# ON A BASIS FOR RECURSIVELY ENUMERABLE SETS

MATHEMATICS

1966

SovietRxiv

## Abstract

## Full Text

UDC 519.513

*MATHEMATICS*

V. A. NEPOMNYASHCHII

# ON A BASIS FOR RECURSIVELY ENUMERABLE SETS

*(Presented by Academician P. S. Novikov on 11 I 1966)*

**1. Introduction.** In Kleene normal form $f(x) = U[\mu y T(\varepsilon, x, y)]$, the function $U$ and the predicate $T$ are primitive recursive. Accordingly, every recursively enumerable set is enumerated by a suitable primitive recursive function. In the work of A. V. Kuznetsov ([1]), the possibility is established of certain related primitive-recursive constructions connected with functions of large range. Many authors have considered narrower classes of predicates and functions suitable for Kleene normal form, for enumeration of recursively enumerable sets, etc. On the one hand, the complexity criteria for these classes were based on the classification of the computation schemes used for recursive functions. It was found that the predicate $T(\varepsilon, x, y)$ can be taken to be elementary (in Kalmar' s sense), rudimentary, and $s$-rudimentary ([2]). On the other hand, parameters of Turing machines were considered as complexity criteria. In ([3]) it is shown that the class of predicates computable on a Turing machine with logarithmic slowdown is the smallest, with respect to computation time, among the classes suitable for Kleene normal form. In ([4]) it is established that the predicate $T(\varepsilon, x, y)$ can be taken to be computable on so-called marked Turing machines (see the definition below).

We introduce the class of predicates $\mathbf{R}_s^{\log}$; we shall show that it is narrower than all the classes listed above; and we shall establish the suitability of $\mathbf{R}_s^{\log}$ for Kleene normal form and related constructions.

**2. Basic concepts.** Let us recall some definitions from ([2–4]). Let $A_0$ be the set of all words in the alphabet $\{1, 2\}$. We establish a one-to-one correspondence between $A_0$ and the set of natural numbers as follows: to the word $a_\nu a_{\nu-1} \ldots a_1 a_0 \in A_0$ there corresponds the number $\sum_{i=0}^{\nu} a_i \cdot 2^i$. By $C(x, y, z)$ we denote the concatenation predicate $xy = z$, where $xy$ means the word obtained by appending the word $y$ to the word $x$. By $xPy$ we denote the predicate "$x$ is a subword of $y$" (i.e., there exist such words $z_1, z_2$, possibly empty, that $y = z_1 x z_2$). The class of rudimentary predicates ($\mathbf{R}$) is the smallest class containing the concatenation predicate $C(x, y, z)$ and closed under: 1) the operations of the algebra of logic; 2) explicit transformations (permutation of arguments, identification of them, replacement of variables by constants, addition of fictitious variables);

3) adjunction of bounded quantifiers. If instead of 3) one adjoins bounded quantifiers of the form*

$$(\exists z)R(x_1,\dots,x_n,z) \overset{df}{=} (\exists z)_{z\le y}[zPy\ \&\ R(x_1,\dots,x_n,z)],$$

$$(\forall z)R(x_1,\dots,x_n,z) \overset{df}{=} (\forall z)_{z\le y}[zPy \to R(x_1,\dots,x_n,z)],$$

then we obtain the class of $s$-rudimentary predicates ($\mathbf{R}_s$).

* $\overset{d}{f}=$ denotes equality by definition.

We shall consider functions mapping $N=\{1,2,\dots\}$ into $N$. We shall say that a function $f(x_1,\dots,x_n)$ is computable on a Turing machine with logarithmic slowdown if there exists a Turing machine computing $f(x_1,\dots,x_n)$ in $\tau$ steps with $\tau \le Cl\log_2 l$, where $C=$ const, and $l$ is the length of the record of the argument in sequential code (see (3)). By **Log** we shall denote the class of predicates computable on a Turing machine with logarithmic slowdown.

We shall say that a predicate $\Omega(x_1,\dots,x_n)$ is computable on a Turing machine with marks if there exists a Turing machine $\mathfrak{N}$ computing $\Omega(x_1,\dots,x_n)$ with the following restrictions: 1) the head of $\mathfrak{N}$ does not go outside the record of the argument in sequential code (see (3)); 2) for every cell $\nu$ of the tape of $\mathfrak{N}$ the following condition is satisfied: if at the initial moment of time $\nu$ contains the symbol $\nu$, then at every subsequent moment of time $\nu$ contains either $\nu$ or the symbol $'$ (a mark); 3) there exists a number $k$ such that at every moment of time on the tape of $\mathfrak{N}$ no more than $k$ cells are occupied by the symbol $'$. By **M** we shall denote the class of predicates computable on a machine with marks. Let us also define two more classes of predicates:

$$\mathbf{R}_s^{\log}=\mathbf{R}_s\cap\mathbf{Log};\qquad \mathbf{R}^{\log}=\mathbf{R}\cap\mathbf{Log}.$$

## 3. Comparison of classes of predicates.

**Theorem 1.**

I. $\mathbf{R}_s^{\log}\subset\mathbf{R}_s$.

II. $\mathbf{R}_s^{\log}\subset\mathbf{R}^{\log}$.*

III. $\mathbf{R}_s\subset\mathbf{M}$.**

**Remark.** It follows from Theorem 1 that the class of predicates $\mathbf{R}_s^{\log}$ is the narrowest among all the classes mentioned.

In the proof of Theorem 1 an essential role is played by

**Lemma 1.** *Let a predicate $\pi(x, y)$ have the following properties: 1) $\pi(x, y) \in \mathbf{R}_s$; 2) $\pi(x, y)$ is true only on such pairs $\langle x, y\rangle$ that $x, y$ are words in the alphabet $\{2\}$; 3) for every $x$ there exist in the truth set of the predicate $\pi(x, y)$ no more than a finite number of pairs $\langle x, y_i\rangle$ $(i = 1, \dots, m)$.*

*Then there exists a constant $K_0$ such that for every pair $\langle x, y\rangle$ from the truth set of $\pi(x, y)$ the following is valid: the ratio of the length of the word $y$ to the length of the word $x$ does not exceed $K_0$.*

## 4. Functions, their graphs and sets of values.

According to (1), a general-recursive function that is not primitive-recursive will be called complex-recursive. It is known that from the "simplicity" of the set of values of a function and of its graph there does not follow, generally speaking, the "simplicity" of the function itself. For example (see (1)), a monotonically increasing function with a primitive-recursive set of values and graph may be complex-recursive. A similar phenomenon occurs for functions of large range. For a function of large range $g(\xi)$ (i.e., one taking each value from $N = \{1, 2, 3, \dots\}$ an infinite number of times) let us define the function $g^-(x, y)$ as that $z$ for which $g(z) = y$, and moreover the function $g(\xi)$ takes the value $y$ on arguments not exceeding $z$ exactly $x$ times. It is known (see (1)) that there exist primitive-recursive functions of large range $g(\xi)$ for which $g^-(x, y)$ are complex-recursive.

Let $\mathbf{R}_s^{\mathrm{Log}}$ denote the class of functions computable on a Turing machine with logarithmic slowdown and such that their graphs belong to the class $\mathbf{R}_s^{\log}$.

**Theorem 2.** I. *There exists a monotonically increasing complex-recursive function such that its graph and set of values belong to $\mathbf{R}_s^{\log}$.*

---

* II refines the result $\mathbf{R}_s \subset \mathbf{R}$, announced in (2) (p. 92).
** III refines the result $\mathbf{R}_s \subseteq \mathbf{M}$, announced in (4).

II. There exists a function of large range $g(\xi)$ in $\mathbf{R}_s^{\mathrm{Log}}$ such that $g^-(x, y)$ is a complex-recursive function.

**Remark.** From Theorems 1, 2 and Lemma 1 it follows that the class $\mathbf{R}_s^{\mathrm{Log}}$ is narrower than the class of functions computable on a Turing machine with logarithmic slowdown, and also than the class of functions with graphs in $\mathbf{R}_s^{\log}$.

The remarks to Theorems 1 and 2 show that $\mathbf{R}_s^{\log}$ and $\mathbf{R}_s^{\mathrm{Log}}$ are rather narrow classes. Nevertheless, in them it is possible to realize a number of constructions that were previously justified for broader classes.

## 5. Enumeration of sets and explicit representation of functions

**Theorem 3.** I. Every recursively enumerable set is enumerable by a function from $\mathbf{R}_s^{\mathrm{Log}}$.

II. Every partial recursive function $f(x)$ is representable in the form

$$f(x) = U[\mu y T(\varepsilon, x, y)],$$

where $U(x)$ belongs to $\mathbf{R}_s^{\mathrm{Log}}$ and does not depend on $f(x)$, $T(\varepsilon, x, y)$ belongs to $\mathbf{R}_s^{\log}$ and does not depend on $f(x)$, while $\varepsilon$ depends on $f(x)$.

III. Every general recursive function $\varphi(x)$ is representable in the form

$$g_1(g^-(1, x1)),$$

where $g_1(\xi)$ is a function of large range from the class $\mathbf{R}_s^{\log}$, not depending on $\varphi(x)$, and $g(\xi)$ is a function of large range from $\mathbf{R}^{s\mathrm{Log}}$, depending on $\varphi(x)$.

IV. Every general recursive function is representable as the difference of two such functions whose graphs and ranges belong to $\mathbf{R}_s^{\log}$.

Let a Turing machine $\mathfrak{A}$ compute the partial recursive function $f(x)$. Suppose, further, that there is some coding method that makes it possible to associate with any finite sequence of configurations of $\mathfrak{A}$ a word in the alphabet $\{1, 2\}$. In the case where $\mathfrak{A}$ halts, by the protocol of the Turing machine $\mathfrak{A}$ we shall mean the sequence of its configurations from the moment of start-up to halting. Denote by $\Pi_{\mathfrak{A}}(x)$ the code of the protocol of $\mathfrak{A}$ under the chosen coding method, if $f(x)$ is defined. If $f(x)$ is not defined, then $\Pi_{\mathfrak{A}}(x)$ is also not defined.

In the proof of Theorems 2 and 3 the fundamental role is played by

**Lemma 2 (on coding).** For every partial recursive function $f(x)$ there exists a Turing machine $\mathfrak{A}$, computing $f(x)$, such that, under a suitable coding method for the protocol of $\mathfrak{A}$, the following condition is satisfied: the predicates 1) $y = \Pi_{\mathfrak{A}}(x)$; 2) $(\exists x)(y = \Pi_{\mathfrak{A}}(x))$; 3) $(\exists x)(y = \Pi_{\mathfrak{A}}(x) \& z = f(x))$ belong to the class $\mathbf{R}_s^{\log}$.

In conclusion the author expresses gratitude to B. A. Trakhtenbrot for posing the problem and for his attention to the work.

Novosibirsk State
University

Received
7 I 1966

## References

1. A. V. Kuznetsov, DAN, **71**, No. 2, 233 (1950).

2. R. M. Smullyan, *Theory of Formal Systems*, Princeton, 1963.

3. B. A. Trakhtenbrot, *Algebra and Logic*, Seminar, II, 4, 1964, p. 33.

4. D. L. Kreider, R. W. Ritchie, Notices Am. Math. Soc., **12**, No. 6, 692 (1965).

*Note: Figure translations are in progress. See original paper for figures.*