of the proof for Lemma 1.3: $\varepsilon_\delta = \varepsilon$, but $\bar{o}_\delta = \delta \neq \varepsilon$, thus $\bar{o} \neq \varepsilon$. $\bar{1}_\varepsilon = \bar{o}$, but $\bar{o}_\varepsilon = \varepsilon \neq \bar{o}$, thus, $\bar{1} \neq \bar{o}$. $\bar{1}_\varepsilon = \bar{o}$, but $\varepsilon_\varepsilon = \varepsilon \neq \bar{o}$, thus, $\bar{1} \neq \varepsilon$. For arbitrary n: induction on n. From Lemma 1.3 immediately follows

L e m m a 1.4: If AF exist for a set A, then A is infinite.

We now introduce two abbreviating notations using the brackets $[\cdot]$ and $\langle,\rangle$:

$\langle x \rangle = \sigma\ (\varepsilon, \bar{o}, x)$, $\langle x_1, \ldots, x_n \rangle = \sigma\ (\langle x_1, \ldots, x_{n-1}\rangle, \overline{n-1}, x_n)$.

$[s,a,c] = \sigma\,(s,a,c)$,

$[s,a^{(1)}/\ldots/a^{(t)},c] = [s,a^{(1)}/\ldots/a^{(t-1)}, \sigma\,(s_a{}^{(1)}{}_{,\ldots,a}{}^{(t-1)}, a^{(t)},c)]$ for $t \geq 2$,

$[s,a_1^{(1)}/\ldots/a_1^{(t_1)}, c_1,\ldots,a_n^{(1)}/\ldots/a_n^{(t_n)}, c_n] =$

$= [[s,a_1^{(1)}/\ldots/a_1^{(t_1)}, c_1,\ldots,a_{n-1}^{(1)}/\ldots/a_{n-1}^{(t_{n-1})}, c_{n-1}], a_n^{(1)}/\ldots/a_n^{(t_n)}, c_n]$

for $n \geq 2$ and $t_j \geq 1$ ($j = 1,\ldots,n$).

We need the following lemmas:

L e m m a 1.5: The n-ary functions $\langle x_1,\ldots,x_n\rangle$ ($n \geq 1$) may be obtained from σ and the functions of C_1^A and U^A by a finite number of substitutions.

L e m m a 1.6: For $n \geq 1$

$$\langle x_1,\ldots,x_n\rangle\ a = \begin{cases} x_1, & \text{if } a = \bar{o} \\ \vdots & \\ x_n, & \text{if } a = \overline{n-1} \\ \varepsilon & \text{otherwise.} \end{cases}$$

L e m m a 1.7: The $(\sum_{j=1}^{n} t_j + n + 1)$-ary functions

$[s,a_1^{(1)}/\ldots/a_1^{(t_1)}, c_1,\ldots,a_n^{(1)}/\ldots/a_n^{(t_n)}, c_n]$ ($n \geq 1$, $t_j \geq 1$) can be obtained from α, σ and the functions of U^A by a finite number of substitutions.

L e m m a 1.8: Suppose $\neg(a_j^{(1)} = a_k^{(1)} \wedge \dots \wedge a_j^{(t_j)} = a_k^{(t_j)})$ for all $j,k \leq n$, $j \neq k$. Then $[s, a_1^{(1)}/\dots/a_1^{(t_1)}, c_1, \dots, a_n^{(1)}/\dots/a_n^{(t_n)}, c_n]_{b^{(1)},\dots,b^{(t)}} =$

$$= (c_l)_{b^{(t_l+1)},\dots,b^{(t)}}, \text{ if } a_l^{(1)} = b^{(1)} \wedge \dots \wedge a_l^{(t_l)} = b^{(t_l)}.$$

(We define $(c_l)_{b^{(t_l+1)},\dots,b^{(t)}} = c_l$, if $t_l = t$.)

L e m m a 1.9: Suppose $\neg(a_j^{(1)} = a_k^{(1)} \wedge \dots \wedge a_j^{(t_j)} = a_k^{(t_j)})$ for all $j,k \leq n$, $j \neq k$. Then

$$[s, a_1^{(1)}/\dots/a_1^{(t_1)}, c_1, \dots, a_n^{(1)}/\dots/a_n^{(t_n)}, c_n]_{b^{(1)},\dots,b^{(t)}} = s_{b^{(1)},\dots b^{(t)}},$$

$$\text{if } \neg(a_j^{(1)} = b^{(1)} \dots a_j^{(t_j)} = b^{(t_j)}) \wedge$$

$$\wedge \neg(b^{(1)} = a_j^{(1)} \dots b^{(t)} = a_j^{(t)}) \text{ for all } j \leq n.$$

P r o o f s : Lemmas 1.5 und 1.7 are obvious. Lemma 1.6 can easily be proven by induction on n, where Lemma 1.3 plays an important role. Lemmas 1.8 and 1.9 are proven by induction on t_1 and n. For starting the inductions, (A1) and (A2) are used.

$[s, a_1^{(1)}/\dots/a_1^{(t_1)}, c_1, \dots, a_n^{(1)}/\dots/a_n^{(t_n)}, c_n]$ is a convenient notation for a complex "storage operation". For instance, $[s,a/b,u,c/d/e,v]$ alters the content of storage s in the following way: the b-th component of the a-th component of storage s will be u, and the e-th component of the d-th component of the c-th component of storage s will be v. All the other components of s remain unchanged.

We give an example for AF. We put A = N, furthermore $p_o = 2$, p_n ... n-th prime number, exp(a,z) ... the exponent of p_a in the prime number decomposition of z ($a \geq o$, $z \geq 1$), and $l(z) = \mu_+(\exp(u,z) = o$ for $u \geq t)$.
Let us define

$$\alpha(a,s) = \exp(a,s+1), \quad \sigma(s,a,c) = p_a^c \cdot \prod_{\substack{j=o \\ j\neq a}}^{l(s+1)-1} p_j^{\exp(j,s+1)} - 1 .$$

It is easy to show that α, σ satisfy (A1) and (A2).

2. A representation theorem for the recursive functions.

D e f i n i t i o n 2.1: Let α, σ be AF for A. The "operator of conditioned iteration P" associates with every function $\phi \in F_1^A$ a function ϕ^P, which is defined by the following equation

$$\phi^P(\eta) = \begin{cases} \eta, & \text{if } \eta_{\bar{o},\bar{o}} = \bar{o} \\ \phi^P(\phi(\eta)), & \text{otherwise.} \end{cases}$$

We define $s(x) = x+1$ $(x \in N)$ and F^{rek} ... set of all partial recursive functions. The central aim of the present note is the proof of the following representation theorem for F^{rek}.

T h e o r e m 2.1: Let α, σ be AF for N. Every partial recursive function may be obtained from α, σ, c_o^1,s and the functionsof U^N by a finite number of substitutions and o n e application of the operator P.

In fact we shall prove the more general Thecrem 2.2, from which Theorem 2.1 easily follows. For the formulation of Theorem 2.2 we need the following definitions.

D e f i n i t i o n 2.2: Let $H \subset F^A$. Then B(H) ($B^1(H)$) is the set of all functions, which may be obtained from the functions of H by a finite number of substitutions and applications (one application) of the operator P.

D e f i n i t i o n 2.3: (Strong [8], p.468): The set of functions $F \subset F^A$ satisfies the axioms of "basic recursive function theory" (BRFT), if

(01) $U^A \subset F$, $C^A \subset F$,

(02) $(E\,\Psi \in F_4)(a,b,c,x)(\,\Psi(a,b,c,x) = \begin{cases} b, & \text{if } x = a \\ c, & \text{otherwise} \end{cases}$

(03) F is closed under substitution

(04) (existence of universal functions for all m > o):

$$(E\,\Phi_m \in F_{m+1})(F_m = \{ f \mid f(x_1,\dots,x_m) = \Phi_m(x,x_1,\dots,x_m) \text{ for } x \in A \}),$$

(05) (S-m-n-"Theorem" for all m,n > o):

$$(ES^m_n \in F_{m+1})(x,x_1,\dots,x_m, y_1,\dots,y_n)$$

$$(S^m_n(x,x_1,\dots,x_m) \text{ is defined and}$$

$$\Phi_n(S^m_n(x,x_1,\dots,x_m), y_1,\dots,y_n) = \Phi_{m+n}(x,x_1,\dots,x_m, y_1,\dots,y_n).$$

We now can formulate

T h e o r e m 2.2: Let g be an arbitrary function of F_1^A.

α, σ AF for A, and $G = \{g, \alpha, \sigma\} \cup U^A \cup C_1^A$. Then

1. B(G) satisfies the axioms of BRFT
2. $B(G) = B^1(G)$.

Of course, from Theorem 2.2 follows (for instance with $g = u_1^1$):

C o r o l l a r y 2.3: Let $\bar{G} = \{\alpha, \sigma\} \cup U^A \cup C_1^A$. Then

1. $B(\bar{G})$ satisfies the axioms of BRFT
2. $B(\bar{G}) = B^1(\bar{G})$.

For getting Theorem 2.1 from Theorem 2.2 we use the known Theorem 2.4.

T h e o r e m 2.4 (see, for instance, Friedmann [3], p.2.1, or Wagner [11], p.5): Let $F^* \subset F^N$, F^* satisfying the axioms of BRFT, and $s \in F^*$. Then $F^{rek} \subset F^*$.

For arbitrary AF α, σ for N $F_{B^1} = B^1(\{s, \alpha, \sigma\} \cup U^N \cup C_1^N)$ satisfies the axioms of BRFT (Theorem 2.2 with g = s and A = N). Obviously, $s \in F_{B^1}$. Thus, applying Theorem 2.4, $F^{rek} \subset F_{B^1}$. As the functions c_y^1 themselves may be obtained from s and c_o^1 by substitutions only $(c_y^1(x) = \underbrace{s(s(\ldots s(}_{y \text{ times}}c_o^1(x))\ldots))$, Theorem 2.1 is proven, provided that Theorem 2.2 is true.

The operator P can be reduced to substitution, primitive recursion and application of the operator μ :

$$\phi^P(\eta) = \phi^*(\eta, \mu_t(\phi^*(\eta, t)_{\bar{o},\bar{o}} = \bar{o})), \text{ where}$$

$$\phi^*(\eta, o) = \eta, \quad \phi^*(\eta, t+1) = \phi(\phi^*(\eta, t))$$

Thus, using Definition 1.1 of [2], p.41, we know also:

C o r o l l a r y 2.5: Let α, σ be AF for N, where $\alpha, \sigma \in F^{rek}$. Then

$B^1(\{\alpha, \sigma, s, c_o^1\} \cup U^N) \subset F^{rek}$.

Together with Theorem 2.1 this leads to

C o r o l l a r y 2.6: Let α, σ be AF for N, where $\alpha, \sigma \in F^{rek}$. Then $B^1(\{\alpha, \sigma, s, c_o^1\} \cup U^N) = F^{rek}$.

Finally, there remains the proof of Theorem 2.2, which consumes the rest of section 2. At first, we have to check the axioms (01),...,(05) (Definition 2.3) for B(G).

(01) $U^A \subset B(G)$ because the definition of G. Further, $c_y^n(x_1,\dots,x_n) = c_y^1(u_1^n(x_1,\dots,x_n)) \in B(G)$ for arbitrary $n \in N$ and $y \in A$.

(02) It is easy to prove (Lemma 1.8) that
$\Psi(a,b,c,x) = \alpha(\alpha(a, \sigma(\varepsilon, x, \bar{1})), [\varepsilon, \varepsilon, c, \bar{1}, b])$
satisfies axiom (02).

In the following we write $(a{=}x \to b, \text{else} \to c)$ instead of $\Psi(a,b,c,x)$ and $(a_1 = x_1 \to b_1, \dots, a_n = x_n \to b_n, \text{else} \to c)$ instead of $(a_1 = x_1 \to b_1, \text{else} \to (a_2 = x_2 \to b_2, \text{else} \to (\dots(a_n = x_n \to b_n, \text{else} \to c)\dots)))$. One easily checks

L e m m a 2.7: The (3n+1)-ary functions $(a_1 = x_1 \to b_1,\dots,a_n = x_n \to b_n, \text{else} \to c)$ can be obtained from the functions of G by a finite number of substitutions.

(03) is satisfied by definition.