PeerJ Preprints NOT PEER-REVIEWED

Multilayered Model of Speech

Andrey Chistyakov *

Saint Petersburg, Russia

ABSTRACT

Today there are no grammar systems, which allow the creation of a fundamentally new word and concept. All existing grammar systems only work by referring to previously chosen terms, on the bases of which all definitions are created. Implementation of operation on grammar systems are introduced for creation of new terms. The main prerequisite of the research was the rejection of finding a universal solution, which is true for any person. Instead of this, all people were divided into groups according MBTI classification. Each group was expected to have standard perception of new knowledge. Assessment has been conducted to each group. A mathematical model was created as result of the communication. The scheme of dividing words and sentences into components are shown in the first section of the article. The second section shows a notation construction from the components and the notation packing in memory. The third section shows the ability for conscious memory access (self-action and self-image). As a result, the model of human speech was structured, in where it is possible to create new terms from new knowledge independently.

Keywords: Logic; Machine learning; Natural language processing; Speech recognition

1 INTRODUCTION

The problem of human speech modeling has been solved more than once, but each time it faced paradoxes and contradictions. The scientific approach was initiated by Boolean algebra, which transgressed into logic of predicates. The human speech is more complicated than the predicate logic, but at the same time it is free of paradoxes, which occur at the first or higher orders. This article is an attempt to solve certain problems by introducing special restrictions.

Speech analysis will be done in some consecutive stages. We should create automata, which can work over context–sensitive grammer, at first. This grammer will be defined in section 2. This definition based on concept of mask, which is responsible to symbol from ingress aphabet. The automata's transition function devides on some parts, which has specific logic operations over its. The concept of class is introduced by analogy with Brown et al. (1992). The main difference with previous studies is the specific function Φ, which can represent any class as data and some data as the class. The section 3 describe the automata's algorithm. There introduced the simple context as the set of symbols between two terminals.

We describe the hierarchical system of one-modifying automatas in section 4. The automata system should recognise symbols in context-sensitive grammer. Analogies for the hierarchical system exists in scientific literature, for example Wang et al. (2006) or Zitouni (2007), but the article has the significant difference. The difference is in memory system with selfmodifications by the operations over the classes from section 3. The selfmodification ability has some special restrictions because we need to pass over infinite looping of any logical algorithm, related to Gödel theorem.

We now turn to generalized grammar in speech, generally independent of a specific language. Basic elements of speech are lexemes, they are the words with a meaning. We will use following lexemes - Nouns, Verbs and Adjectives. Nouns and Verbs are clear, because they exist in all human languages. Nouns names objects and subjects, Verbs - actions between them. Adjectives marks qualities and quantities. Lexemes do not exist by themselves, but only being bound to sentence with subjects, predicate, and other words, including function words.

Adjectives represent some qualities of the object, which a person initially receives from feeling an interaction with an object, so it is possible to say that adjectives are linked with feelings. Verbs represent actions, which are performed by a person or any other subject. Those actions are usually performed through muscle reflexes. Therefore Verbs are linked with muscle reflexes. Nouns serve as markers of feelings/reflexes triggers. It will be shown from which they originate. Basic sentence are some stable sequences of ingress and egress signals. In this sense, words or lexemes, as letters sequences, are also lower level sentences.

But speech consists of not only egress and ingress sentences, it also includes the human memory. Memory is presumed to be multilevel, objectual, and context oriented. It contains complex, processed reflexes. Memory is arranged in a hierarchical

*Email: ghoort1@gmail.com

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1576v1 | CC-BY 4.0 Open Access | rec: 10 Dec 2015, publ: 9 Dec 2015

graph, its nodes are classes and objects. Class is a Noun collection of Adjectives and Verbs. Object is an implementation of class. Also classes can be represented as data and they may be included in the object. Based on this feature, classes can be converted between each other and packed into memory.

A concept of option is introduced for definition of memory work. This is the main innovation of the concept of the proposed model. In the option's scope there will be two possible options of information decryption. Both of these concepts are not equivalent and they can exist in two different people, but not together in the one person. Every option is fixed randomly, during the creation of the brain, and will not change during the whole lifetime of the person.

A total of four independent options will be introduced, which means that a possibility of 16 different options of human speech are possible in this model. It is the most significant difference from all other attempts of creating human speech modeling mechanisms. It is presumed that this division was made possible in the result of evolution, for the interpretation of all incoming information, while every individual can intake only a part from the "raw" flow, to process and transfer to other individuals, prepared in way which can be used.

The possibility of introducing optionalization is grounded on different psychological theories, based on Karl Jung's archetypes Jung (1971), primarily Myers–Briggs Type Indicator Myers and Myers (1995).

2 BASIC DEFINITIONS

Lets introduce basic definitions for the beginning.

Definition 1. Let a point be a pair from an address, where the address is a natural number, and value equal to 0 or 1.

Definition 2. Layer is one-dimensional finite array of points, with consistent and continuous numbering of addresses from 0 to N, where N is the array length. The array length is fixed during the working process, but values of points are variable.

Definition 3. Let the stack be finite, consistent, and continuous array of layers.

Definition 4. Mask μ is a fixed, finite array of a pair $(i,[0,1])$, where i are natural numbers from 0 to N.

It is necessary to clarify that even though the mask is consistent, i.e. it has no points, which have equal addresses, but all addresses of the mask cannot be continuous. Mask's addresses should have gaps, and their meaning will be discussed below.

Basic operations, in the system of layers and masks, are offset by definition and by blocking the mask range. Suppose that at some point of time values, at a layer, are unchangeable. Then it becomes possible to check all layers, and find a combination of values in the layer, which are represented in the mask. The least address element, of first point of the layer, appropriates to the mask, is called **begin of block** or **offset**. The last address is called as **end of block**. The gap between the beginning and the end - including, is called **block of layer**. A block of layers always matches to a specific mask. Operation of finding entry of a mask - is called obtaining an offset.

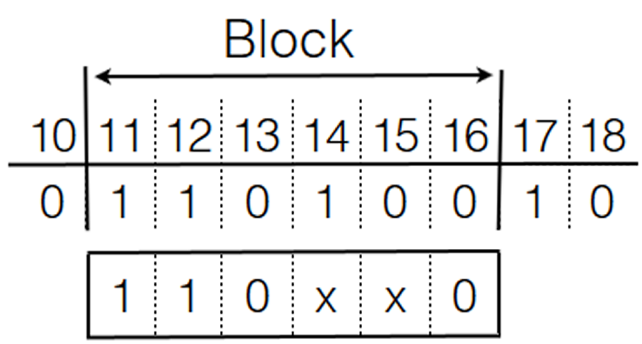

Figure 1. Mask

Requirement 5. Blocks on the same layer should not intersect each other.

Definition 6. Let B be a block of a layer, which consists of serial, continuous set $(b_0, b_1, \ldots, b_m)$. Then it is possible to define the set of the Boolean function $\mathfrak{F}(B)$ of the type $f(b_i, \ldots, b_k) \to [0,1]$. The set $\mathfrak{F}(B)$ will be closed under Boolean operations $\oplus$ (XOR) and & (AND). Let's call $\mathfrak{F}(B)$ as set of **Adjectives** and the result of these values as **quantities**.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1576v1 | CC-BY 4.0 Open Access | rec: 10 Dec 2015, publ: 9 Dec 2015

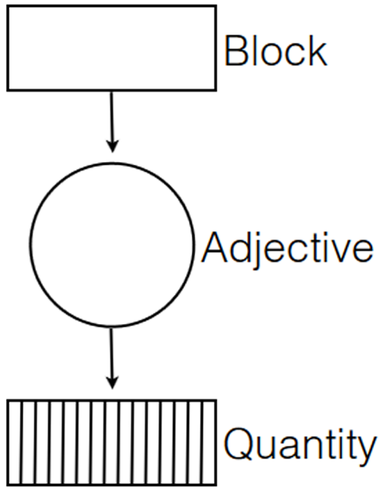

Figure 2. Quality and Adjective

Definition 7. Let there be a set of qualities $Q = (q_0, \dots q_n)$ and block B. Then it is possible to define the set of the Boolean function $\mathfrak{P}(Q,B)$ of the type $f(q_0, \dots q_n, b_i, \dots, b_k) \to [0,1]$. The set $\mathfrak{P}(Q,B)$ will be closed under the Boolean operations $\oplus$ (XOR) and & (AND). Let's call $\mathfrak{P}(Q,B)$ as set of **Verbs** and the resulting values as **actions**.

Definition 8. Let's call the mask set μ, as an array of quantities $Q = (q_0, \dots q_n)$ and an array of actions A, where each element of A is an offset from the beginning of mask, as **Noun**

Definition 9. Let's call the Noun set $\mathtt{N}$, Adjectives $(\mathtt{A}_0, \dots, \mathtt{A}_k)$, and Verbs as **simple class**. Moreover, verb for simple class will be the ingress. Verb will use qualities and blocks as arguments.

We note that an action may be employed, but not on every point of a layer.

Definition 10. Let's call a class without any verbs, or adjective, as an empty class. An empty class will contain only a noun.

Under a simple class six operations can be performed:

Noun Specialization Create a new class with a new noun $\mathtt{N}'$ which equals to $\mathtt{N}$ with additional adjective condition $\mathtt{A}_q = 1$.

Noun Argumentation Create a new class with a new noun $\mathtt{N}'$ which equals to $\mathtt{N}$ with additional verb condition $\mathtt{V}_q = 1$.

Verb Addition Add to class a new verb $\mathtt{V}'$ which equals $\mathtt{V}_l \oplus \mathtt{V}_m$.

Verb Multiplication Add to class a new verb $\mathtt{V}'$ which equal $\mathtt{V}_l \,\&\, \mathtt{V}_m$.

Adjective Addition Add to class a new adjective $\mathtt{A}'$ which equals to $\mathtt{A}_l \oplus \mathtt{A}_m$.

Adjective Multiplication Add to class a new adjective $\mathtt{A}'$ which equals to $\mathtt{A}_l \,\&\, \mathtt{A}_m$.

Due to finiteness of the quantity of verbs and adjectives in the class, it is possible to introduce the specific bijective function Φ, such that it will display a class of operation to the layer.

If for a new adjective creation there is a unique operation, then the creation of the new verb has an internal ambiguity. It is because for a new verb we need to find a new point of action. In order to solve the ambiguity, we need to show one or more active adjectives, which are changed by the verb action point. The action point should be an argument of all active adjective and should not be an argument of all others.

Definition 11. Let's set classes $\mathtt{C}$ and six operations between them. There is a fixed finite array of elements $(C_0, \dots, C_N)$, which are called as basis. Now we can introduce the subset of classes $\tilde{\mathtt{C}} \subset \mathtt{C}$, which can be created from the basis, by a finite count of operations. And we can introduce bijection Φ from $\tilde{\mathtt{C}}$ to the natural number set $\mathbb{N}$. Let's call it as **overclass mapping**.

The specific form of Φ mapping is not important for the further consideration, note that it is invariant during the whole life.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1576v1 | CC-BY 4.0 Open Access | rec: 10 Dec 2015, publ: 9 Dec 2015

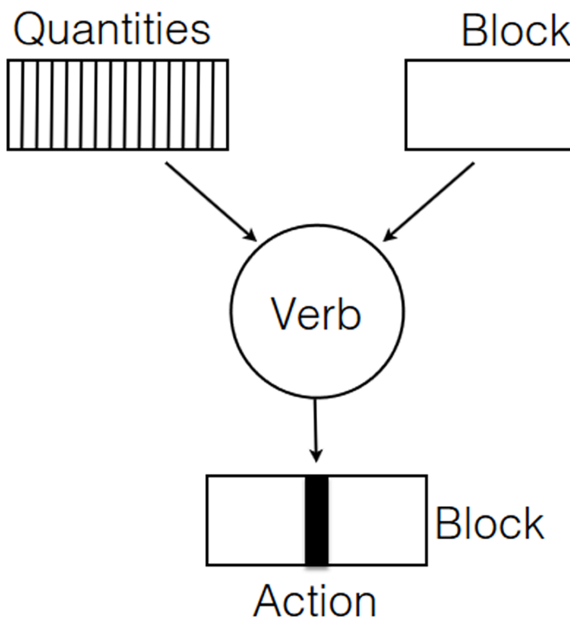

Figure 3. Action of Verb

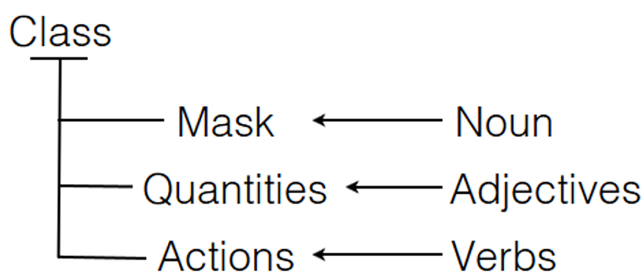

Figure 4. Simple class structure

3 LAYERS WORKING

The mind system is made of stack and common objective memory. First, let's describe layers and their interactions. Null or signal layer is responsible for working with sensory and motor neurons. They excite and inhibit under the influence of the external environment. The current job does not put a purpose to solve the problem of response, so we will not consider the generation of egress signals.

Let's consider layers by induction. There are three layers, $i+1$, $i+2$ and $i+3$. For brevity let's call them as 1st, 2nd and 3rd, and remembering that they are not absolute numbers of layers.

On the 1st layer, there seems to be an excitement of elements. Thereafter, regularities are detected on a layer by overlaying noun's masks. Let all possible nouns be detected and the whole layer will be covered by blocks. It is not necessary to block all points of layer, some points can be free from blocking.

After this, on the 2nd layer, masks are forcibly excited of the same nouns in the same order, but without duplicated elements and gaps. i.e. if on the 1st layer nouns created in a sequence of (A,B,B,B,C,C,D,A,C), then at the 2nd layer there will be a sequence of (A,B,C,D,A,C).

On the next stage, the excitement state will be transmitted to the 3rd layer, also by detecting the noun's masks. Let the sequence $(A,B,\ldots,A,C)$ be responsible for the noun E, and this sequence be called **sentence** of E. It is the simplified version, in any case the nouns which will be important of 2nd layer, are those which were covered in the ask of the 3rd.

Requirement 12. Masks, which are forcibly excited on the layer, will not be detected on it.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1576v1 | CC-BY 4.0 Open Access | rec: 10 Dec 2015, publ: 9 Dec 2015

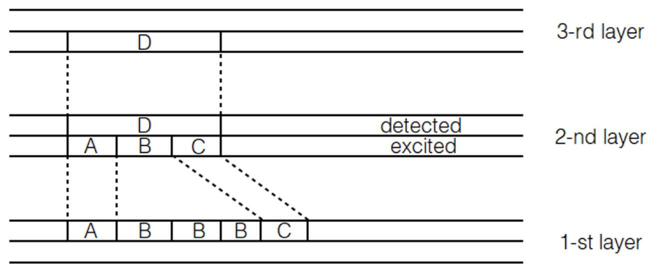

Figure 5. Object detection

Definition 13. Let's call the simple class example which has specific qualities and actions as simple **object**. Every simple object is linked to the simple class, on which it acts. In the current examination objects were created on 1-st, 2-nd and 3-rd layers.

Definition 14. Let's call the set of the simple classes and simple objects, which are denoted by the classes, as 0-level **context**. Let's call the pair of simple class and 0-level contexts as **complex class** 1-level.

Context is made on the 3rd level, made in the 1st and 2nd levels, in which further changes are contained. Now the 2nd layer controls the classes of context and 1st layer controls the objects.

Let's consider the behavior of objects of the 1st layer. They are in one context of 3rd layer, so they are connected together. In the beginning, they load their qualities from appropriate blocks of the 1st layer. In pseudocode:

for all object o in context **do**
  for all adjective a in o **do**
    let q_a is quality of a
    let $B(o)$ is block of o
    $q_a \leftarrow a(B(o))$
  end for
end for

Then placement of verbs happens. Let the verb f acts on the class D, and its quantities $(q_0,\ldots,q_n)$ be its arguments. Let $B(d)$ be the block of the object b. Then we should find classes with required quantities in context. Let class C_b contain such quantities. Then objects of the class C_b act on each object d of class D:

$$d \to d' = f(B(d), b_i) \text{ for each } i \in (0,\ldots,k)$$

In pseudocode:

for all object d in context **do**
  for all verb v in d **do**
    let a_v is action of v
    let $B(d)$ is block of d
    let qualities $(q_0,\ldots,q_k)$ are linked with v
    for all object b in context **do**
      let C_b is a class of b
      if C_b has qualities $(q_0,\ldots,q_k)$ **then**
        $a_v \leftarrow v(B(d), b)$
      end if
    end for
  end for
end for

It is very important to note that if verbs change the state of the block, instead of adjectives which change the objects, then it affects the following verb's actions.

4 CONTEXT WORKING

Let's introduce a series of definitions by induction:

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1576v1 | CC-BY 4.0 Open Access | rec: 10 Dec 2015, publ: 9 Dec 2015

PeerJ Preprints NOT PEER-REVIEWED

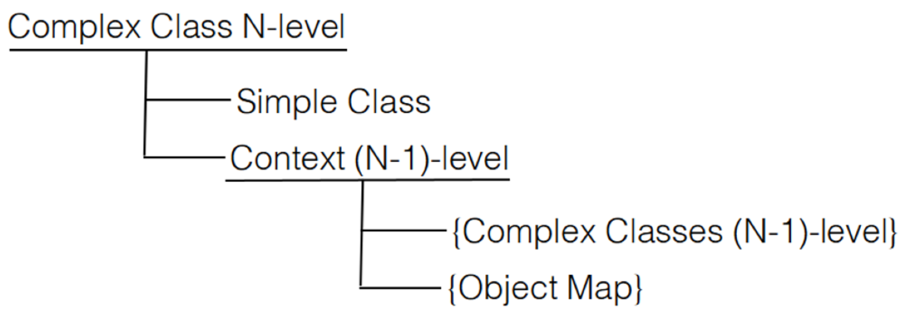

Figure 6. Complex Class

Definition 15. Let's call a pair from the set $(N-1)$-level a complex class and the set $(N-1)$-level a complex objects as N-level **context**. Let's call the pair from a simple class and set of pairs of $(N-1)$-level complex objects and $(N-1)$-level contexts as N-level **complex class**.

In previous section there was a mind stack which contained 3 layers. Later we will consider a mind stack which will contain 6 layers. It this case verbs actions can be applied on 4 lower layers. Let us understand what will happen to them sequentially.

Suppose that required masks were identified at the lowest layer, and after that masks were found on all layers, including the 6th layer. Now let's consider what happens at the 2nd layer. Because contexts are found on the 4th layer, the 2nd one has objects with adjectives and verbs. So excitation will spread onto the 2nd layer. The points on the 2nd layer are responsible for classes on the 1st layer. And due to existence function Φ, which connects data with class operations, modified classes will act on the 1st layer rather than original ones.

Since basic classes were in a global context, they were chosen from common memory of the system. So the modified classes on 2nd layer are already local ones for the their contexts. The local classes are placed into context and completely replace the lower layers for global classes with the same masks. Further operations on local classes only fulfil this context.

The next stage begins when the 5th layer happens. After this, it is possible to modify objects on the 3rd layer. So what happens when object with attached context are changed? There are 2 options: current context is modified or new empty context is created. Mentioned above, every object has qualities and actions. So it is possible to have two situation, each with its own named:

Static New context will be created if any quality on object is modified. Object with old attached context be placed into memory and will be available when specific qualities combination is met.

Dynamic New context will be created if any action on object is modified. Object with old attached context be placed into memory and will be available when specific actions combination is met.

In every mind system only one version of the system happens. It does not change during the lifetime Myers and Myers (1995). Such a case is called **option**, this option is between static and dynamic. Let's call it **context option**.

In the layer system, described in the previous section, the upper layer creates classes on the lower one. In context memory the reverse process takes place, where upper class is modified by a set of classes from the lower level. For that there are so-called **spontaneous** class operations. Every mind system has 3 spontaneous operations, one for nouns, one for verbs, and one for adjectives. Let's introduce appropriate options:

For nouns:

Ratio Noun Specialization operation is spontaneous.

Irratio Noun Argumentation operation is spontaneous.

For verbs:

Logic Operation XOR for verbs is spontaneous.

Ethics Operation AND for verbs is spontaneous.

For adjectives:

Intuition Operation XOR for adjectives is spontaneous.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1576v1 | CC-BY 4.0 Open Access | rec: 10 Dec 2015, publ: 9 Dec 2015

Sensorics Operation AND for adjective is spontaneous.

Now it is possible to show how spontaneous operations work. First, we introduce some notations. Let's represent an object by the symbol as O and classes by C. Upper index of classes and objects will represent a level. Apostrophe in classes C' will represent what class is modified with the respect to the global class of C.

The 3rd layer has an object O_1^3 with a class $C_1^{3'}$. The object O_1^3 has context within which there are objects $\{O_i^2\}$ and the classes $\{C_0^{2'}, C_1^{2'}, \ldots, C_N^{2'}\}$. The task is to modify the class $C_1^{3'}$ and the $\{C_0^{2'}, C_1^{2'}, \ldots, C_N^{2'}\}$ so that the methods (verbs and adjectives) of $C_1^{3''}$ could create lower level classes, and to modify lower level classes, so they could create objects $\{O_i^2\}$. So the objects have to be immutable and classes in context could be modified. Later we will describe the algorithm step by step. But it should be noted that the algorithm creates not the only result, but many available results, and only the final step will choose the most appropriate one.

Algorithm Begin

Step 1 Described previously, there is a bijection Φ between natural numbers and class, based on a global basis. Now consider the inverse transformation of Φ^{-1}, which presents the class as ones and zeros. Let call it as binary representation of class. We will give the table a conventional example of the changes taking place.

$$
\begin{array}{ll}
C_0^{2'} & 10101010\ldots10 \\
C_1^{2'} & 11101010\ldots11 \\
\ldots & \ldots \\
C_N^{2'} & 10101110\ldots00
\end{array}
$$

Step 2 Binary representation of all classes in context is modified by using noun spontaneous operation with condition that modified classes should describe all of its objects $\{O_0^2, \ldots, O_M^2\}$ in same context. The aim is to find same parts in all classes of the binary representations. All later steps will only be about the differences of binary classes representations.

$$
\begin{array}{llll}
C_0^{2''} & 0101 & 001\ldots10 & 10 \\
C_1^{2''} & 0101 & 101\ldots11 & 10 \\
\ldots & \ldots & & \\
C_N^{2''} & 0101 & 010\ldots00 & 10
\end{array}
$$

Step 3 Let's move to the class $C_1^{3'}$. It has some adjectives, each of which take on a binary representation, and as a result gives meaning of quality. It is necessary to create new adjectives via adjective spontaneous operation, which will distinguish all different binary parts through $C_1^{3'}$ qualities.

$$
\begin{array}{llll}
 & & q_0, q_1, \ldots & \\
C_0^{2''} & 0101 & 001\ldots10 & 10 \\
C_1^{2''} & 0101 & 101\ldots11 & 10 \\
\ldots & \ldots & & \\
C_N^{2''} & 0101 & 010\ldots00 & 10
\end{array}
$$

Step 3.1 An additional step will be taken into action, if all possible algorithm results are considered invalid. In this step it is possible to introduce new adjectives which directly represent bits from layers to new qualities.

Step 4 Before this step, let's remember that all classes in contexts were created from global contexts, by a series of verbs transformations. It is very important to note that verbs from $C_1^{3'}$ were used sequentially, when action from one was argument for another. Now it is necessary to do so what lower class will be created faster.

Let there be two verbs $V_1 \to A_1$ and $V_2 \to A_2$, which create actions A_1 and A_2. Also A_1 is an argument for V_2. Therefore the verb $V_1 \circ V_2 \to A_2$ leads to the required value of A_2. A subtask of this step is to find a way, how it is possible, by using spontaneous verb operation, to create new verbs without intersections, by its arguments, and the ability to create classes for its contexts or parts of them.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1576v1 | CC-BY 4.0 Open Access | rec: 10 Dec 2015, publ: 9 Dec 2015

Step 4.1 is also an additional step for invalid results. Let there be context attached to O_1^4 and it be contained in the context of O_1^3. And let this context detect sentences where the quality from O_2^3 has always the same value with the bit x from the block of O_1^3. Then in the mind, after a few tries, the system will create a verb, which connects quality to action over bit x.

Step 5 All previous steps gave some set of possible realizations of class $C_1^{3\,?}$. But the final version will be chosen after a uniqueness and an accuracy check. The memory system waits when the object O_1^3 will be loaded to layer. The upper layer will contain the sentence with O_1^3. The lower layer will contain a set of sentences with the set of that class. If all lower classes are in proposed realization, then it will be chosen as $C_1^{3\prime\prime}$ instead of $C_1^{3\prime}$.

Algorithm End

The process is simplification of memory system, it reduces the whole classes count and creates new verbs and adjectives. It called as **abstraction**. It is important to note that division into spontaneous and non-spontaneous operations required for solving problems with infinite looping. If count of nouns, verbs and adjective is finite then system will have finite count of there derivatives by spontaneous operations. The different combination of context creation and spontaneous operations options give 16 variants of behavior of the entire system.

5 FEEDBACK AND SELFCONTROL

All modified classes are connected by contexts into a hierarchical graph, where upper nodes are dependant of the lower ones. In this system, all changes go strictly upward by the described algorithm. New descending branches are added to the graph. Every class in memory has a full name:

$$\left(\left[O_i^N, O_j^{N-1}, \ldots, O_k^n\right], C_k^{n\prime}\right)$$

which has an unambiguous binary representation - due to bijection of Φ and context option.

This graph is displayed on a signal layer by a bijection. All changes of this graph will change the state of the signal layer. Suppose that it is possible to control the abstraction process by resource allocation for all required subtasks. Then it will also be possible to show a signal layer on not only the acting class, but also the possible $C_k^{n\,?}$, which are created in the process.

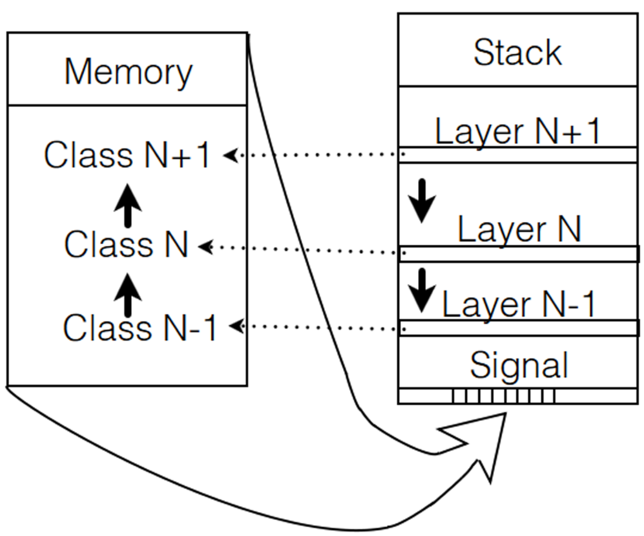

Figure 7. Feedback

Definition 16. Global class, which is located on the 3rd layer according to signal, and contains context with graph of memory, is called **Ego**.

If a stack detects new sequence, not described by current nouns, it will become a new empty class with that sequence in its mask.

Let's consider the function of the suggested model in depth. This model has the following functional description:

Calculator Item, which performs $\oplus$ and & for all ingress data.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1576v1 | CC-BY 4.0 Open Access | rec: 10 Dec 2015, publ: 9 Dec 2015

Generator Item, which creates new sequenses and applies mask operations.

Due to requirement 5 every generator becomes indefinitely connected with a layer or a level. It reads data strictly for the lower the layer or the upper level.

Calculators can work on any layer or level in any quantity, for the reason of that there are no requirements to work in a single thread. They already have the data for work. All synchronization operations are performed by generators.

So it is possible to describe the self-control system. Let's introduce the bijection from a pair $\{C,G\}$, where C is a calculator and G is a generator, connected with it, to a singal layer, and call it as a self-control bijection. Using the self-control bijection we can apply the calculation resources to a more important tasks and remove them from less important ones.

6 ACKNOWLEDGMENTS

I thank Ilya Kruglov for editing of this manuscript.

REFERENCES

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, pages 467–479.

Jung, C. G. (1971). *Psychological Types*. Princeton University Press.

Myers, I. B. and Myers, P. B. (1995). *Gifts Differing: Understanding Personality Type*. CPP.

Wang, Y., Wang, Y., Patel, S., and Patel, D. (2006). A Layered Reference Model of the Brain (LRMB). *IEEE Transactions on Systems, Man, and Cybernetics (C)*, 36:124–133.

Zitouni, I. (2007). Backoff hierarchical class n-gram language models: effectiveness to model unseen events in speech recognition. *Computer Speech Language*, pages 88–104.

PeerJ PrePrints | https://doi.org/10.7287/peerj.preprints.1576v1 | CC-BY 4.0 Open Access | rec: 10 Dec 2015, publ: 9 Dec 2015