The formalized language described in the present paper was originally conceived as a metalanguage for the formal description of the semantics of algorithmic languages, and was therefore given the name meta-algorithmic. The first version of the meta-algorithmic language was described in [1]. The language has been considerably improved in what follows, although the fundamental ideas remain the same. In this paper a formal description of the meta-algorithmic language in its final form is given; however, it is first necessary to discuss the purpose of the creation of this language and indicate the way to achieve it.

COMMUNICATION BETWEEN MAN AND MACHINE AND THE FORMALIZATION OF SEMANTICS

A metalanguage for describing the semantics of algorithmic languages is an algorithmic language [1]. The semantics of any other algorithmic language can be described by any universal algorithmic language; hence the choice of any particular algorithmic language as a metalanguage is, in principle, a question of convenience (although in practice "inconvenience" often means "impossibility"). Since we are dealing with the formal semantics of descriptions intended for use by man, the algorithmic metalanguage which we select must be suitable for a human being. But since an algorithm is essentially an instruction for a machine, the problem of the metalanguage is a problem of communication between man and machine which is convenient not for the machine but for the human being.

Machine-independent algorithmic languages are widely used in modern programming (FORTRAN, ALGOL, COBOL, etc.) and are convenient for recording problems of a certain field, since their construction is based on the formalization of a number of concepts which are important and characteristic for a given special field. But we need a language oriented not to any specific concepts, but intended for the description of any languages and concepts (a metalanguage); hence such a language will be convenient for a human being only if it includes some extremely general but at the same time important features of human thought.

In searching for such features we turn to natural languages and their extension, the formalized languages of mathematics. The most important feature of these languages is the presence in them of a hierarchy of concepts, arising from the capacity of the brain, by abstracting from many concrete situations, to create linguistic models of reality, which are then used by specifying them suitably for any given situation. A natural language may be represented as a multi-stage pyramid constructed on the ground of sensory experience. In our (semantic) approach the elements of this pyramid must be considered to be morphemes, the minimal meaningful units of a language. Combined in chains the morphemes form linguistic objects (linguistic objects of (morphemes): words, groups of words, sentences. The linguistic objects on the lowest floors of the pyramid specify the most concrete concepts, close to sensory experience: "ill," "cold," "hare," "stone," etc., from which more abstract and more complex concepts are constructed, and based on the latter still more complex concepts, and so on; all these concepts are specified by linguistic objects. Somewhere in the middle stories of the pyramid are the concepts "north," "number," "work," "rank of colonel," and somewhere at the very top "alienation," "homozygotic," "bicompleteness." In natural language it is impossible to introduce an exact measure of the abstractness or complexity of a concept and distribute the corresponding linguistic elements among the stories; hence our pyramid must be understood as being only provisional and illustrative. However, the principle of the formation of complex and abstract concepts by composition and abstraction from simpler and more concrete objects undoubtedly forms the basis for the construction of languages.

We now take some linguistic objects, for example, a word, and ask the question: what does it mean to understand this word? It is obvious that the physical carrier of the word has no significance; only the connection of this word with other words, the "complex of sensations," and ultimately with the experience of the senses, is significant. Consequently, to understand a word means to know how to go in the opposite direction along the path of its construction. To understand an abstract concept means to know how to concretize it in every given situation; to understand a complex concept means to know how to reduce it to a number of simpler ones. Both the former and the latter mean the replacement of a linguistic object occupying a higher position in the linguistic pyramid by a number of objects occupying lower positions. We will call this operation concretization of the linguistic object. In some forms of activity we do not pursue it to complexes of sensations; however it is supposed that we know how to do this, otherwise the word has no meaning for us. Therefore, the semantics of a linguistic object defines the rule for concretizing it, and the semantics of the language as a whole defines the aggregate of concretization rules, which make it possible in a series of steps to reduce each linguistic object to some irreducible elementary objects ("complexes of sensations").

The scheme for defining the semantics of objects of a natural language which has been sketched is undoubtedly very much simplified; however without simplification any formalization would be impossible, and in fact
our problem consists of the formalization of semantic descriptions. Hence in the construction of a meta-algorithmic language this scheme must be taken as the basis. Before proceeding to the formal description of the meta-algorithmic language, we will describe its basic features briefly and informally.

To indicate the fact that a given object is subject to concretization, the concretization symbol $X$ is introduced into the meta-algorithmic language. For example, if $x$ is some (algebraic) variable, its value will be denoted by $X^x$ ("concretization of $x$"), so that if the value of $x$ is 7, sooner or later $X^x$ will be replaced by 7. Parentheses are used for imparting the necessary structure to expressions, in particular for isolating expressions belonging to a given concretization symbol. Expressions enclosed in parentheses and also some elementary expressions are called terms. Concretization rules are written as statements, which are separated from one another by the paragraph sign $\$ at the beginning of each statement. A statement has a left side, the "high level" term, and a right side, an equivalent expression of "lower level." For example, the concretization rule stating that the value of $x$ is 7 is written as the statement $\$X^x \sim 7$, where the substitution symbol $\sim$ means that $X^x$ must be replaced by 7. In this statement $X^x$ is the left side, and $\sim$ is the right side (but the right side does not always begin with the symbol $\sim$). For writing incompletely defined expressions in statements free variables are used. For example, $S_1$, $W_1$, $E_1$ (read: "the symbol 1, the term $x$, the expression $a"$).

In all natural languages there exists the possibility of introducing new terms and redefining old ones. It is obvious that it is necessary to provide for this in the meta-algorithmic language also, particularly since otherwise, in the method of recording the values of the variables which we have adopted (and this method is very convenient), we would not even be able to ascribe a new value to a variable. In the meta-algorithmic language it is permissible to record a new statement in the memory: for this the operation denoted by the symbol $\rightarrow$ is used. Turning to future statements it is necessary to distinguish the symbols $\$, $\|$, etc. occurring in them from the "real" $\$, $\|$, etc. Therefore, statements considered as expressions are written in a special transformed form ($\mu$-transformation), which is executed by using the symbol $'$ (prime).

The semantics of the meta-algorithmic language itself are determined by a meta-algorithmic machine which performs the concretization of an expression in accordance with an available set of statements. Consequently, the operation of this machine simulates the linguistic activity of a human being using a hierarchical system of concepts. Thus, the meta-algorithmic machine is "cleverer" (and, of course, more complex) than a Turing machine, which merely simulates the fact of linguistic activity itself (so to speak, the "linguisticness" of the activity). It may be considered as the next step along the path of approximating the machine to a human being, and hence communication with such a machine proceeds by a more "human" method and more conveniently for man than communication with a Turing machine. An algorithm (collection of propositions) in the meta-algorithmic language may be constructed as a hierarchy of concepts increasingly more complex and specialized.

Since the use of a hierarchy of concepts is a necessary (perhaps a fundamental) element of human thought, a meta-algorithmic language which permits the formalized consideration of a hierarchy of concepts may be considered to be useful for studying and simulating thought processes. Man-machine communication and the simulation of human thought are essentially two aspects of the same problem.

FORMAL DESCRIPTION OF THE META-ALGORITHMIC LANGUAGE

(In reading this section reference to the next section for examples and explanations is recommended.)

1.1. To describe a language means to define some set of objects, which we will call lingemes, and some operations connected with these objects. The first part of the description is called syntax, the second semantics.

1.2. The meta-algorithmic language is a one-dimensional symbolic language, that is, there exists a finite number of unseparable signs, and each lingeme is a string of signs, that is, a finite sequence of elements, each of which is identified with one of the signs.

A detailed description of the fundamental concepts of symbolic languages is contained in the article "Theory of algorithms" by A. A. Markov [2], where the terms letter and word have the same meanings as sign and string in our work.

We introduce a series of syntactic concepts which will be required both for defining lingemes and for describing semantics.

1.3. The syntax of one-dimensional symbolic languages is based on the concept of strings of signs (or simply strings) given above, and the concept of the union of several strings denoting the sequential attachment of strings one after another in the order in which they follow the word "union." A string not containing a single sign will be called empty.

Many syntactic concepts may be given simple recursive definitions; it is convenient to describe them by the normal Backus form [3]. In definitions 2.1–2.14 apart from 2.3, given in this form, the words in quotation marks denote syntactic concepts which we will use later as ordinary Russian words (without quotation marks). The sign $::=$ is read "is by definition"; the vertical stroke $|$ (read: "or") separates the various parts of a case of definition; a sequence of elements of the language and syntactic concepts implies the union of the corresponding strings; it is here supposed that neither quotation marks "$", nor the sign $::=$, nor $|$ occur among the signs of the meta-algorithmic language.

To denote briefly the individual objects of the meta-algorithmic language, we will use capital Latin letters, possibly with subscripts.

2.1. "sign" $::=$ "proper sign" | "improper sign"
2.2. "proper sign" $::=$ $<|$ $\|C|$ $|$ $|$ $\|S|$ $|$ $W|$ $E|$ $'$ $\sim|$ $\rightarrow|$ $\leftarrow|$