

THE STRUCTURE OF A SEMANTIC NEURAL NETWORK REALIZING MORPHOLOGICAL AND SYNTACTIC ANALYSIS OF A TEXT

D. E. Shuklin

UDC 519.7

A synchronized linear tree is considered as the structure of connections between neurons in a semantic neural network, which makes it possible to perform operations of morphological and syntactic analysis. Such a tree can be considered as a finite automaton. A dictionary entry is realized by a neural subautomaton. Wordforms of a dictionary entry are substates of such a subautomaton. An inflection consists of the transition of the subautomaton from one substate to another.

Keywords: *semantic neural network, synchronized linear tree, morphological analysis, syntactic analysis, trees and automata, dictionary entries and automata, programming and training of neural networks.*

The process of processing a text in a natural language can be divided into some levels, such as morphological analysis, syntactic analysis, and synthesis. At the level of morphological analysis, a text in a natural language is transformed from a nonformalized form into a formalized internal representation; at the level of syntactic analysis, data are processed in their internal representation and new data are derived on their basis that also have a formalized form, and, at the level of synthesis, a result is formed, which is represented in the natural language and is adequate to the corresponding internal formalized representation.

The level of analysis of a text can be divided into some successive operations, namely, the operations of morphological, syntactic, and semantic analysis. The morphological analysis consists of determination of individual words in the text and division of them into morphemes. The syntactic analysis of the text consists of determination of all the syntactic attributes and syntactic relations that feature these words and are necessary for semantic analysis [1]. Semantic analysis requires the presence of a simulation model of the outside world. This analysis consists of final formation of an internal formalized representation of the text by comparison of the facts presented in the text with knowledge obtained from the simulation model [2].

Let us consider the operations of morphological and syntactic analysis of a text in a natural language. The separation of individual words from a text represented in the electronic form presents no problem since words are separated in this case by the special "space character." For the Russian language, the problem of morphological and syntactic analysis can be solved with the help of data contained in the Grammatical Dictionary of the Russian Language [3]. This dictionary provides information on word-building, inflections, and determination of syntactic attributes of individual words.

The realization of a level of analysis of texts requires the use of some formal model of such an analysis. By the Gödel theorem, the realization of a formalism can be easier than the description of the formalism. Mentioning the Gödel theorem, von Neumann writes: "... In such a case, one can make something more quickly than he describes it, one can draw a scheme more quickly than he gives a general description of all its functions and all conceivable circumstances. Of prime importance is the understanding of the fact that a network consisting of formal neurons can perform any operations that can be described by words, and this fact greatly simplifies matters at low levels of complexity. But this need not be a simplification at high levels of complexity. It is quite possible that, at high levels of complexity, the converse statement of this [Gödel] theorem is

Kharkov State Technical University of Radioelectronics, Ministry of Education of Ukraine, Kharkov, Ukraine. Translated from *Kibernetika i Sistemnyi Analiz*, No. 5, pp. 172-179, September-October, 2001. Original article submitted June 19, 2001.

valuable, i.e., it simplifies matters since it guarantees the reverse situation, i.e., the entire logic can be expressed in terms of these constructions [formal neurons], and the prime statement can be incorrect.” [4]. It follows from this statement of Neumann that, possibly, in the capacity of formal representation, one should use realizations of formal mathematical models and not these models themselves. A formal neural network is a variant of such formal representation. Therefore, an effort can be made to use it to solve problems of morphological and semantic analysis of texts and the problem of analysis of inflections.

In [4], the functioning of the formal McCulloch-Pitts neural network is described. In the McCulloch-Pitts neural network, individual neurons are the logic operations AND, OR, and NOT. Depending on the result of execution of a logic operation, a neuron resides at an *excited* state or at a *quiescent* state, which correspond to the logic values *true* and *false*. “... The operating time of each neuron guarantees efficient and constructive character of a logic system obtained in this case.” [4]. Since the number of neurons in a neural network is finite and the number of states of a neuron is finite, such a network is an automaton with a finite number of states. In the McCulloch-Pitts network, a neuron can be only at two logic states and provides the realization of only functions of the algebra of logic. A natural language manipulates fuzzy and incomplete concepts. Therefore, the McCulloch-Pitts neural network does not provide the processing of fuzzy concepts of a text in a natural language without additional efforts.

In order to solve problems of morphological and syntactic analysis of texts and the problem of analysis of inflections, we will use a semantic neural network [5] whose properties are close to those of the formal McCulloch-Pitts neural network. The distinction of the semantic neural network from the latter one lies in the fact that the logic operations of Boolean algebra are performed in the McCulloch-Pitts network, and the operations of fuzzy logic are performed in the semantic neural network [5]. In fuzzy logic, for determination of the truth degree of a statement, the confidence factor is used, i.e., a number belonging to some interval, for example, to [0, 1]. The maximum value of this interval is usually interpreted as the complete confidence in the occurrence of an event, and its minimum value is interpreted as the complete confidence in its absence. In contrast to probability theory, the confidence factor expresses a subjective confidence in the occurrence of an event and is statistically meaningless [6]. In the semantic neural network, neurons correspond to elementary concepts of a natural language and process discrete gradient values. Each neuron of such a network has a finite number of states. Hence, the semantic neural network being used can be considered as a finite automaton.

In the capacity of the structure of a semantic neural network performing morphological and syntactic analysis, we choose a clocked linear tree [7]. Let us roughly estimate the volume of the neural network required for realization of a Russian grammatical dictionary. The dictionary of A. A. Zaliznyak [3] contains about 100,000 dictionary entries. We assume that each dictionary entry contains 10 words each of which consists of 10 symbols. Then the total amount of the information to be packed equals $10 \cdot 10 \cdot 100,000 = 10^7$ symbols. In the semantic neural network, each symbol corresponds to an individual neuron. Let, on the average, each neuron occupy 1000 bytes [8]. Then the total amount of storage for the semantic neural network realizing the grammatical dictionary of A. A. Zaliznyak will be equal to 10^{10} bytes or about 10 GB. Storage devices of such a volume are already not exotic for a long time and, hence, we can represent the entire grammatical dictionary in the form of a semantic network in which to each form of a word from a dictionary entry corresponds an individual word in the form of a fragment of the neural network. In practice, the grammatical dictionary realized in the form of a semantic neural network will have a smaller volume. This can be explained by the fact that, in a clocked linear tree used for storage of wordforms, identical symbolic sequences of different words are stored as one fragment of the linear tree.

The layer that extracts the meaning of a text and is realized in the form of a clocked linear tree can be considered as a finite automaton since the number of neurons in a network is bounded and they have a finite number of states and connections. This automaton transits from one state to another after arrival of the next symbol of the input sequence at the layer of extracting meaning. It is convenient to consider the layer of extracting meaning as a collection of finite subautomata (whose number is equal to the number of dictionary entries) rather than as one automaton. It also makes sense to assume that one neuron has one gradient substate between its excited and quiescent states. Let each of these substates be an elementary meaning. To one active substate of a neural automaton correspond one or several excited neurons. Then one fragment of a clocked linear tree (one automaton) will contain some subautomata whose number is equal to the number of dictionary entries or several states that are simultaneously active in one automaton. This solution will make it possible to cope with the problem of multivaluedness of natural languages.

Let us consider a model of a neuron of a dictionary entry. The realization of a semantic neural network on the sequential computer imposes additional requirements on the operating speed of neurons. It is necessary, wherever possible, to increase the operating speed of individual neurons and to decrease the number of neurons in the network since, in this case, neurons are processed sequentially, one after another, and, hence, the total time of computation of one clock cycle of the

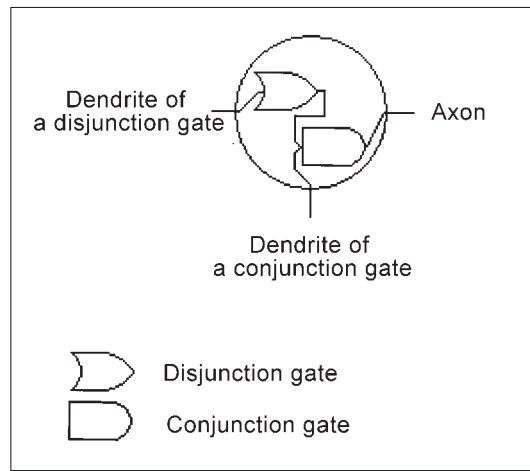


Fig. 1

system is equal to the number of processed neurons multiplied by the time of processing one neuron. To decrease the number of neurons and to increase their operating speed, we combine a disjunction gate and a conjunction gate into one neuron (Fig. 1).

In this case, each neuron has two dendrite trees, one of which performs the operation of disjunction of input gradient values and the other performs the operation of conjunction of input gradient values and the result of the disjunction operation. We denote a neuron by a circle and place the dendrite of its disjunction gate to the left of the circle, the dendrite of its conjunction gate at the bottom or at the top of the circle, and the axon of the latter gate to the right of the circle (Fig. 1). For convenience, in the circle, we write symbols corresponding to the dendrite of the conjunction gate.

Let us consider a model of a dictionary entry. An individual dictionary entry consists of a primary word supporting the base semantic load of the entry and a group of wordforms obtained from the primary word by inflection (conjugation or declension). Let one dictionary entry be a group of neurons or one neural subautomaton in the layer of extracting meaning. Let the total number of substates of a dictionary entry be equal to the number of wordforms of this entry. Let each substate of such a subautomaton be represented by one excited neuron. In this case, if two different neurons of one subautomaton are simultaneously excited, then we assume that the subautomaton has two different active substates simultaneously. Each dictionary entry has its main neuron corresponding to this entry. The main neuron of a dictionary entry is always excited after recognition of a word belonging to this dictionary entry. To each wordform corresponds an individual neuron. It is excited after recognition of the wordform.

The layer of extracting meaning includes neurons that do not belong to individual dictionary entries. These neurons correspond to the attributes of wordforms that are common to many dictionary entries, such as their gender, case, number, tense, etc. They are excited when the wordforms having the corresponding attributes are excited. We assume that the states of the neurons corresponding to the attributes of wordforms also belong to the subautomata of the dictionary entries with which these neurons are connected. Then several dictionary entries can simultaneously be at the same state. For example, all the dictionary entries recognizing the word "коча" are simultaneously at the state "Noun," or, what is the same, their common neuron corresponding to the attribute "Noun" is at the excited state.

The set of excited neurons of a subautomaton corresponds to the set of attributes belonging to an individual wordform recognized by the subautomaton. The problem of classification or determination of a dictionary entry and a wordform from a given symbolic sequence is reduced to the passage of an excitation wave through the layer of extracting meaning and to the excitation of the corresponding subautomaton for the corresponding dictionary entry. The problem of analysis of an inflection is reduced to the provision of the transition of such a subautomaton from its initial state corresponding to the wordform with which the inflection begins to the final state corresponding to the wordform into which the initial wordform must be transformed.

Let us consider the structure of connections of a neural network providing the solution of the problems described above. A clocked linear tree provides the solution of the problem of classification of a wordform on the basis of dictionary entries and the determination of the attributes of this wordform. In the case of multivaluedness, all the dictionary entries and the wordforms corresponding to all the individual values of the wordform are excited in the clocked linear tree. For example,

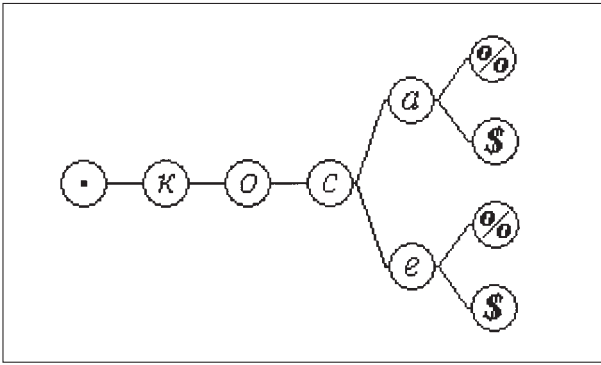


Fig. 2

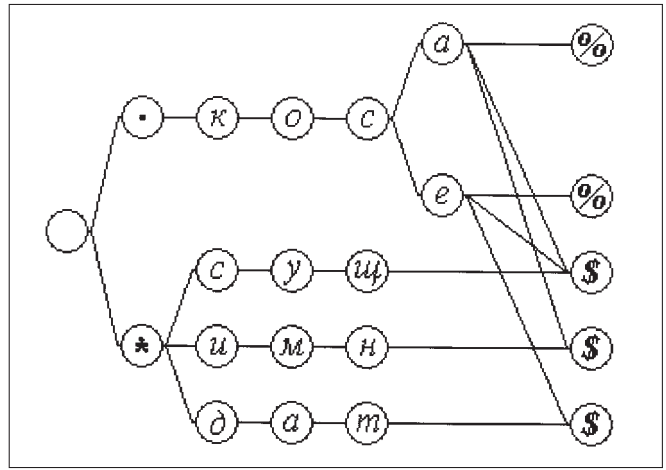


Fig. 3

after applying the word “*косе*” to the input of a clocked linear tree [7], the dictionary entries corresponding to the values “*коса-волосы*,” “*коса-инструмент*,” and “*коса-ландшафт*” are excited. In this case, two wordforms corresponding to the dative and prepositional cases are excited in each dictionary entry.

To solve the problems of word-building and analysis of inflections, we can also use a clocked linear tree. In this case, such a tree is used as a switching circuit commuting the excitation that switches a subautomaton from one state to another. The states of a subautomaton are switched after applying special commands to the input of the clocked linear tree. These commands are recognized by the clocked linear tree and transformed into gradient values at the outputs of the effector neurons corresponding to them, which leads to the excitation or inhibition of the neurons corresponding to the states of a dictionary entry.

Let us describe the internal structure of a wordform in a clocked linear tree. We denote by “_” the unprintable space character. For the convenience of separation of wordforms from service symbolic sequences, we begin each word with a reserved character that has no analogs among the symbols from the symbolic sequence being processed. For convenience, we will restrict the set of symbols of the outside world to the set of the letters and Arabic numerals used in the Russian language. We use the full stop symbol “.” in the capacity of the reserved character “*start*” of a wordform. The receptor of the reserved character “.” is excited by the space “_” located before the first symbol that is not the space character in a wordform. Each wordform in an input sequence also ends with the symbol “_”. Let us introduce two different receptors “%” and “\$” that respond to the space character and to the reserved characters “%” and “\$,” respectively. We will use the first receptor in the capacity of the detector of the end of a wordform considered as a unit dictionary entry (an ordinary word) and the second receptor in the capacity of the detector of a wordform attribute (case, number, conjugation, etc.) that also coincides with the end of the wordform. A fragment of the clocked linear tree constructed for the words “*.коса_*” and “*.косе_*” is given in Fig. 2.

The clocked linear tree presented in Fig. 2 passes to the state “.” after arrival of the reserved character “.” at its input and then passes to the state “.к” after arrival of the symbol “к;” next, it sequentially passes to the states “.ко,” “.кос,” and “.косе” and then simultaneously passes to the substates “.косе%” and “.косе\$” after arrival of the symbols “о,” “с,” “е,” and “_,” respectively.

Let us consider attributes of wordforms. We denote by the symbol “*” the first reserved character of a wordform attribute playing, for wordforms, the same role as the symbol “.”. The beginning of a wordform and its attribute are denoted by different reserved characters to decrease the size of the search tree since this can increase the operating speed of a sequential computer system. However, in order to solve the problem of analysis of inflections on a parallel computer system, it would suffice to distinguish between the reserved characters “\$” and “%.” In Fig. 3, we give an example of the structure of connections of the dictionary entry specifying the following attributes: the noun “*сущ\$,” the subjective case “*имн\$,” and the dative case “*дат\$” of the words “*.коса_*” and “*.косе_*” After arrival of the word “*.коса_*” at the dictionary entry, it passes to the excited substates “.коса%,” “*сущ\$,” and “*имн\$,” and after arrival of the word “*.косе_*” it passes to the ed substates “.косе%,” “*сущ\$,” and “*дат\$.”

We introduce the operation of forced connection and denote it by the reserved character “-”. This operation connects two neurons that are at the excited states after arrival of two fragments of symbolic sequences located to the left

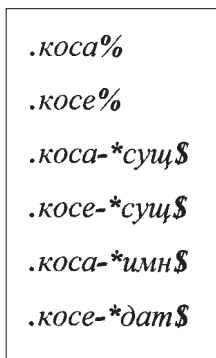


Fig. 4

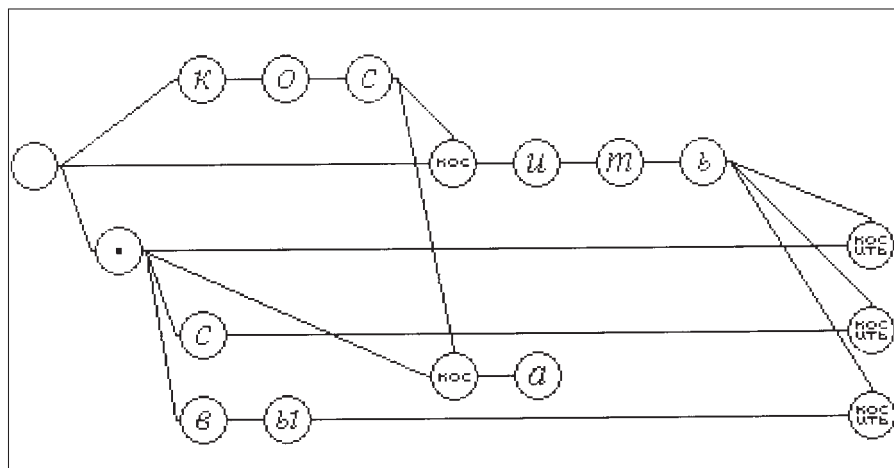


Fig. 5

and to the right of this character at the input of a clocked linear tree. In this case, the excited neuron of the left side of the expression forms an axon, and the excited neuron of its right side forms a dendrite. The specification for compilation of the subnetwork presented in Fig. 3 is given in Fig. 4.

A receptor neuron distinguishes between individual symbols of an input symbolic sequence. At its output, it generates a signal reflecting the presence or absence of the corresponding symbol in the text being analyzed. An effector neuron recognizes individual fragments of the input symbolic sequence. Replacing the output signal of a receptor by the output signal of an effector in a clocked linear tree, we can use fragments of symbolic sequences as input symbols.

To denote such fragments in the input symbolic sequence, we will use the reserved characters of parentheses “(” and “).” The specification for compilation of the layer of extracting meaning presented in Fig. 5 is given in Fig. 6.

We have reduced the problem of analysis to the provision of the passage of an excitation wave through the layers of a clocked linear tree. The result of analysis is represented in the form of a collection of substates of this tree. The problem of analysis of inflections is reduced to the provision of the transition of the subautomaton representing a dictionary entry from the state of a declinable wordform to the state of the result of its declension. To provide this functionality, we should introduce connections to provide transitions between states of a dictionary entry. It is worth noting that, in the case of inflection, a changeable word is already available and processed by a dictionary entry. Therefore, the initial state for the problem of inflection is one of the excited states of the dictionary entry. The transition from the excited state corresponding to the initial form of a word to the final state corresponding to the final form of the word occurs after arrival of the corresponding reserved character at the linear clocked tree.

In the capacity of reserved characters providing the transition of dictionary entries from one state to another, we choose symbolic sequence rather than one symbol. This is possible owing to the fact that, as has been mentioned earlier, any neuron can be considered as an effector or a receptor. If some symbolic sequence arrived at the input of a clocked linear tree is successfully recognized, then one or several neurons of this network are excited. The signal from an excited neuron can be used as an output signal of a receptor. The connection providing the transition of a dictionary entry from one substate in another is represented by a neuron performing the operation of conjunction of two input signals, one of which arrives from the neuron representing the initial substate of the transition and the other is the signal from the receptor recognizing the reserved transition character; the output of the connecting neuron is the exciting neuron representing the substate that must be set after arrival of the reserved character.

In Fig. 7, an example of the specification for programming inflections of the words “коса-косе” and “косе-коса” in the network of Fig. 3 is given. In Fig. 8, the structure of the neural network obtained as a result of realization of this specification is given. After arrival of the symbolic sequence “.коса_(*дам\$)_” at this structure, the states “.косе%,” “*суц\$,” and “*дам\$” are excited, which are the same states that are excited after arrival of the symbolic sequence “.косе_.” If we apply the symbolic sequence “.косе_(*имн\$)_” to this structure, then the states “.коса%,” “*суц\$,” and “*имн\$” are excited, i.e., the same states that are excited after arrival of the sequence “.коса_.”

The reserved character “&” is used to denote the simultaneous excitation of several neurons. Each of several symbolic sequences arriving at the input of a clocked linear tree one after another sequentially excites its own collections of neurons.

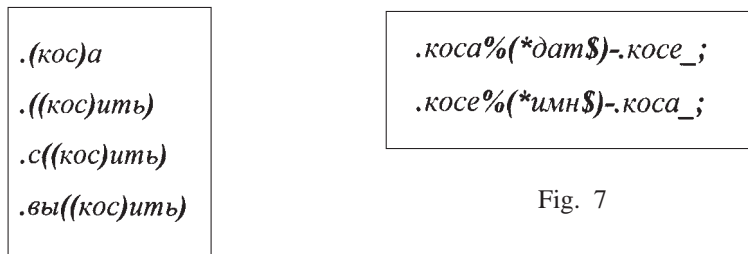


Fig. 7

Fig. 6

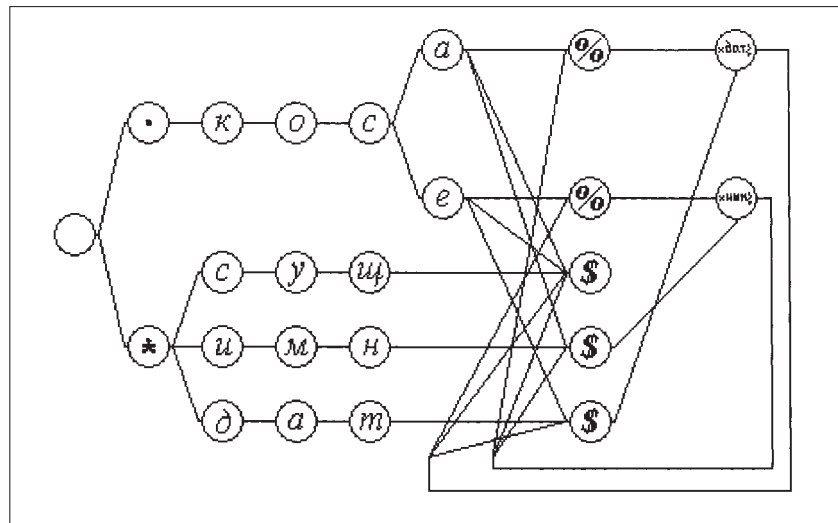


Fig. 8

The symbol “&” placed between these sequences means that the neurons excited by these sequences are simultaneously combined into one collection.

In the capacity of a programming sequence, the reserved character “&” leads to the creation of a neuron that performs the operation of conjunction of the output signals of all the neurons combined into this collection. The reason is that the axons combined into the collection of excited neurons must be connected with a trained neuron that performs the operation of conjunction of all the signals arriving at its inputs. This neuron is excited only if all the neurons belonging to the above-mentioned combination are simultaneously excited.

In Fig. 9, the specification for programming a fragment of a neural network that recognizes a sequence consisting of a plural noun and a plural verb is given. In Fig. 10, the structure of connections of the clocked linear tree obtained after realization of this specification is shown. In this figure, the effector that is conditionally denoted by “I” is excited after arrival of symbolic sequences such as “машины едут” or “спортсмены бегут,” and it is at the quiescent state after arrival of the sequence “машина едет” or “спортсмены бежит.”

A clocked linear tree is programmed (trained) by the principle of storing the entire new information arriving at its input. Some information is considered to be new if, at the moment of its arrival, it is absent in the neural network being programmed. In the programming mode, the excited neurons are searched for at each clock cycle of operation of the neural network being trained. If excited neurons are found, then we consider that the meaning is successfully extracted by the network from the arriving symbolic sequence, and its further training is not needed. If excited neurons are not found, then new neurons are so created that they are excited in the network after arrival of the programming sequence. All the neurons created are at the excited state. Thus, after the next clock cycle of programming, the network is at such a state as if it has yet to be programmed but has already been trained beforehand and has successfully processed an input symbolic sequence.

The proposed structure of a semantic neural network is capable of performing the operations of morphological and syntactic analysis and the operation of analysis of inflections with some restrictions. It cannot correctly perform the morphological analysis and word-building of words with prefixes without the additional separation of these prefixes by the

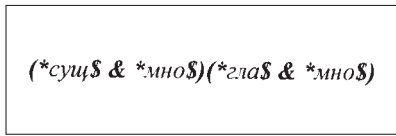


Fig. 9

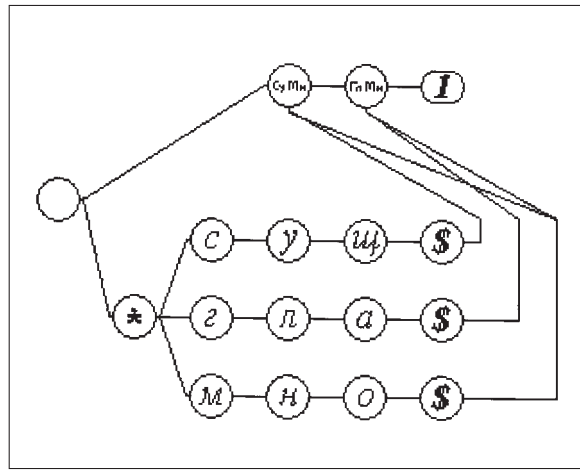


Fig. 10

reserved characters “(” and “).” This drawback can be compensated by considering a word without a prefix and the same word with a prefix as two different words and notions rather than two forms of one word. A linear tree does not make it possible to process recursive relations between words in the case where, in the capacity of an element of some relation, this relation or another relation including the former one is used. This drawback leads to the excessive increase in the number of neurons and to the necessity to program a semantic neural network that takes into account the most probable deep recursive relation and to transform this relation into a sequence of nonrecursive relations in the course of programming. Thus, the proposed structure of a network can extract some subset of projective relations, which depends on the structure of the network, from a symbolic sequence. The possibility of processing any projective relations by a semantic neural network requires the pursuance of new investigations.

REFERENCES

1. T. Vinograd, Understanding Natural Language [Russian translation], Mir, Moscow (1976).
2. D. A. Pospelov, A Fantasy or a Science: On the Way to Artificial Intelligence [in Russian], Nauka, Moscow (1982).
3. A. A. Zaliznyak, Grammatical Dictionary of the Russian Language: Inflections [in Russian], Rus. Yaz., Moscow (1980).
4. J. Von Neumann and A. Burks (ed.), Theory of Self-Reproducing Automata [Russian translation], Mir, Moscow (1971).
5. Z. V. Dudar' and D. E. Shuklin, “A semantic neural network as a formal language of description and processing of meaning of texts in a natural language,” Radioelectronics and Informatics, No. 3, 72-76 (2000).
6. H. Ueno and M. Isidzuka (eds.), Knowledge Representation and Use [Russian translation], Mir, Moscow (1989).
7. D. E. Shuklin, “The structure of a semantic neural network extracting the meaning from a text,” Kibern. Sist. Anal., No 2, 43-48 (2001).
8. Z. V. Dudar' and D. E. Shuklin, “Realization of neurons in semantic neural networks,” Radioelectronics and Informatics, No. 4, 89-96 (2000).