CYBERNETICS AND THE THEORY OF CONTROL

==================================

1963

SovietRxiv

Abstract

Full Text

CYBERNETICS AND THE THEORY OF CONTROL

N. V. BELYAKIN

PLACEMENT OF INTERMEDIATE INFORMATION IN COMPUTATIONS ON NONERASING TURING MACHINES

(Presented by Academician P. S. Novikov on 5 IV 1963)

1°. In note ([1]) a class $\mathfrak{M}(A, B)$ of Turing machines of the following kind was introduced. Let A, B be two disjoint finite alphabets. Each machine in $\mathfrak{M}(A, B)$ has external alphabet $A \cup B$ and a tape infinite in one direction. In this case the machine either does not change the scanned letter at all, or may replace a letter from the alphabet A by a letter from the alphabet B. We shall call such machines **nonerasing**. They are a generalization of the machines considered in ([2,3]).

In ([1]) it was shown how these machines can be used to compute operators that transform sequences $\{x(n)\}$ of letters of the alphabet A into sequences of letters of the alphabet B–$\{z(n)\}$. Namely, suppose that at each step the tape is extended by one cell, with the initial contents of the n-th cell being $x(n)$. Then the letters from the alphabet B written by the machine form, generally speaking, a sequence $\{z(n)\}$ not everywhere defined, so that each machine computes some partial recursive operator (in particular, it may turn out to be general recursive).

It was shown that the operators computable in this way form a narrow class, not even containing all general recursive operators. The reason for this may be explained heuristically as follows.

The machine cannot erase the results of intermediate computations; therefore every letter from the alphabet B written by it is interpreted as an element of the output sequence. In this connection, an organization of computations is of interest in which a certain part of the tape is used for placing intermediate results and contains neither input nor output information. This question is the subject of the present article.

2°. We shall assume that the set of tape cells is divided into two recursive subsets C, C', which we shall call the **draft** and the **clean copy**, respectively. Let us distinguish in the alphabet A one symbol, which we shall interpret as the blank sign. Suppose a machine from $\mathfrak{M}(A, B)$ is given, and the sequence $\{x(n)\}$, $x(n) \in A$ $(n = 1, 2, ...)$, is supplied to the cells of the clean copy, while the initial contents of the draft cells are the blank sign. Then the sequence of letters of the alphabet B appearing in the clean copy during the operation of the machine will be interpreted as the output sequence corresponding to the input sequence $\{x(n)\}$. The letters written by the machine in the draft are not taken into account. In this case we shall say that the machine **computes an operator with draft** C.

Remark. Usually, in the external alphabet of a Turing machine, one of the signs is distinguished and called blank. Its special role among the other letters appears when the machine is used to process finite words, in particular in computing functions. It is precisely in this case that at any moment of the machine' s operation the entire tape, except for a finite segment of it, is occupied by the blank sign.

filled with the blank symbol. In the problem considered in the present article, this asymmetry does not occur, since any input symbol may be supplied infinitely many times. Therefore the name "blank symbol" for the chosen letter from A is due only to considerations of clarity.

Definition. A recursive set C is called a **universal scratchpad** over the alphabets A, B if, for every general recursive operator with input alphabet A and output alphabet B, there exists a machine from $\mathfrak{M}(A, B)$ that computes this operator with scratchpad C.

The existence of universal scratchpads follows, for example, from the following lemma.

Lemma. The set of elements of a periodic sequence (having an infinite complement) is a universal scratchpad.

The proof of the lemma can be obtained by modifying the constructions used in work ([2]).

The aim of this article is to establish a necessary and sufficient condition for a recursive set to be a universal scratchpad.

3°. In order to formulate the main result of the article, we introduce one class of sets that is also of independent interest. These are sets that can be enumerated in a very simple interpretation by nonerasing Turing machines.

Let a nonerasing machine $M \in \mathfrak{M}(A, B)$ be given, and suppose that it begins to operate on a blank tape, scanning the first cell in its initial state. Associate with M the set $\psi(M)$ of the numbers of those cells into which it will write letters from B. Obviously, $\psi(M)$ is recursively enumerable; if it is infinite, then we shall call it a **T-set over the alphabet** B. It is clear that not every recursive set is a T-set. This follows at least from the fact that the distance between neighboring elements of a T-set does not exceed the number of states of the machine enumerating it. Further, trivially, the set of elements of any periodic sequence is a T-set. Finally, we note that one can construct an example of a recursive T-set that contains no periodic sequence, and also an example of a nonrecursive T-set. It remains an open question whether the class of T-sets expands when the alphabet B is enlarged.

We now give (without proof) a characterization of the class of universal scratchpads.

Theorem 1. A recursive set C with infinite complement is a universal scratchpad over the alphabets A, B if and only if C is a T-set over the alphabet B.

Remark. The condition of Theorem 1 is no longer sufficient for the possibility of computing, with the given scratchpad, every partial recursive operator over the alphabets A, B. Indeed, one can construct such a partial recursive operator (with input alphabet A and output alphabet B) that is not computable by any machine from $\mathfrak{M}(A, B)$, if the set of even numbers is taken as the scratchpad.

4°. Let us now consider a somewhat simpler problem. Namely, suppose that machines from $\mathfrak{M}(A', B')$ are used for computing general recursive operators over the alphabets A, B, where $A \subset A'$ or $B \subset B'$. In this case one can also introduce the notion of universality of a scratchpad and pose the question of characterizing the class of universal scratchpads.

It turns out that the result obtained is essentially different when $A = A'$ and when $A \subset A'$. Let A' contain at least one letter not belonging to A. Choose precisely this letter as the blank symbol with which the scratchpad will be filled when computing operators. Since only letters from A are then supplied to the clean copy, an "explicit" distinction is obtained between scratchpad and clean-copy cells.

Using this circumstance greatly simplifies the organization of computations; as a result, every recursive set with an infinite complement proves to be a universal scratchpad such that, for some $0 < a < 1$, every segment of the tape of length $n > N(a)$ contains at least an scratchpad cells. If $A = A'$, $B \subset B'$, then one obtains a result analogous to Theorem 1. In this case the necessary and sufficient condition for universality of the scratchpad is that it be a T-set over the alphabet B'.

5°. In conclusion let us consider the class $\mathfrak{M}'_C(A, B)$ of machines with alphabet $A \cup B$, which satisfy less stringent restrictions on erasure. Namely, suppose that a recursive set C (the scratchpad) is distinguished on the tape, and in the scratchpad cells any replacement of letters is permitted, while in the remaining cells, as before, only replacement of letters from A by letters from B is allowed. Analogously to what was done above, these machines can be used for computing operators. However, in the present case there is no need to distinguish partial recursive and general recursive operators, as will be seen below. Therefore we shall call a universal scratchpad over $\mathfrak{M}'_C(A, B)$ any recursive set C such that in the class $\mathfrak{M}'_C(A, B)$ every partial recursive operator with input alphabet A and output alphabet B is computable.

In order to characterize the class of universal scratchpads, we introduce sets representing a generalization of T-sets to the case of arbitrary machines. Let M be any Turing machine, in whose alphabet we shall distinguish the blank symbol and the significant symbols, and let M begin work by scanning, in the initial state, the first cell of a blank tape. Associate with the machine M the set of numbers of all those cells in which it ever writes a significant symbol. Denote by Π_n the class of sets associated in this way with all possible Turing machines with n significant symbols. Obviously,

$$\Pi_1 \subseteq \Pi_2 \subseteq \cdots \subseteq \Pi_n \subseteq \cdots.$$

It can be shown that for infinitely many n one has

$$\Pi_n \subset \Pi_{n+1}.$$

The classes Π_n do not comprise all recursive sets, although they do contain some nonrecursive (recursively enumerable) sets.

If a scratchpad C belongs to none of the Π_n, then for any choice of alphabets A, B (each of which contains at least two letters) one can construct a general recursive operator with input alphabet A and output alphabet B that is not computable in the class $\mathfrak{M}'_C(A, B)$. An analogous result is obtained if C belongs to Π_n and does not belong to Π_{n-1}, while the alphabet B contains fewer than n letters.

The main result of this subsection is

Theorem 2. *If $C \in \Pi_n$ and the alphabet B contains more than n letters, then, for an arbitrary choice of the alphabet A, the set C is a universal scratchpad over $\mathfrak{M}'_C(A, B)$.*

It remains an open question whether the scratchpad C is universal in the case when the cardinality of the alphabet B is equal to the least n such that $C \in \Pi_n$.

Institute of Mathematics with Computing Center
Siberian Branch of the Academy of Sciences of the USSR

Received
23 III 1963

REFERENCES

1. N. V. Belyakin, DAN, **148**, No. 1 (1962).
2. Hao Wang, J. Assoc. Comp. Mach., No. 1 (1957).
3. W. Oberschelp, Arch. math. Log. und Grundlagenforsch., **4**, Nos. 1–2 (1958).

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