RUDIMENTARY PREDICATES AND TURING COMPUTATIONS

MATHEMATICS

1970

SovietRxiv

Abstract

Full Text

UDC 519.513

MATHEMATICS

V. A. NEPOMNYASHCHII

RUDIMENTARY PREDICATES AND TURING COMPUTATIONS

(Presented by Academician P. S. Novikov on 8 IV 1970)

1. Introduction. In important constructions of the theory of algorithms, such as Kleene normal form, protocols of Turing machines, etc., instead of the class of primitive-recursive predicates one often uses narrower classes. Such classes are obtained both by restrictions on recursive schemes for computing predicates (the classes of bounded-arithmetic (Ap) predicates ([1]), constructively arithmetic (K), rudimentary (R), s-rudimentary (R_s) predicates ([2])), and by restrictions on the parameters of Turing computations (the classes of predicates computable on Turing machines with logarithmic slowdown (Log) ([3]), with tags (M) ([4]), in real time (T) ([5])). In ([6]) it is indicated that, for analogous purposes, one may use the class $R_s^{\log}$, obtained by restrictions of both types. For related constructions in ([7]) (§ 33) the class of context-free languages (L_{kc}) is used. From ([2]) it follows that $Ap \supseteq K \supseteq R \supseteq R_s$. In ([2]), p. 92, the following result of Bennett was announced: $K = R \supset R_s$. In ([6]) it is indicated that $R_s^{\log} \subset R_s \subset M$. From ([2,3,8]) it follows that all the above-mentioned classes of predicates (except the class of primitive-recursive predicates) belong to the class of predicates computable on Turing machines with linear space, and also that K belongs to the smallest class of the Grzegorczyk hierarchy ([9]). In connection with the facts listed, a number of questions naturally arise. We shall note those of them which are investigated in the present work.

1. Since the classes R and Log are obtained by restrictions of different types and $R \not\subseteq Log$, the hypothesis arises that they are incomparable (i.e. $Log \not\subseteq R$). Is this true? An analogous question for the cases when Log is replaced by M, T, or L_{kc}.
2. The class Ap is generated by more powerful operations than R. Is Ap broader than R?
3. Is it true that $(Log \cap M \cap T \cap L_{kc}) \subseteq R_s$?
4. All predicates from R_s indicated in ([2]) are defined in nonarithmetic terms (for example, "x is a subword of y," etc.). Do the simplest arithmetic predicates belong to R_s (for example, $y = x + 1,\ x \leq y$)?

It turns out that the answers to all questions 1-4 are negative. In obtaining the answers to questions 1, 2, the main role is played by Theorem 1. It gives a sufficient condition for the rudimentarity of predicates in terms of restrictions on the parameters of computations of predicates on two-tape Turing machines from ([10]).

2. Two-tape Turing machines. We consider Turing machines $\mathfrak{M}$ with an input and a working tape ([10,11]). The program of $\mathfrak{M}$ consists of commands of the form

$$q_i m_j m_k \to q_i' m_k' s_\rho' s_\mu'',$$

which are interpreted as follows: if $\mathfrak{M}$ is in state q_i and reads the symbol m_j on the input tape and the symbol m_k on the working tape, then it passes into state q_i', writes m_k' on the working tape and shifts the input (working) head in the direction s_ρ' (s_μ'') by one cell: $s_\rho', s_\mu'' = \Pi$ (to the right),   (to the left),   (no shift). $\mathfrak{M}$ is, generally speaking, a nondeterministic machine, since its program may contain two commands with the same left-hand side. Different applications of these commands yield different processing of the input word. During any processing of an input word the input head does not go beyond its bounds. $\mathfrak{M}$ accepts the input word p if there exists

such a processing p, under which $\mathfrak{M}$ passes into the distinguished state, $\mathfrak{M}$ computes the predicate $\omega(x_1, \dots, x_r)$ *, if $\mathfrak{M}$ admits all and only such words

$$p = *x_1 * \dots * x_r*$$

for which $\omega(x_1, \dots, x_r)$ is true.

For an input word p, denote by $T_{\mathfrak{M}}(p)$ $(L_{\mathfrak{M}}(p))$ the maximum number of steps (of scanned cells of the working tape) of $\mathfrak{M}$ under any processing p **. Let

$$L_{\mathfrak{M}}(n) = \max_{|p|=n} L_{\mathfrak{M}}(p) \ \ ***, \qquad T_{\mathfrak{M}}(n) = \max_{|p|=n} T_{\mathfrak{M}}(p).$$

Theorem 1. *Suppose there exist a nondeterministic two-tape Turing machine $\mathfrak{M}$ and integers $\alpha > 1$, $\beta > 1$, C such that the predicate $\omega(x_1, \dots, x_r)$ is computable on the machine $\mathfrak{M}$ with*

$$T_{\mathfrak{M}}(n) \le n^{\alpha}, \qquad L_{\mathfrak{M}}(n) \le n^{1-1/\beta}$$

(for all $n \ge C$). Then $\omega(x_1, \dots, x_r)$ is a rudimentary predicate.

From this theorem it follows that

Corollary 1. *The class of predicates computable on nondeterministic two-tape Turing machines $\mathfrak{M}$ with*

$$L_{\mathfrak{M}}(n) \le C \log_2 n$$

(where C is a constant depending on the predicate) is contained in the class **R**.

3. Comparison of the class R with other classes. By the method proposed by G. S. Tseitin and in ([11]), the following is proved.

Theorem 2. *Suppose there exist a nondeterministic two-tape Turing machine $\mathfrak{M}$ and an integer $\alpha > 1$ such that the predicate $\omega(x_1, \dots, x_r)$ is computable on the machine $\mathfrak{M}$ with*

$$T_{\mathfrak{M}}(n) \leq n^{2-1/\alpha}$$

(starting from some n). Then there exists a nondeterministic two-tape Turing machine $\mathfrak{M}'$ computing the predicate $\omega(x_1, \dots, x_r)$ with

$$T_{\mathfrak{M}'}(n) \leq n^4 \quad \text{and} \quad L_{\mathfrak{M}'}(n) \leq n^{1-1/3\alpha}$$

(starting from some n).

From Theorems 1, 2 it follows that

Corollary 2. **** **Log** $\subset$ **R**.

Corollary 3. **T** $\subset$ **R**.

From Theorems 1, 2 ((7), §19, (13)) it follows that

Corollary 4. **M** $\subseteq$ **R**.

Corollary 5. $\mathbf{L}_{\mathrm{kc}} \subset \mathbf{R}$ *****.

Remark. Instead of Corollary 2, from the same references one obtains the following, stronger fact: the class of predicates computable on deterministic one-tape Turing machines $\mathfrak{M}$ with

$$T_{\mathfrak{M}}(n) \leq Cn^{2-1/\alpha}$$

(α is an integer > 1, C is a constant, α and C depend on the predicate) is strictly contained in **R**. Analogously for Corollary 3.

4. Bounded-arithmetic predicates. **Ar** is the smallest class of predicates containing $x = y$ and closed under: 1) the operations of the algebra of logic; 2) prefixing bounded quantifiers; 3) substitution, in place of variables, of polynomials with natural coefficients (1). The definition of **K** is obtained from the definition of **R** (6), if instead of the predicate $C(x, y, z)$ one takes the predicates $x + y = z$ and $x \cdot y = z$. From Corollary 1 it follows that $x + y = z$ and $x \cdot y = z$ belong to **R**. Hence, and from Theorem 6 of Ch. IV (2), we have

Corollary 6 (Bennett). **K** $=$ **R**.

By prefixing quantifiers bounded by a polynomial $\pi(\bar{y})$ to a predicate

$$\Omega(\bar{x}, \bar{y}, z) \quad (\bar{x} = x_1, \dots, x_m;\ \bar{y} = y_1, \dots, y_n),$$

we mean the operations

$$(\exists z)(z \leq \pi(\bar{y})\ \&\ \Omega(\bar{x}, \bar{y}, z)), \qquad (\forall z)(z \leq \pi(\bar{y}) \to \Omega(\bar{x}, \bar{y}, z)).$$

From the definition of **Ar** the following is easily obtained.

Lemma. **Ar** *is the smallest class containing the predicates $x + y = z$ and $xy = z$ and closed under: 1) the operations of the algebra of logic; 2) linear transformations* *****; 3) prefixing quantifiers bounded by polynomials.*

---

* We do not distinguish between numerical and word (in the alphabet $\{1,2\}$) predicates, keeping in mind the known one-to-one correspondence between words and numbers (see, for example, ([6])).

** $\mathfrak{M}$ must halt under any processing p.

*** $|p|$ denotes the length of the word p.

**** For definitions of $\mathbf{R}_3$, $\mathbf{R}$, $\mathbf{Log}$, $\mathbf{M}$ see ([6]), $\mathbf{T}$ see ([5]).

***** $\mathbf{L}_{\text{kc}}$ is the class of context-free languages over the alphabet $\{1,2\}$. For the definition see ([7]).

****** For the definition see ([8]).

With the aid of this lemma and Theorem 1 one establishes

Theorem 3. *The classes of rudimentary and bounded-arithmetical predicates coincide.*

Remark. Thus the class $\mathbf{R}$ is closed under such powerful operations as the prefixing of quantifiers bounded by polynomials.

5. s-Rudimentary predicates. Denote by $s(x)$ the following predicate: "there exists an $n = 1, 2, \ldots$ such that $x = 1\ldots1\,2\ldots2$*."

$$\underbrace{1\ldots1}_{n}\ \underbrace{2\ldots2}_{n}$$

Theorem 4. $s(x)$ *does not belong to the class of s-rudimentary predicates.*

From this follow

Corollary 7. $\log \cap M \cap T \cap L$ does not enter the class $\mathbf{R}_s$.

Corollary 8. The predicates $x + 1 = y$ and $x \leq y$ do not belong to the class $\mathbf{R}_s$.

Remark. Since the predicate $x \leq y$ belongs to the class $\mathbf{R}$ ([2]), Bennett' s result (announced in ([2])) that $\mathbf{R}_s$ is a proper subclass of $\mathbf{R}$ follows from Corollary 8.

Note added in proof. After obtaining the present results, we learned that Corollary 5 is also contained in ([14]).

Computing Center Siberian Branch of the Academy of Sciences of the USSR Novosibirsk

Received 19 III 1970

CITED LITERATURE

[1] A. V. **Kuznetsov**, On the theorem on canonical form for ordinal-recursive functions, Appendix to the book by R. L. Goodstein, *Mathematical Logic*, Moscow, 1961, p. 149.
[2] R. M. **Smullyan**, *Theory of Formal Systems*, Princeton, 1963.
[3] B. A. **Trakhtenbrot**, Turing computations with a logarithmic delay, *Algebra and Logic*, Seminar III, 4, 1964, p. 33.
[4] D. L. **Kreider**, R. W. **Ritchie**, *Zs. math. logic u. Grundl. d. math.*, **12**, No. 3, 243 (1966).
[5] B. A. **Nepomnyashchii**, On certain automata capable of computing a basis for recursively enumerable sets, *Algebra and Logic*, Seminar V, 5, 1966, p. 69.
[6] B. A. **Nepomnyashchii**, *DAN*, **170**, No. 6, 1262 (1966).
[7] A. V. **Gladkii**, Lectures on mathematical linguistics for students of NGU, Novosibirsk, 1966.
[8] R. W. **Ritchie**, *Trans. Am. Math. Soc.*, **106**, No. 1, 139 (1963).
[9] A. **Grzegorczyk**, Some Classes of Recursive Functions, *Rozprawy matematyczne*, 4, 1953.
[10] J. **Hartmanis**, P. M. **Lewis II**, R. E. **Stearns**, *Proc. IFIP Congress* 65, 1, 31 (1965).
[11] J. E. **Hopcroft**, J. D. **Ullman**, *J. Assoc. Comp. Math.*, **15**, No. 3, 414 (1968).
[12] R. **Freivald**, Complexity of recognizing symmetry on Turing machines with input, *Algebra and Logic*, Seminar IV, 1, 1965, p. 47.
[13] V. A. **Nepomnyashchii**, Computation on Turing machines with marks, Abstracts of scientific communications of the IX All-Union Algebraic Colloquium, Gomel, 1968, p. 145.
[14] N. D. **Jones**, *Math. Syst. Theory*, **3**, No. 2, 102 (1969).

$$^{*}\quad l^n = l \dots l, \quad |l^n| = n \quad (l = 1, 2).$$

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