

REALIZATION OF A BINARY CLOCKED LINEAR TREE AND ITS USE FOR PROCESSING TEXTS IN NATURAL LANGUAGES

D. E. Shuklin

UDC 519.7

An implementation and the functioning of a synchronized binary linear tree are described. A structure of the so-called time line is proposed; the structure is used at the discourse level to process semantic connections between elements of various sentences. The structure of a system that processes texts in natural languages is considered. It is assumed that a semiotic feedback can lead to an internal monologue and self-consciousness. Based on the synchronized linear tree, a realization of semiotic feedback is described. It is shown that a model of the above-mentioned internal monologue can be used to synthesize phrases in natural languages.

Keywords: *neural network, morphological analysis, syntactic analysis, internal monologue.*

It is common practice to divide the processing of a text in a natural language into the operations of morphological and syntactic analysis, semantic analysis, and synthesis. In the capacity of the structure of a semantic neural network [1] that performs morphological and syntactic analysis, we use a clocked linear tree [2]. The semantic analysis of contents of a text is based on a model of the corresponding object domain in the form of a neural expert system [3]. The operation of synthesis of the text is realized by a program that restores the text, depending on an excited effector neuron in the clocked linear tree.

A clocked linear tree consists of sublayers of neurons. To each clocked sublayer corresponds some wavefront of processing. The neurons of the first sublayer correspond to the first letter of a word, those of the second sublayer correspond to the second letter, etc. The total number of sublayers is equal to the maximal number of letters in one word. The first, second, third, etc. sublayers consist of neurons recognizing, respectively, the first letter, the first two letters, the first three letters, etc. [2, Fig. 1]. The sublayer of the wavefront contains clocked neurons that perform the conjunction operation and unclocked neurons that perform the disjunction operation. The so-called aggregating sublayers of unclocked neurons that perform the function of disjunction are placed between the sublayers of clocked neurons that perform the function of conjunction. As a result, we obtain a multi-layer structure in which an aggregating sublayer is before each sublayer of the wavefront [2, Figs. 2 and 3].

To simplify the structure of connections between neurons and to reduce the number of neurons in the entire system constructed from neurons, unclocked OR gates from an aggregating sublayer and the clocked AND gates of the wavefront that are connected with the axons of the OR gates can be combined into one logical element [4, Fig. 1]. Such an element has two dendritic trees, one of which performs the function of disjunction of input gradient values and the second performs the function of conjunction of input gradient values and the result of the function of disjunction. Such a logical element called an OR-AND gate consists of two neurons. It can be represented in the form of the circle presented in Fig. 1. Dendrite 3 of its OR gate is to the left of this circle, dendrite 4 of its AND gate is from above or from below of the circle, and axon 5 is to the right of the circle. We denote additional semantic connections 6 of such an OR-AND gate by lines with dots on the boundary of the circle. For convenience, within the circle, we will write symbols corresponding to dendrites of its AND gate or a dot denoting the excitation of the OR-AND gate.

Kharkov State Technical University of Radioelectronics, Kharkov, Ukraine, shuklin@hotmail.com. Translated from *Kibernetika i Sistemnyi Analiz*, No. 4, pp. 36-43, July-August, 2002. Original article submitted March 11, 2002.

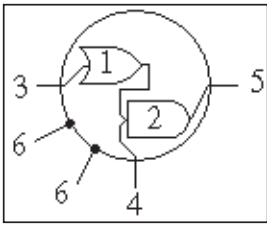


Fig. 1. An OR-AND gate; here, the digits 1, 2, 3, 4, 5, and 6 denote, respectively, an OR gate, an AND gate, an OR-gate dendrite, an AND-gate dendrite, an axon, and additional connections.

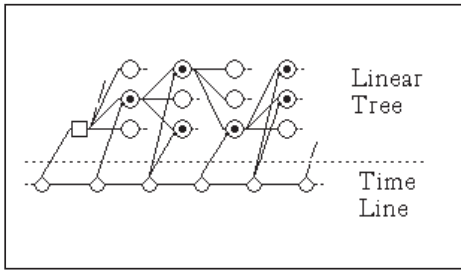


Fig. 2. Connections of OR-AND gates with time-line neurons; here, \odot is an exited OR-AND gate, \circ is a passive OR-AND gate, \diamond is a time-line neuron, and \square is the root neuron of a linear tree.

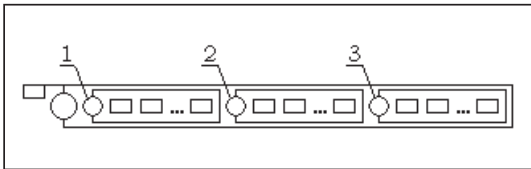


Fig. 3. The structure of an OR-gate neuron of a linear tree: 1 — *sCurrBase*, 2 — *sPrevBase*, and 3 — *sNextBase*.

One sublayer of OR-AND gates corresponds to the adjacent sublayers of unlocked aggregating OR gates and clocked AND gates. As is easily seen, the structure of connections of OR-AND gates is similar to the structure of connections of AND gates in a simplified clocked tree [2, Fig. 1].

If the problem of recognition of spoken languages and written texts need not be solved, then the problem of processing a text in a natural language with the use of a sequential digital computer system can be noticeably simplified. In this case, the text being analyzed is already in the electronic form. If a user enters (in the interactive mode) an incorrect text, then the system can make request for reentering the text. Therefore, it is common practice to require that only correct texts be entered in processing systems. In this case, there is no need to process inaccuracies and errors of recognition, which requires the use of fuzzy logic. A text can be analyzed with the use of a binary semantic neural network based on only two gradient levels, namely, false (0) and true (1). If required, the possibility always exists of repeating the full-scale realization of a semantic neural network.

In realizing a clocked linear tree, one should provide two modes of operations of the tree, namely, the modes of programming (training) the tree and processing a text. At the stage of programming, the system is trained to use the rules of morphology and syntax of a concrete language. In the mode of processing a text, the rules acquired by the system are not corrected and are only applied to input texts. As is described in [4], in the course of programming a clocked linear tree, only the information that is new for the tree is stored and the information available in it is not repeatedly stored. Before the moment of applying some training sequence at the input of the linear tree, some structure is formed in the neural network. In the course of programming a linear tree, the wave of processing is propagated along neural connections of this structure in the same manner as in the mode of processing the text. However, in the course of programming, the subsystem of synthesis of neural structure is activated. In contrast to the mode of processing a text, in the course of wave attenuation in the mode of programming, new OR-AND gates are added to the clocked linear tree. These OR-AND gates are connected by their OR-gate dendrites with the exited neurons of the previous wavefront and by AND-gate dendrites with the exited receptor neurons. It is obvious that new neurons are excited by input signals from the exited OR-AND gates of the previous wavefront and exited receptors of the current quantum of time, providing a continuous wave of processing. In the course of programming a clocked linear tree, the subsystem of synthesizing the neural structure processes the reserved characters (, , . , * , \$, % , _ and - [4] and provides the control of formation of the structure of the neural network.

The process of analysis of a text in a clocked linear tree is distributed in time. In a clocked linear tree, to each quantum of time corresponds some wavefront of processing. The time consists of a sequence of quanta of time represented in the form of a line in which each quantum of time is connected with the previous and succeeding quanta. The result of analysis of the processed part of a text during one quantum of time is the instantaneous state of the clocked linear tree, which is a snapshot of the set of neurons, the set of connections between neurons, and the set of internal states of neurons.

Let us introduce the concept of a time line. A time line consists of a group of neurons. To each quantum of time corresponds some wavefront of processing and some time-line neuron. The neurons of a time line are connected with one another and form a line in which each neuron is connected with two other time-line neurons, namely, with one neuron of the preceding quantum of time and with one neuron of the succeeding quantum of time. Each time-line neuron forms semantic

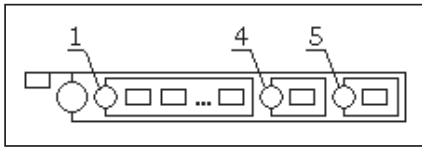


Fig. 4. The structure of a time-line neuron: 1 — $sCurrBase$, 4 — $sPrevLine$, 5 — $sNextLine$.

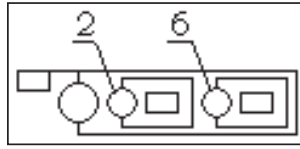


Fig. 5. The structure of a receptor neuron: 2 — $sPrevBase$, 6 — $sCharCode$.

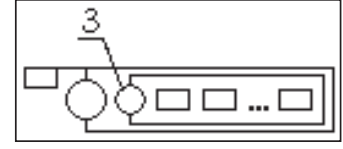


Fig. 6. The structure of a receptor root neuron and a base root neuron: 3 — $sNextBase$.

connections with all the OR-AND gates of the clocked linear tree that are exited during the corresponding quantum of time. This set of exited OR-AND gates forms a wave of processing corresponding to the time-line neuron with which connections are created. A time line is a scratchpad memory for the events that occur in the course of analysis of a text. It can provide the analysis of the text at the discourse level owing to the possibility of restoration of the network state at preceding moments of time and also support the process of programming the structure of a neural network.

The presence of a time line makes it possible to additionally simplify the realization of a clocked linear tree. All the OR-AND gates that are in the exited state during some quantum of time are connected by semantic connections with the neuron that corresponds to this quantum in the time line. Therefore, in this way, one can exclude the exchange of gradient data between OR-AND gates and completely deprive the neuron of its internal state, including excitation. The state of an OR-AND gate (an excited or a passive state) at the given moment of time is determined depending on whether this OR-AND gate has connections with the corresponding time-line neuron (Fig. 2). Hence, a clocked linear tree can be represented in computer memory in the form of a unidirectional graph in which additional attributes are assigned to nodes and connections and whose neurons (nodes) have no internal states.

A binary clocked linear tree consists of layers of OR-AND gates. In the network being described, a realization of the disjunction operation is not needed. In an OR-AND gate, the disjunction operation is performed implicitly and is represented by the collection of various paths of arriving the wave of activity at a definite OR-AND-gate neuron along its AND-gate dendrite. In the mode of processing a text, the conjunction operation is transformed into the pseudo-conjunction operation or operation of addition of the set of connections of the AND-gate dendrite belonging to an OR-AND-gate neuron of the linear tree to the set of connections of a time-line neuron or, in the mode of programming (training), into the operation of equating these collections.

The neurons of a binary clocked linear tree represent the nodes of some graph, and the axons and dendrites of these neurons represent oriented edges of this graph. As was shown in [5], the realization of connections between neurons consists of lists of records of some structure. The connections of the same type are in one list, which is stored in the body of a neuron as one section. For realization of a clocked linear tree, the simplest structure that provides the required functionality consists of one field with a name *id*. This field contains the identifier of the neuron with which the connection is established. Thus, sections consist of lists of connections between neurons.

Let us consider the internal structure of neurons. An OR-AND-gate neuron of a linear tree is presented in Fig. 3. It contains three sections considered below.

Section 1 — $sCurrBase$ is intended for storage of the list of identifiers of the neurons that belong to the receptor layer. The connections of this section are united by the pseudo-conjunction operation. In the training mode, to excite an OR-AND gate, only the receptors that belong to this section must be excited. This occurs when the list of the identifiers of Section 1 — $sCurrBase$ of a neuron of the linear tree and the list of 1 — $sCurrBase$ of a time-line neuron are equal. In the mode of processing a text, in order that a neuron could transmit the activation wave to the next neurons along the connections contained in Section 3 — $sNextBase$, at least all the receptors of this section must be excited. This is provided by addition of the list of identifiers of Section 1 — $sCurrBase$ of the OR-AND gate of the linear tree to the list of identifiers of neurons 1 — $sCurrBase$ of the time-line neuron.

Section 2 — $sPrevBase$ is intended for storage of the list of identifiers of the OR-AND gates that belong to the preceding wavefront of processing. The wavefront of processing arrives at the current neuron along these connections. If there is several such connections, then the front of processing can arrive from different neurons that possibly belong to different layers of the wavefront during one quantum of time. Thus, an OR-AND gate can simultaneously belong to several wavefronts. The arrival of the activation wave at at least one input of a neuron is sufficient to excite it. Hence, the connections contained in Section 2 — $sPrevBase$ are grouped according to the disjunction principle.

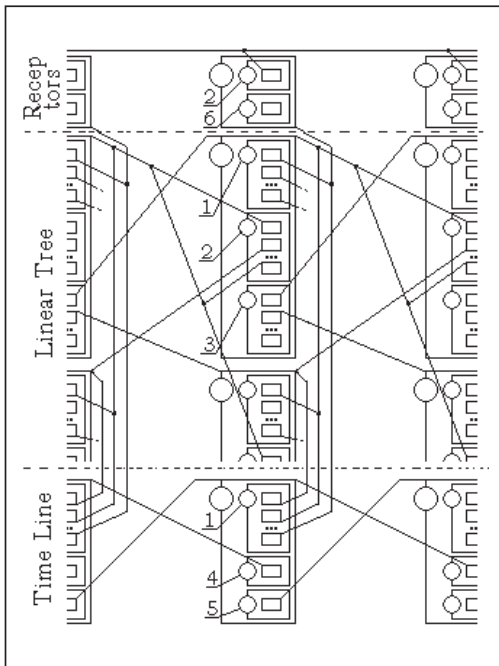


Fig. 7. A formed structure of a network:
 1 — *sCurrBase*, 2 — *sPrevBase*, 3 —
sNextBase, 4 — *sPrevLine*, 5 —
sNextLine, and 6 — *sCharCode*.

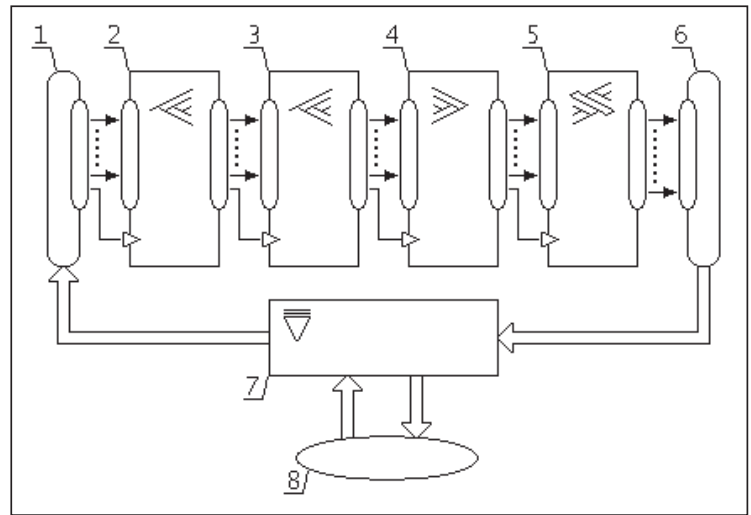


Fig. 8. The block diagram of clocked linear trees; here, 1, 2, 3, 4, 5, 6, 7, and 8 denote, respectively, receptors, a linear tree of morphological analysis, a linear tree of syntactic analysis, an expert system with a model of the object domain, the block of synthesis of reactions of the system (text synthesis), effectors, a commutator, and the system environment.

In binary realization of a semantic neural network, the activation wave is an abstract concept and is not actually propagated along any neurons. Since a single-thread realization of a virtual machine is considered, only one neuron is processed by the virtual machine at one moment of time; therefore, we will say that this neuron has the focus of processing. Instead of the propagation of a wave of processing, the focus of processing moves along the network being processed and realizes a sequential traversal of neurons along the tree of their connections.

Section 3 — *sNextBase* is intended for storage of the collection of identifiers of OR-AND gates that belong to the next wavefront of processing. Along these connections (axons), neurons transmit waves of activation. All the neurons whose identifiers are in this section will obtain the focus during the propagation of the wavefront of processing. A neuron can simultaneously be activated by several neurons belonging to the same wavefront or to fronts of different waves.

A time-line neuron is a chain of neurons. Each neuron in this chain determines one quantum of the operating time of a clocked linear tree. The first neuron of the line determines the first quantum of time at which the semantic neural network begins its operation. The last neuron in the line represents the present moment of time. In the sections of the first and last neurons, the time line, in contrast to those of the other neurons of the line, there are no identifiers of neurons of nonexistent quanta of time.

The structure of a time-line neuron is presented in Fig. 4.

Section 1 — *sCurrBase* is intended for storage of the collection of identifiers of the exited OR-AND gates that are in the linear tree being processed, including the exited receptors. This section of a time-line neuron contains the list of identifiers of the neurons of the linear tree that are exited at a moment of time; among them is the time-line neuron itself. This list contains all the exited receptors and all the exited OR-AND gates of the linear tree being considered.

Section 4 — *sPrevLine* is intended for storage of the identifier of one neuron of the line of the preceding quantum of time.

Section 5 — *sNextLine* is intended for storage of the identifier of one neuron of the succeeding quantum of time.

The structure of a receptor neuron is given in Fig. 5.

Section 2 — *sPrevBase* is intended for storage of the identifier of the receptor root neuron. Along this connection, the wavefront of excitation arrives at the current receptor neuron. Section 6 — *sCharCode* is intended for storage of the code of the symbol that is recognizable by the receptor.

The structures of a receptor root neuron and a base root neuron are given in Fig. 6.

Section 3 — *sNextBase* is intended for storage of the collection of identifiers of the neurons that belong to the first excitation wave. The root neuron transmits an activation wave along these connections. All the neurons whose identifiers are in this section will obtain the focus during the propagation of the first wavefront of processing.

An example of a constructed network structure is presented in Fig. 7.

The operation of a clocked linear tree begins with the reset of its time line, i.e., the elimination of the results of the previous stage of operation of the tree that are stored in the time line. The reset can destroy (destruct) all the time-line neurons. However, the neurons need not be destructed. The clearing of the time line is necessary only for clearing computer memory. When enough resources are available, the time line can store information on the entire process of functioning of the system, including the previous sessions of user-system interaction. After the reset, a new time-line neuron is created; this neuron corresponds to the zero quantum of time of the current front of processing. In Section 1 — *sCurrBase* of this neuron, a list is formed, which consists of one identifier of the base root neuron. Then the first quantum of time begins, during which the first wavefront of processing is generated.

During the first quantum of time, the time-line neuron corresponding to the quantum of time of the first wavefront of processing is created. Between the neurons of the zero and first quanta, a connection is established with the use of the corresponding fields of 4 — *sPrevLine* and 5 — *sNextLine*. The identifier of the neuron of the first quantum is added to Section 5 — *sNextLine* of the neuron of the zero quantum, and the identifier of the neuron of the zero quantum is added to Section 4 — *sPrevLine* of the neuron of the first quantum. The list of identifiers of the receptor neurons that are exited during the first quantum of time, i.e., that successfully recognize a symbol from the input text, is added to Section 1 — *sCurrBase* of the neuron of the first quantum. The list of identifiers of the exited receptors is taken from Section 1 — *sCurrBase* of the time-line neuron of the receptor layer or from Section 1 — *sCurrBase* of a time-line neuron of the linear tree of the adjacent lower level of abstraction. For example, for the linear tree that realizes syntactic analysis, the receptor level is the linear tree that realizes morphological analysis.

After filling the list of identifiers of the exited receptors, the wavefront is formed. The focus of processing is shifted from the neuron of the first quantum of the time line to the neuron of the zero quantum by means of traversal of one level of all the neurons whose identifiers are located in Section 4 — *sPrevLine* of the neuron of the first quantum. Then the focus of processing is shifted from the neuron of the zero quantum of the time line to the base root neuron by means of traversal of one level of all the neurons whose identifiers are in the Section 1 — *sCurrBase* of the neuron of the zero quantum of the time line. Then the focus of processing is shifted from the base root neuron to the OR-AND-gate neurons of the first wavefront of the clocked linear tree. This is a result of the traversal of one level of all the neurons whose identifiers are located in Section 3 — *sNextBase* of the base root neuron. The check of excited states is performed for all the OR-AND gates of the wavefront by computation of the value of the pseudo-conjunction function. The functioning of the pseudo-conjunction function depends on whether the linear tree functions in the mode of training or in the mode of processing. Each exited OR-AND gate of the first wave of processing establishes a semantic connection with the neuron of the first quantum of the time line.

If the pseudo-conjunction operation is successfully performed in the training mode, then the identifier of the OR-AND gate from the wavefront of processing is added to the list of identifiers of Section 1 — *sCurrBase* of the neuron of the first quantum of the time line. If the pseudo-conjunction operation is unsuccessfully performed in the training mode, then a new OR-AND gate with the list in Section 1 — *sCurrBase*, the latter list being equal to the list of Section 1 — *sCurrBase* of the neuron of the first quantum of the time line, is added to the base, and its identifier is added to the list of Section 1 — *sCurrBase* of the neuron of the first quantum of the time line. After termination of the traversal of one wavefront of processing, in Section 1 — *sCurrBase* of the neuron of the first quantum of the time line, the identifiers of the OR-AND gates of the linear tree that have recognized the first symbol of the input sequence of symbols are accumulated. In this algorithm, the focus of processing is shifted by the method of traversal of a tree up to level 3 in depth. The next quanta of time i and the next wavefronts of processing differ from the ones described above only in that the role of the neurons of the preceding quantum is played not by the root neuron of the zero quantum but the neurons of the wavefront of quantum $i-1$.

In practice, a network that extracts the meaning from a text consists of several clocked linear trees (Fig. 8). Each tree processes the notions of a definite level of abstraction and a special time line is constructed for this purpose. In the block diagram of a system (see Fig. 8) that processes a natural language, commutator 7 contains a queue of symbols of a text that are processed by the system. The input text from environment 8 is transmitted to commutator 7 and passes through it to the receptor layer 1. Clocked trees 2 and 3 perform, respectively, morphological and syntactic analysis of the text [4]. The results of syntactic analysis of the text arrive at the input of expert system 4 in which a model of the object domain assigned to texts in the natural language being used is contained. The expert system performs the semantic analysis of the meaning contained in the text [3]. The results of semantic analysis are transmitted to the block of synthesis of symbolic sequences 5 and then to

the effector layer 6. From the effector layer, a symbolic sequence is transmitted to commutator 7. There are two types of symbolic sequences transmitted from the effector layer, namely, an external reaction of the system and an internal intermediate result. The type of a sequence is determined after execution of blocks 2, 3, 4, and 5 in the form of the excitation of the effectors that corresponds to the reserved characters that switch the modes of commutator 7. If a symbolic sequence is an external reaction of the system, then commutator 7 transmits it to external environment 8. If the symbolic sequence is an intermediate result, then commutator 7 applies it to receptor layer 1. As a result, a semeiotic feedback arises in the system. The system can estimate and correct the obtained intermediate results of processing and transmit them to the external environment only after completing the formation of the result.

Commutator 7 was realized as a program; therefore, it is located in Fig. 8 between the layers of effectors 6 and receptors 1. The results of experiments show that the presence of a semeiotic feedback and repeated processing of the symbolic sequences synthesized in block 5 considerably facilitates the realization of word-building and inflections since, in this case, these operations are produced in blocks 2 and 3. It was also revealed that the presence of internal reserved characters circulating through the semeiotic feedback loop makes it possible to realize, at a primitive level, the operations of planning of conversations and control of attention within the framework of several discussed subjects. The processes arising in the system described are akin to the phenomena of internal monologues and self-consciousnesses that are well known from psychology [6]. In the system realized, expert system 4 is not sufficiently integrated to determine the degree of influence of the presence or absence of a model of an object domain on interactive processes. An experiment with a system that has a rather advanced model of an object domain is of interest. However, the construction of such an advanced imitating model is impossible without recruiting a large number of qualified experts in such different fields of science as psychology, linguistics, and cybernetics.

REFERENCES

1. Z. V. Dudar and D. E. Shuklin, "A semantic neural network as a formal language for description and processing of meaning of texts in natural languages," *Radioelectronics and Informatics*, No. 3, 72-76 (2000).
2. D. E. Shuklin, "The structure of a semantic neural network extracting the meaning from a text in real time," *Kibern. Sist. Anal.*, No. 2, 43-48 (2001).
3. D. E. Shuklin, "The use of a semantic neural network in an expert system that transforms the meaning of a text in a natural language," *Radioelectronics and Informatics*, No. 2, 61-65 (2001).
4. D. E. Shuklin, "The structure of a semantic neural network realizing morphological and syntactic analysis of a text," *Kibern. Sist. Anal.*, No. 5, 172-179 (2001).
5. Z.V. Dudar and D. E. Shuklin, "Realization of neurons in semantic neural networks," *Radioelectronics and Informatics*, No. 4, 89-96 (2000).
6. G. M. Kuchinskii, *Psychology of Internal Dialogue* [in Russian], Universitetskoye, Minsk (1988).