# **Methods of Eliminating Features from Linguistic Equations**

Dmitry Sitnikov<sup>a</sup>, Polina Sytnikova<sup>a</sup>, Andrii Kovalenko<sup>a</sup>

*<sup>a</sup> Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, City, 61166, Ukrain* 

#### **Abstract**

In this paper approaches to modelling relations between discrete linguistic features are considered. Linguistic equations as a tool for describing complicated logic dependencies between semantic and syntactic features have been investigated. Finite predicate equations have been considered from the viewpoint of quick finding hidden dependencies in data. A way to defining the tightness of links between discrete features has been suggested. For this purpose, different types of substitution operators have been investigated. A class of finite predicates that allows eliminating non-salient features without an increase in the size of the original formula has been considered in relation with some linguistic examples. The results obtained can be used not only in applied linguistic, but also in other fields where deductive inferences in knowledge bases are important.

#### **Keywords <sup>1</sup>**

Knowledge base, linguistic equations, finite predicates, logic equations, variable elimination

## **1. Introduction**

To formalize information on objects and processes in databases, a variety of discrete mathematics methods are used. In cases where such information represented by discrete information features has a complicated logic structure, in particular, to represent it formally, logic equations with Boolean variables are used. Logic methods of pattern recognition suppose composing and solving logic equations with variables that take on values 1 and 0, depending on whether the given object has a certain property. Solving such equations allow either identifying the object by the available set of values for feature variables or determine unknown properties of the given object [1]. A natural generalization of Boolean algebra equations are finite predicate algebra equations [2] that provide the possibility of operating with arbitrary feature variables defined on finite sets (alphabets). Using such equations for building logic inferences in knowledge bases allows extending the possibilities of logic methods for pattern recognition [3].

For solving a variety of linguistic problems, some of which are describe in this paper, a solution is suggested based on finite predicate algebra equations.

A universal way for solving finite algebra equations is the transformation of the predicate defined by a system of logic equations and initial variable values to the perfect disjunctive normal form [1]. Nevertheless, such a procedure implies exhaustive search for a great number of intermediate solutions, and its practical implementation requires significant time and memory resources.

We will show that for some quite general types of linguistic equations, when peculiarities of their structure are taken into consideration, it is possible to develop much simpler methods their solution. Such methods differ from, for example, heuristic algorithms [3], that suppose finding all sets of values for semantic features, which slows down the solution finding process if the number of variables increases and time increases exponentially.

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine EMAIL: dmytro.sytnikov@nure.ua (D. Sitnikov); polina.sytnikova@nure.ua (P. Sytnikova); andrey.kovalenko@nure.ua (A. Kovalenko) ORCID: 0000-0003-1240-7900 (D. Sitnikov); 0000-0002-6688-4641 (P. Sytnikova); 0000-0003-2882-5082 (A. Kovalenko) ©️ 2021 Copyright for this paper by its authors.



Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CEUR Workshop Proceedings (CEUR-WS.org)

## **2. Related Works**

The scientific field NLP deals with natural language processing. Its origin goes back to the middle of the last century. In the last decades it has become one of the most important artificial intelligence technologies. The application of logic methods to the solution of applied linguistic problems is now widely spread not only in linguistic research, but also in other scientific fields related to discrete information processing. One of the main achievements of mathematical linguistic has lately become the application of complicated logic methods to the investigation of natural language syntax [5].

The first stage of solving many practical problems is the construction of a mathematical model that is often represented in the form of equations. Linguistic equations are used in many fields. For example, in [6] the authors use linguistic equations for the description of fuzzy logic inference for solving the problem of increasing effectiveness of electronic control detail production in the automobile industry. In [7] linguistic equations are a basis for developing various types of fuzzy models with different types of rules that describe relations in these models. An interesting application of NLP techniques is also textual information processing in medical reports [8].

A broad field of using linguistic equations has led to the development of methods and algorithms of their solution. For example, in [9] a system of linguistic equations of a special type where each equation can contain operations of concatenation is considered. In [10] equations with formal languages, using all Boolean operations and concatenation have been investigated, issues of solution existence and uniqueness being considered.

Building a linguistic equation is always associated with the description of the set of linguistic features related to a concrete task. In [11] a language for defining mathematical problems and its association with natural language is analyzed by forming corresponding feature sets. In [12] the authors have analyzed sets of linguistic features for developing a model of linguistic constructs for the analysis of writing quality

In several research papers Boolean algebra tools are considered as an approach to solving linguistic equations. For example, paper [13] is devoted to solving a system of Boolean equations with the operations of union and negation. It is determined there whether such a system has solutions.

At present, since huge datasets that are available on the Internet, approaches to solving linguistic problems change and require scalable methods for the analysis of data and texts. In paper [14] it is proposed to use Big Data methods for improving ways for solving semantic problems related to natural language texts.

## **3. Tightness of links between features**

In many practical tasks associated with the semantic processing of natural language information it s not necessary to obtain all semantic feature value sets, but it is required to obtain one or several value sets for target features that are interesting for user. Often it is necessary to find variable value sets under predefined initial conditions represented in the form of a fixed set of values for other features. When such problems are solved the variables that are not included in the initial conditions and are not target ones are eliminated from the equation by the application of existence quantifiers [1].

When knowledge bases with linguistic variables and corresponding inferences are considered questions concerning determining the tightness of links between object features arise. Also, it is often important to know if such links are salient. Probably a formal link between features is stronger if fewer sets of variable values satisfy the equation. At that, if any sets of variable values satisfy the original equation, one can conclude that there is no relation between these variables.

Besides, when practical problems are solved, the following questions arise:

1. How do concrete values of a given feature substituted in the logic equation affect links between other features?

2. How strong is the logic dependence between two (or more) given features?

In order to obtain an answer to the first question, it is necessary to consider the predicates (and equations correspondingly) that after the substitution of a certain feature value are transformed into predicates with a stronger link between features, and the predicates for which the substitution of a given feature value leads to weakening the logic link between features.

In order to obtain an answer to the second question, it is necessary to eliminate from the original equation with the help of the existence quantifier all the variable except the considered ones and investigate the resulting equation with a fewer number of variables, which describes all allowable sets of investigated feature values.

The mentioned procedures will be considered in the next sections.

In order to answer the posed questions, it is necessary to consider different types of finite algebra predicated and effective methods for eliminating variables from such equations.

## **4. Eliminating variables with the help of logic quantification operations and simplifying finite predicate formulae.**

Let predicate P depend on variables x, y, ..., z. Let us define the substitution operator  $a(P)$  (a is an element from the domain for the variable  $x$ ) that is applied to the predicate  $P$  in the following way:

$$
a(P(x, y, \ldots, z)) = P(a, y, \ldots, z).
$$

Let us call this operator a *limiting* one, if the following condition holds

$$
P(a, y, \dots, z) \rightarrow P(x, y, \dots, z) \tag{1}
$$

for any  $x, y, \ldots, z$ .

Let us call this operator a *spreading* one, if the following condition holds:

$$
P(a, y, \ldots, z) \leftarrow P(x, y, \ldots, z) \tag{2}
$$

for any  $x, y, \ldots, z$ .

Let us call this operator a *shifting* one, if both conditions (1) and (2) do not hold.

Limiting operators *strengthen* the logic link between discrete features, spreading substitution operators *weaken* such a link, shifting operators *transform* the link between the features in an arbitrary way.

Let us represent the predicate  $P$  as follows:

$$
P(x, y, ..., z) = x^{a_1} P_1(y, ..., z) \vee x^{a_2} P_2(y, ..., z) \vee ... \vee x^{a_n} P_n(y, ..., z).
$$

Then  $a_1(P) = P_1(y, ..., z)$ 

Obviously, the operator  $a_1(P)$  will be a limiting one, if  $P_1 \rightarrow P_i \ \forall i = 1, 2, ..., n$ .

The operator  $a_1(P)$  will be a spreading one, if  $P_1 \leftarrow P_i \ \forall i = 1, 2, ..., n$ . The operator  $a_1(P)$  will be a shifting one, if both conditions do not hold.

Consider the application of the substitution operator  $a_1$  to a predicate  $P(x, y)$ , where the variables x, y and z have the domains  $\{a_1, a_2\}$ ,  $\{b_1, b_2\}$  and  $\{c_1, c_2\}$  correspondingly.

Suppose

$$
P = x^{a_1} y^{b_1} z^{c_1} \vee x^{a_2} y^{b_1} z^{c_2} \vee x^{a_2} y^{b_1} z^{c_1}.
$$

Then

$$
a_1(P) = y^{b_1} z^{c_1} = (x^{a_1} \vee x^{a_2}) \wedge y^{b_1} z^{c_1} = x^{a_1} y^{b_1} z^{c_1} \vee x^{a_2} y^{b_1} z^{c_1}.
$$

The predicate P, except for those disjuncts contained in  $a_1(P)$ , one more disjunct  $x^{a_2}y^{b_1}z^{c_1}$ . It means that the operator  $a_1$  is a limiting one for the predicate  $P$ . In terms of the introduced definitions, for the given example  $P_1 = y^{b_1}z^{c_1}$ ,  $P_2 = y^{b_1}z^{c_2}$  V  $y^{b_1}z^{c_1}$  and, obviously,  $P_1 \rightarrow P_2$ .

Consider then the predicate

$$
P = x^{a_1} y^{b_1} z^{c_1} \vee x^{a_1} y^{b_1} z^{c_2} \vee x^{a_2} y^{b_1} z^{c_1}.
$$

Then

$$
a_1(P) = y^{b_1} z^{c_1} \vee y^{b_1} z^{c_2} = (x^{a_1} \vee x^{a_2}) \wedge (y^{b_1} z^{c_1} \vee y^{b_1} z^{c_2}) =
$$
  
=  $x^{a_1} y^{b_1} z^{c_1} \vee x^{a_2} y^{b_1} z^{c_2} \vee x^{a_2} y^{b_1} z^{c_1} \vee x^{a_2} y^{b_1} z^{c_2}.$ 

The operator  $a_1$  for this predicate is a spreading one. For the given example  $P_1 = y^{b_1} z^{c_1} \vee y^{b_1} z^{c_2}$ ,  $a P_2 = y^{b_1} z^{c_1}$ . It means  $P_1 \leftarrow P_2$ .

In the case when the predicate P, for example, is represented in the form

$$
P = x^{a_1} y^{b_1} z^{c_1} \vee x^{a_1} y^{b_2} z^{c_2} \vee x^{a_2} y^{b_1} z^{c_2}.
$$
  
\n
$$
a_1(P) = y^{b_1} z^{c_1} \vee y^{b_2} z^{c_2} = (x^{a_1} \vee x^{a_2}) \wedge (y^{b_1} z^{c_1} \vee y^{b_2} z^{c_2}) =
$$
  
\n
$$
= x^{a_1} y^{b_1} z^{c_1} \vee x^{a_1} y^{b_2} z^{c_2} \vee x^{a_2} y^{b_1} z^{c_1} \vee x^{a_2} y^{b_2} z^{c_2}
$$

i.e., the operator  $a_1$  is a shifting one.

The general method for variable elimination looks as follows [1]. Consider the finite predicate algebra equation

$$
f(x_1, x_2, \dots x_q, x_{q+1}, \dots, x_g, x_{g+1}, \dots, x_n) = 1,
$$
\n(3)

where each variable  $x_1, x_2, \ldots, x_n$  has the domain  $A_i = \{a_{i1}, a_{i2}, \ldots, a_{ik_i}\}, i = 1, n$ , at that, for the system of features the laws of truthfulness should hold:

$$
x_i^{a_{i1}} \vee x_i^{a_{i2}} \vee \dots \vee x_i^{a_{ik}} = 1, i = \overline{1, n}.
$$

Also the laws of falseness should hold:

<span id="page-3-0"></span> $x_i^{a_{il}} \wedge x_i^{a_{im}} = 0, l \neq m, l, m = \overline{1, k_i}.$ 

Suppose also that for the given values  $\{a_{1j_1}, a_{2j_2},..., a_{qj_q}\}$  of the variables  $x_1, x_2,..., x_q$  it is necessary to compute the values of variables  $x_{q+1}, \ldots, x_n$  such that the equation [\(3\)](#page-3-0) will be true for some values of the variables  $x_{a+1}, \ldots, x_a$ .

Let us define this problem mathematically:

 $\exists x_{q+1} \dots \exists x_g f(a_{1j_1}, \dots a_{qj_q}, x_{q+1}, \dots, x_g, x_{g+1}, \dots, x_n) = 1,$  (4) which can be represented in the finite predicate language as follows:

$$
V_{j=1}^{k_{q+1}} a_{q+1,j} \dots V_{j=1}^{k_g} a_{gj} f(a_{1j_1}, \dots a_{qj_q}, a_{q+1,j}, \dots, a_{g,j}, x_{g+1}, \dots, x_n) = 1,
$$
  
where only for the variables  $x_{g+1}, \dots, x_n$  possible sets of values should be found.

The system of linguistic equations of the form

$$
y^{B_i} = g_i(x_1, x_2, \dots, x_m), \ i = \overline{1, n} \tag{5}
$$

that satisfies the condition

$$
g_i(x_1, x_2, \dots, x_m) \wedge g_j(x_1, x_2, \dots, x_m) = 0, \tag{6}
$$
  

$$
i \neq j, \ i, j, = \overline{1, n}
$$

can be transformed to the following form:

$$
\mathsf{V}_{i=1}^{n} \mathsf{y}^{B_{i}} \wedge g_{i}(x_{1}, x_{2}, \dots, x_{m}) = 1. \tag{7}
$$

Nevertheless, the algorithm of variable elimination with the help of the existence quantifier has a high complexity if arbitrary predicates are considered.

Suppose the predicate  $P(x_1, x_2, ..., x_n)$  has the following form:

$$
P = P_1(x_{i_{11}}, x_{i_{12}}, \dots, x_{i_{1n}}) P_2(x_{i_{21}}, x_{i_{22}}, \dots, x_{i_{2n}}) \wedge \dots \wedge P_n(x_{i_{n1}}, x_{i_{n2}}, \dots, x_{i_{nn}}),
$$

where

$$
\{x_{i_{11}}, x_{i_{12}}, \dots, x_{i_{1n}}\} \cap \{x_{i_{21}}, x_{i_{22}}, \dots, x_{i_{2n}}\} \cap \dots
$$
  

$$
\cap \{x_{i_{n1}}, x_{i_{n2}}, \dots, x_{i_{nn}}\} = \{x_{j_1}, x_{j_2}, \dots, x_{j_m}\} = A.
$$

Suppose also we should eliminate variables the variables from the set A.

Consider the application of the existence quantifier to the variable  $x_{j_1}$  from the set A

$$
\exists x_{j_1}(P) = \mathop{\cup}\limits_{i_1} P_1^{i_1} P_2^{i_1} \dots P_n^{i_1},
$$

where  $P_g^{i_1} = a_{j_1 i_1}(P_g)$ ,  $g = \overline{1, n}$ .

For example,  $P = P_1(x_1, x_2, x_3, x_4) \wedge P_2(x_2, x_3, x_5) \wedge P_3(x_2, x_3, x_6, x_7)$ . Let the domain for  $x_2$ consist of 3 values, and the domain for  $x_3$  from 2 values. Then

$$
\exists x_2(P) = P_1(x_1, a_{21}, x_3, x_4) P_2(a_{21}, x_3, x_5) P_3(a_{21}, x_3, x_6, x_7) \vee P_1(x_1, a_{22}, x_3, x_4) P_2(a_{22}, x_3, x_5) P_3(a_{22}, x_3, x_6, x_7) \vee P_1(x_1, a_{23}, x_3, x_4) P_2(a_{23}, x_3, x_5) P_3(a_{23}, x_3, x_6, x_7) =
$$
  
=  $\vee_{i_1} P_1^{i_1}(x_1, x_3, x_4) P_2^{i_1}(x_3, x_5) P_3^{i_1}(x_3, x_6, x_7).$ 

The quantification looks as follows:

 $\exists x_{j_2}(P) (\exists x_{j_1}(P)) = \mathop{\text{V}}_{i_2 i_1} P_1^{i_1 i_2} P_2^{i_1 i_2} \dots P_n^{i_1 i_2}$ , где  $P_g^{i_1 i_2} = a_{j_2 i_2}(a_{j_1 i_1}(P_g))$ ,  $g = \overline{1, n}$ . Then for the considered example

$$
\exists x_3(P)\exists x_2(P) = P_1(x_1, a_{21}, a_{31}, x_4)P_2(a_{21}, a_{31}, x_5)P_3(a_{21}, a_{31}, x_6, x_7) \vee P_1(x_1, a_{21}, a_{31}, x_4)P_2(a_{21}, a_{31}, x_5)P_3(a_{21}, a_{31}, x_6, x_7) \vee P_1(x_1, a_{22}, a_{31}, x_4)P_2(a_{22}, a_{31}, x_5)P_3(a_{22}, a_{31}, x_6, x_7) \vee P_1(x_1, a_{23}, a_{31}, x_4)P_2(a_{23}, a_{31}, x_5)P_3(a_{23}, a_{31}, x_6, x_7) \vee P_1(x_1, a_{21}, a_{32}, x_4)P_2(a_{21}, a_{32}, x_5)P_3(a_{21}, a_{32}, x_6, x_7) \vee P_1(x_1, a_{22}, a_{32}, x_4)P_2(a_{21}, a_{32}, x_5)P_3(a_{21}, a_{32}, x_6, x_7) \vee P_1(x_1, a_{22}, a_{32}, x_4)P_2(a_{22}, a_{32}, x_5)P_3(a_{22}, a_{32}, x_6, x_7) \vee P_1(x_1, a_{22}, a_{32}, x_4)P_2(a_{22}, a_{32}, x_5)P_3(a_{22}, a_{32}, x_6, x_7) \vee P_1(x_1, a_{23}, a_{33}, x_4)P_3(a_{23}, a_{33}, x_6, x_7) \vee P_1(x_1, a_{23}, a_{33}, x_4)P_3(a_{23}, a_{33}, x_5)P_3(a_{23}, a_{33}, x_6, x_7) \vee P_1(x_1, a_{23}, a_{33}, x_4)P_2(a_{23}, a_{33}, x_5)P_3(a_{23}, a_{33}, x_6, x_7) \vee P_1(x_1, a_{23}, a_{33}, x_4)P_2(a_{23}, a_{33}, x_5)P_3(a_{23}, a_{33}, x_6, x_7) \vee P_1(x_1, a_{23}, a_{33}, x_
$$

$$
\begin{aligned}\n&\n\mathsf{V}\,P_1(x_1, a_{23}, a_{32}, x_4)P_2(a_{23}, a_{32}, x_5)P_3(a_{23}, a_{32}, x_6, x_7) = \\
&= \mathsf{V}\mathsf{V}\,P_1(x_1, a_{2i_1}, a_{3i_2}, x_4)P_2(a_{2i_1}, a_{3i_2}, x_5)P_3(a_{2i_1}, a_{3i_2}, x_6, x_7) = \\
&= \mathsf{V}\mathsf{V}\,P_1^{i_1i_2}(x_1, x_4)P_2^{i_1i_2}(x_5)P_3^{i_1i_2}(x_6, x_7).\n\end{aligned}
$$

It can be seen from the above example that, when the existence quantifier is applied sequentially to the variables from the set A (i.e., to the variables common for all the predicates), the general formula for the original predicate  $n^n$  operations more. Nonetheless, there exist many problems for which the conditions are defined by the predicate that has such a structure that the complexity of problem solving with the help of eliminating non-salient variables and finding values of target variables is much lower than in the general case. There exist also cases where it is possible to simplify a predicate obtained at an intermediary stage of solving the equation. Let us consider such cases.

Consider a particular case of the defined task, where the set A consists of one element:

$$
P = P_1(x_1, \ldots, x_{k_1}, x_l) P_2(x_{k_1+1}, \ldots, x_{k_2}, x_l) \wedge \ldots \wedge P_n(x_{k_{n-1}+1}, \ldots, x_{k_n}, x_l),
$$
(8)

where

<span id="page-4-2"></span><span id="page-4-1"></span><span id="page-4-0"></span> ${x_1, \ldots, x_{k_1}} \cap {x_{k_1+1}, \ldots, x_{k_2}} \cap \ldots \cap {x_{k_{n-1}+1}, \ldots, x_{k_n}} = \emptyset.$ 

Then, as a result of the application of the existence quantifier to the variable  $x_l$ , we obtain the disjunctions of predicate conjunctions the variables of which do not intersect:

$$
\exists x_l(P) = \bigvee_l P_1(x_1, \dots, x_{k_1}, a_{li}) \land \dots \land P_n(x_{k_{n-1}+1}, \dots, x_{k_n}, a_{li}) = \\
= \bigvee_l P_1^i(x_1, \dots, x_{k_1}) \dots P_n^i(x_{k_{n-1}+1}, \dots, x_{k_n}).
$$
\n(9)

Let us investigate the possibility of minimizing the obtained predicate. **Statement 1.** Let  $P = P_1(x_1, ..., x_{k_1}) \wedge ... \wedge P_n(x_{k_{n-1}+1}, ..., x_{k_n}), G = G_1(x_1, ..., x_{k_1}) \wedge ... \wedge$ 

$$
G_n(x_{k_{n-1}+1},...,x_{k_n}).
$$
 The implication 
$$
P \leftarrow G
$$
 (10)

holds if and only if  $\forall i = 1, n \; P_i \leftarrow G_i$ .

**Proof.** *Necessity*. Let  $P_i \to G_i$   $\forall i = 1, n$ , i.e.,  $G_i = P_i \vee \lambda_i$ . Then  $G = \bigwedge_i G_i = \bigwedge_i (P_i \vee \lambda_i) = \bigwedge_i P_i \vee \lambda_i$  $\bigwedge_{i,j} P_i G_j \vee \bigwedge_i \lambda_i = P \vee \Lambda.$ 

Hence the implication [\(10\)](#page-4-0) is true.

*Sufficiency.* Let the implicatio[n \(10\)](#page-4-0) is true. Suppose,  $\exists k \in \{1...n\}$  such that  $P_k \to G_k$ . Then, if the predicate  $P_k$  contains such an elementary conjunction  $C_k$  that it is not present in the predicate  $G_k$ , since the domains for the predicates do not intersect, the predicate G will not contain  $C_k$  as well, whereas it is present in the predicate  $P$ . In this case the implication  $(10)$  $(10)$  $(10)$  is false, which contradicts the premise.

**Consequence.** Let the predicate  $P$  satisfies the condition  $(8)$ , i.e., the application of the operator  $\exists x_l(P)$  is defined by the formula [\(9\).](#page-4-2) Then, if  $\forall k = \overline{1,n}$   $P_k^{i_1} \rightarrow P_k^{i_2}$ , the addend  $i_2$  is simplified and formula [\(9\)](#page-4-2) is minimized.

<span id="page-4-3"></span>**Statement 2.** Let 
$$
P_q = a_{lq}(P), P_g = a_{lg}(P)
$$
. The implication  

$$
P_q \rightarrow P_g
$$
 (11)

is true for the predicate  $P$  one of the following conditions holds:

 $a_{lq}$  is a shifting operator, and  $a_{lg}$  is a spreading operator;

 $a_{lq}$  is a limiting operator, and  $a_{lq}$  is a spreading operator;

 $a_{lq}$  is a limiting operator, and  $a_{lq}$  is a shifting operator.

**Proof**. *Necessity*. Suppose the implication [\(11\)](#page-4-3) is true. Suppose  $a_{1a}$  is a spreading operator, and  $a_{1a}$ is a limiting operator. Then  $a_{lq}(P) \rightarrow P, P \rightarrow a_{lq}(P)$ , which contradicts condition [\(11\).](#page-4-3) Suppose that  $a_{lq}$  is a spreading operator, and  $a_{lg}$  is a shifting one. Since the substitution operators  $a_{lq}$  and  $a_{lg}$  are applied to the same variable  $x_l$ , the predicate P contains addends of the form  $x_l^{a_{lq}}A_l$  and  $x_l^{a_{lgj}}$ , where  $\bigvee_j B_j \to \bigvee_i A_i$ . Then  $a_{lq}(P) = \bigvee_i A_i \vee C$ ;  $a_{lg}(P) = \bigvee_j B_j \vee C$ , where C does not depend on the variable  $x_l$ . Hence,  $P_g \rightarrow P_q$ , which contradicts the premise.

Suppose  $a_{lq}$  and  $a_{lg}$  are shifting operators. Then the analogous considerations demonstrate the fact that  $A_i$  and  $B_j$  are different and the condition [\(11\)](#page-4-3) does not hold.

Let  $a_{lq}$  is a shifting operator, and  $a_{lg}$  is a limiting operator. Then in elementary conjunctions containing the predicate  $x_l^{a_{lg}}$ , are not present in the disjunctive normal form for the predicate *P*, and the predicate  $P$  can be presented as follows:

$$
P = \bigvee_m x_l^{a_{lq}} A_m \vee_l B_l \vee C,
$$

where  $B_l$  contains the "recognition" of the variable  $x_l$ , besides  $x_l^{a_{lg}}$  and  $x_l^{a_{lg}}$ , and C does not contain "recognitions" of this variable. Then  $a_{lq}(P) = \bigvee_m A_m \vee C$ ,  $a_{lg}(P) = C$  and  $a_{lg}(P) \rightarrow a_{lq}(P)$ , which contradicts the premise.

If both operators are limiting or spreading ones, their application to the predicate  $P$  is the same. *Sufficiency.* Follows from the first part of the proof.

Example. Let  $P = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{31}} \vee x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{33}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{33}}$ . Then  $a_{21}(P) =$  $= x_1^{a_{11}} x_3^{a_{31}} \vee x_1^{a_{12}} x_3^{a_{33}}$  is a spreading substitution operator, and  $a_{22}(P) = x_1^{a_{12}} x_3^{a_{33}}$  is a shifting operator and

$$
a_{21}(P)\rightarrow a_{22}(P).
$$

Let  $P = x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{33}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{33}} \vee x_3^{a_{31}}$ . Then  $a_{11}(P) = x_3^{a_{31}}$  is a limiting operator, and  $a_{12}(P) = x_1^{a_{12}} x_3^{a_{33}} \vee x_2^{a_{22}} x_3^{a_{33}} \vee x_3^{a_{31}}$  is a shifting one, at that,  $a_{11}(P) \rightarrow a_{12}(P)$ .

Let  $P = x_1^{a_{11}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{33}}$ . Then  $a_{32}(P) = x_1^{a_{11}}$  is a limiting operator, and  $a_{31}(P) = x_1^{a_{11}} \vee x_1^{a_{12}} x_2^{a_{22}}$  is a spreading operator and  $a_{32}(P) \rightarrow a_{31}(P)$ .

## **5. A method of feature elimination from predicates represented in the disjunctive and conjunctive normal forms.**

Let us consider some types of predicate equations whose structure allows us to substantially simplify the process of variable elimination.

Suppose we have a model represented in the form of systems of logic equations that are written in the disjunctive normal form where all conjunctions of different elements on the left side are equal to zero. Then, by transforming the system to a single equation  $((5)-(7))$ , we obtain an equation in the disjunctive normal form. By denoting  $G_{ij}$   $i_{th}$  disjunction in the equation corresponding to the description of  $j<sub>th</sub>$  object in the subject field, the following equation is obtained:

$$
P = \bigvee_{i,j} G_{ij}.\tag{12}
$$

Then  $\exists x_l(P) = \bigvee_{i,j} (G_{ij})$ . The application of the quantifier  $\exists x_l(G_{ij})$  does not change  $G_{ij}$ , if  $G_{ij}$  des

not contain the variable  $x_l$ . Thus, the application of the quantifier in this case is equivalent to eliminating the recognition of the given variable from the elementary conjunction. It follows from the described properties that in the case of disjunctive normal form eliminating variables from equation (7) by the application of the existence quantifier simplifies the given equation, since the number of recognitions (elementary predicates with one variable that are equal to 1 if and only if the value of the variable is the same as the given element) does not increase (very often decreases).

Let a model be represented in the conjunctive normal forms in which every elementary disjunction is a unary predicate. Suppose also that property (6) holds. In this case this condition means that for any two conjunctive normal forms on the right side of the equations the following statement is true: we can find two elementary disjunctions from different conjunctive normal forms the multiplication of which is zero. Then for solving linguistic equation of this type it is possible to apply the existence quantifier to the intermediary variables to eliminate them. After transforming the given system of equations to a single logic equation we obtain an expression that is written as the disjunction of conjunctive normal forms containing elementary conjunctions represented by unary predicates, which substantially simplifies the process of eliminating the non-salient variables.

Let us denote elementary disjunctions in the form of unary predicates as  $D_{ii}(j = 1, k)$ . The application of the existence quantifier to the disjunction of the conjunctive normal forms means the application of the quantifier to this variable in every conjunctive normal form  $\Phi_i$ , for which the following formula is true:

$$
\Phi_i = \Lambda_{i=1}^k D_{ij}.
$$

Every disjunctive normal form  $D_{ij}$  contains the disjunction of a certain number of recognition predicates for  $x_j$ . Taking into consideration the above notation, it is possible to write down the following equation:

$$
\exists x_j \Phi_i = D_{i1} D_{i2} \dots D_{i(j-1)} (\exists x_j D_{ij}) D_{i(j+1)} \dots D_{ik} \equiv D_{i1} D_{i2} \dots D_{i(j-1)} D_{i(j+1)} \dots D_{ik},
$$

where  $i = \overline{1, T}$ , T is the cardinality of the subject field. If the conjunctive normal form  $\Phi_i$  does not contain some variable  $x_c$ , the application of the existence quantifier does not change  $\Phi_i$ .

Thus, the application of the existence quantifier to the intermediary variables does not lead to any increase in the number of recognitions. In some cases, eliminating variable with the help of the quantifier does not change the original formula. Nevertheless, as a rule, the application of the described method leads to a substantial decrease in the number of formula terms (recognitions). Hence, the using of the quantifier in the considered cases does not complicate the original model.

Let us represent a generalized method for solving systems of linguistic equations with target variables and initial conditions.

1. Check if the conjunctions of any 2 predicates on the right sides of equations are zeros.

2. Represent the original system in the form of a single equation.

3. Substitute the initial values of selected variables in the obtained equation.

4. Eliminate all variables except for the target ones by the application of the existence quantifier.

5. The ordered sets of values for the target variables that satisfy the equation obtained at the previous stage is the solution for this problem.

Let us consider an example of solving a system of linguistic equations the right sides of which are disjunctive normal forms. Consider the following system of logic equations:

$$
\begin{cases}\ny^{A} = x_{1}^{a_{11}} x_{2}^{a_{21}} x_{3}^{a_{32}} \vee x_{1}^{a_{12}} x_{2}^{a_{21}} x_{3}^{a_{33}} \vee x_{1}^{a_{13}} x_{2}^{a_{23}}, \\
y^{B} = x_{1}^{a_{11}} x_{2}^{a_{22}} \vee x_{2}^{a_{21}} x_{3}^{a_{31}}, \\
y^{C} = x_{1}^{a_{11}} x_{2}^{21} x_{3}^{a_{32}} \vee x_{1}^{a_{12}} x_{2}^{a_{21}} x_{3}^{a_{32}}.\n\end{cases}
$$
\n(13)

where  $y = \{A, B, C\}, x_1 \in \{a_{11}, a_{12}, a_{13}\}, x_2 \in \{a_{21}, a_{22}, a_{23}\}, x_3 \in \{a_{31}, a_{32}, a_{33}\}.$ 

It is required for the initial condition  $x_2 = a_{21}$  to find values of the target feature  $x_1$ .

Let us solve the problem step by step in accordance with the algorithm.

1. Check the fact that the paired conjunctions of the right sides of equations are zeros.

$$
x_1^{a_{11}}x_2^{a_{21}}x_3^{a_{32}} \wedge x_1^{a_{11}}x_2^{\hat{a}_{22}} = 0, x_1^{\hat{a}_{12}}x_2^{a_{21}}x_3^{a_{33}} \wedge x_1^{a_{11}}x_2^{a_{22}} = 0, x_1^{\hat{a}_{13}}x_2^{a_{23}} \wedge x_1^{a_{11}}x_2^{a_{22}} = 0, x_1^{a_{11}}x_2^{a_{21}}x_3^{a_{32}} \wedge x_2^{a_{21}}x_3^{a_{31}} = 0, x_1^{a_{12}}x_2^{a_{21}}x_3^{a_{33}} \wedge x_2^{a_{21}}x_3^{a_{31}} = 0, x_1^{a_{13}}x_2^{a_{23}} \wedge x_2^{a_{21}}x_3^{a_{31}} = 0.
$$

Hence, the conjunction of the right sides of the second equation is zero. Further,

$$
x_1^{a_{11}}x_2^{a_{22}} \wedge x_1^{a_{11}}x_2^{21}x_3^{a_{32}} = 0, x_2^{a_{21}}x_3^{a_{31}} \wedge x_1^{a_{11}}x_2^{21}x_3^{a_{32}} = 0, x_1^{a_{11}}x_2^{a_{22}} \wedge x_1^{a_{12}}x_2^{a_{21}}x_3^{a_{32}} = 0, x_2^{a_{21}}x_3^{a_{31}} \wedge x_1^{a_{12}}x_2^{a_{21}}x_3^{a_{32}} = 0.
$$

i.e., the conjunction of the right sides of the second and third equations is zero. By analogy, the conjunction of the right sides of the first and third equations is zero.

2. Let us represent the system (13) in the form of a single equation:

$$
y^A \wedge \left(x_1^{a_{11}} x_2^{a_{21}} x_3^{\bar{a}_{32}} \vee x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{33}} \vee x_1^{a_{13}} x_2^{a_{23}}\right) \vee y^{\bar{B}} \wedge \left(x_1^{a_{11}} x_2^{a_{22}} \vee x_2^{a_{21}} x_3^{a_{31}}\right) \vee
$$
  
 
$$
\vee y^C \wedge \left(x_1^{a_{11}} x_2^{21} x_3^{a_{32}} \vee x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{32}}\right) = 1.
$$

3. Substitute the initial value of the variable  $x_2 = a_{21}$  in the obtained equation to get the following:  $y^A \wedge (x_1^{a_{11}} x_3^{a_{32}} \vee x_1^{a_{12}} x_3^{a_{33}}) \vee y^B \wedge (x_3^{a_{31}}) \vee y^C \wedge (x_1^{a_{11}} x_3^{a_{32}} \vee x_1^{a_{12}} x_3^{a_{32}}) = 1$ 

4. Use the quantifiers for a sequential elimination of the variables  $y$  and  $x_3$  from the last equation:

$$
\exists y \left( y^A \wedge \left( x_1^{a_{11}} x_3^{a_{32}} \vee x_1^{a_{12}} x_3^{a_{33}} \right) \vee y^B \wedge \left( x_3^{a_{31}} \right) \vee y^C \wedge \left( x_1^{a_{11}} x_3^{a_{32}} \vee x_1^{a_{12}} x_3^{a_{32}} \right) \right) = \\ = \left( x_1^{a_{11}} x_3^{a_{32}} \vee x_1^{a_{12}} x_3^{a_{33}} \right) \vee \left( x_3^{a_{31}} \right) \vee \left( x_1^{a_{11}} x_3^{a_{32}} \vee x_1^{a_{12}} x_3^{a_{32}} \right) = 1.
$$

Further,

$$
\exists x_3 \big( x_1^{a_{11}} x_3^{a_{32}} \vee x_1^{a_{12}} x_3^{a_{33}} \big) \vee \big( x_3^{a_{31}} \big) \vee \big( x_1^{a_{11}} x_3^{a_{32}} \vee x_1^{a_{12}} x_3^{a_{32}} \big) =
$$
  
= 
$$
\big( x_1^{a_{11}} \vee x_1^{a_{12}} \big) \vee 1 \vee \big( x_1^{a_{11}} \vee x_1^{a_{12}} \big) = 1.
$$

We will get the identity 1 = 1. This means that for the given initial condition the feature  $x_1$  can take on any value from its domain.

Consider an example of solving a system of linguistic equations the right sides of which are conjunctive normal forms. Suppose we have the following system of equations:

$$
\begin{cases}\ny^A = (x_1^{a_{11}} \vee x_1^{a_{12}})(x_2^{a_{21}} \vee x_2^{a_{23}})x_3^{a_{31}},\ny^B = (x_1^{a_{11}} \vee x_1^{a_{13}})(x_2^{a_{21}} \vee x_2^{a_{22}})x_3^{a_{32}},\ny^C = x_1^{a_{13}} x_2^{a_{23}}.\n\end{cases}
$$
\n(14)

where

 $y = \{A, B, C\}, x_1 \in \{a_{11}, a_{12}, a_{13}\}, x_2 \in \{a_{21}, a_{22}, a_{23}\}, x_3 \in \{a_{31}, a_{32}, a_{33}\}.$ It is required for the initial condition  $x_3 = a_{31}$  to find values of the target feature  $x_2$ .

Solving the defined problem will be carried out in accordance with the suggested algorithm:

1. Check the fact that the paired conjunctions of the right sides of the equations (14) are zeros. The conjunction of the right-hand sides of the first and second equations is equal to zero, since  $x_3^{a_{31}} \wedge$  $\wedge x_3^{a_{32}} = 0$ . The conjunction of the first and third sides of the equations is zero, since  $(x_1^{a_{11}} \vee x_1^{a_{12}}) x_1^{a_{13}} = 0$ . Finally, the conjunction of the right sides of the second and third equations is zero, since  $(x_2^{a_{21}} \vee x_2^{a_{22}}) x_2^{a_{23}} = 0$ .

2. Let us represent the system (14) in the form of a single equation:

 $y^A \wedge (x_1^{a_{11}} \vee x_1^{a_{12}})(x_2^{a_{21}} \vee x_2^{a_{23}}) x_3^{a_{31}} \vee y^B \wedge (x_1^{a_{11}} \vee x_1^{a_{13}})(x_2^{a_{21}} \vee x_2^{a_{22}}) x_3^{a_{32}} \vee y^C \wedge x_1^{a_{13}} x_2^{a_{23}} = 1.$ 

3. Substitute in the resulting equation the initial value of the variable  $x_3 = a_{31}$ . We get the following result:

$$
y^{A}(x_1^{a_{11}} \vee x_1^{a_{12}})(x_2^{a_{21}} \vee x_2^{a_{23}}) \vee y^{C} x_1^{a_{13}} x_2^{a_{23}} = 1.
$$

4. Use the existence quantifiers for a sequential elimination of the variables  $y$  and  $x_1$  from the last equation:

$$
\exists y(y^A(x_1^{a_{11}} \vee x_1^{a_{12}})(x_2^{a_{21}} \vee x_2^{a_{23}}) \vee y^C x_1^{a_{13}} x_2^{a_{23}}) = (x_1^{a_{11}} \vee x_1^{a_{12}})(x_2^{a_{21}} \vee x_2^{a_{23}}) \vee x_1^{a_{13}} x_2^{a_{23}} = 1, \\
 \exists x_1(x_1^{a_{11}} \vee x_1^{a_{12}})(x_2^{a_{21}} \vee x_2^{a_{23}}) \vee x_1^{a_{13}} x_2^{a_{23}} = (x_2^{a_{21}} \vee x_2^{a_{23}}) \vee x_2^{a_{23}} = 1.
$$

The application of the Boolean identity  $a \vee a = a$  gives us the possibility to obtain the following equation:

<span id="page-7-0"></span>
$$
x_2^{a_{21}} \vee x_2^{a_{23}} = 1. \tag{15}
$$

From [\(15\)](#page-7-0)  $x_2$  can be found directly:  $x_2 \in \{a_{21}, a_{23}\}.$ 

#### **6. A method for feature elimination from splitable predicates**

Consider a class of problems that can be described with the help of logic equations having a more complicated structure.

The book [3] has considered a task related to mathematical description of the Russian language morphology, and a general approach to solving this problem has been described. This approach is illustrated on the example of a mathematical description of noun declensions. In Ukrainian, the models will be similar as far as mathematical formulae are concerned, although they will differ substantially from the described ones in the sense of dependences between semantic features, and this matter should be carefully investigated as a very prospective research field. In the English language, we do not observe such a variety of syntactic features, but they are often replaced with special word collocations and a variety of particles that completely change the sense of a verb. This is a great field for further research.

For an unambiguous definition of the first letter in an ending for the main forms of words for the substantive declension (the first letter can take on one of the letters  $\{a, e, \ddot{e}, u, o, v, b, o, a, \}$  a complete and nonreducible set of features has been determined: $x_1$  is the case with the values  $u$ ,  $p$ ,  $\partial$ ,  $\theta$ ,  $m$ ,  $n$ (nominative,...,prepositional);  $x_2$  is the gender with the values *M*, *x*, *c*;  $x_3$  is the number (plural or singular) *e* and *M*;  $x_4$  is the feature of animacy with the values *o* and *H*,  $x_5$  is the feature of stress with the values *y* and  $\delta$ ,  $x_6$  is the sign with the values *c* and *H*,  $x_7$  is the last letter of the basis for the word form with the values  $\tilde{p}$ ,  $\tilde{p}$ , the type of the basis of the word form with the values *т* is hard, *м* is soft.

Thus, the problem definition should describe links between the linguistic variables as follows:

$$
L(x_1, x_2, \ldots, x_8, y_1) = 1.
$$

The first ending letter  $y_1$  and 8 semantic features  $x_1, x_2, \ldots, x_8$  are interconnected with the help of the finite predicate. Since this set of features is complete, the given equation defines the following function:

$$
y_1^{\sigma} = F_{\sigma}(x_1, x_2, \dots, x_8). \tag{16}
$$

The predicate  $F_{\sigma}$  is written in the form of finite algebra formulae and defines recognitions of the variable  $y_1$  from the set  $\{a, e, \ddot{e}, u, o, y, \text{h}, \text{to}, \text{h}, \text{h}\}$ .

Let us illustrate the general task using the described example. An ending starts from the letter *я* in word forms with a soft basis. The form should end in *б*, *в*, *д*, *м*, *п*, *м*, *н*, *п*, *р*, *с*, *m*, *ф*, having the basis ending in *а, е, и, й, о, у, ы, ю, я*, 1) for the singular a) for the feminine nominative case, b) for the genitive case with the masculine and neuter, and animality, 2) for the plural a) in the nominative and accusative case with inanimation and neuter gender, b) in the dative, instrumental and prepositional cases. These rules can be written with the help of finite algebra equations as follows:

$$
y_1^a = (x_8^M(x_7^6 \vee x_7^8 \vee x_7^7 \vee x_7^2 \vee x_7^7 \vee x_7^M \vee x_7^H \vee x_7^H \vee x_7^W \vee x_7^W \vee x_7^W \vee x_7^W \vee x_7^W \vee x_7^V \vee x_7^V \vee x_7^V \vee x_7^V \vee x_7^V \vee x_7^V \vee x_7^W \vee x_7
$$

From the given example and the way the equations are built we can track the following structure of the model: each addend of the given predicate consists of multipliers whose domains do not intersect. At that each multiplier is represented either in the disjunctive normal form or has the same structure as the entire predicate. Thus, it can be split into addends that, in their turn, consist of similar multipliers with the domains that do not intersect. Mathematically, we can represent the predicate  $F_{\sigma}$  in the following form:

$$
F_{\sigma} = A_1 \vee A_2 \vee \dots \vee A_n,\tag{17}
$$

where each  $A_i$ ,  $i = 1, n$  can be represented as follows:

 $A_i = A_i^1(x_1,...,x_{k_1}) \wedge A_i^2(x_{k_1+1},...,x_{k_2}) \wedge ... \wedge A_i^l(x_{k_{l-1}+1},...,x_{k_l})$  $(18)$ 

The predicates  $A_i^j$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, l}$  are written in the disjunctive normal form or can be represented with the help of formulae (17), (18). We call such predicates *splitable*.

In the paper [4] we consider quite a large class of predicates for which it is possible to find an effective algorithm of eliminating variables without any increase in the size of the original formula. In paper [4] the following properties of the existence quantifier have been considered:

1.  $\exists xx^a = 1$ .

2.  $\exists x(P(x) \vee Q(x)) = \exists x P(x) \vee \exists x Q(x)$ .

3.  $\exists x(P(x) \land Q(y) = \exists x P(x) \land Q(y).$ 

4.  $\exists y(P(x) \rightarrow Q(y)) = P(x) \rightarrow \exists y Q(y).$ 

5.  $\exists y(P(x) \rightarrow Q(y)) = P(x) \rightarrow \exists y Q(y).$ 

Suppose  $P_i(x) \wedge P_j(x) = 0, i \neq j, i, j = 1, 2, ..., k$ ,

then

$$
\exists y \left( \left( P_1(x) \to Q_1(y) \right) \land \left( P_2(x) \to Q_2(y) \right) \land \dots \land \left( P_k(x) \to Q_k(y) \right) \right) = \\ = \left( P_1(x) \to \exists y Q_1(y) \right) \land \left( P_2(x) \to \exists y Q_2(y) \right) \land \dots \land \left( P_k(x) \to \exists y Q_k(y) \right).
$$

6. If the identity  $P_i(x) \equiv 0$  does not hold for any  $i = 1, 2, ..., k$  and  $P_i(x) \wedge P_j(x) = 0$  for  $i \neq j, i, j = 1, 2, ..., k$ , then:

 $\exists x((P_1(x) \rightarrow Q_1(y)) \land (P_2(x) \rightarrow Q_2(y)) \land \dots \land (P_k(x) \rightarrow Q_k(y))) = Q_1(y) \lor Q_2(y) \lor \dots \lor Q_k(y).$ The above properties allow formulating rules for building a class  $\Delta_x$  of finite predicates defined on the set of variables  $\{x, y, \ldots, z\}$ . The subset  $\Delta_x$  of the set  $\Sigma$  is defined as follows:

1. All recognitions  $x^a, x^b, \ldots, x^c, (a, b, \ldots, c$  are symbols from the domain for the variable x) belong to  $\Delta_{\gamma}$ .

2. All predicates that do not depend on the variable x belong to  $\Delta_x$ .

3. If the predicates  $P_1$  and  $P_2$  belong to  $\Delta_x$ , the predicate  $P = P_1 \vee P_2$  belongs to  $\Delta_x$ .

4. If the predicate  $P_1$  belongs to  $\Delta_x$ , and the predicate  $P_2$  does not depend on x, then the predicate  $P = P_1 \wedge P_2$  belongs to  $\Delta_x$ .

5. If the predicate  $P_1$  does not depend on x, and the predicate  $P_2$  belongs to  $\Delta_x$ , then the predicate  $P = P_1 \rightarrow P_2$  belongs to  $\Delta_x$ .

6. Suppose the predicates  $P_1, P_2, \ldots, P_k$  do not depend on  $x$ ;  $P_i \wedge P_j = 0$  for  $i \neq j, i, j = 1, 2, \ldots, k$ , the predicates  $Q_1, Q_2, ..., Q_k$  belong to  $\Delta_x$ ; then  $P = (P_1 \rightarrow Q_1) \wedge (P_2 \rightarrow Q_2) \wedge ... \wedge (P_k \rightarrow Q_k)$ belongs to  $\Delta_{x}$ .

7. If the predicates  $P_1, P_2, \ldots, P_k$  depend only on  $x$ ;  $P_i \wedge P_j = 0$  for  $i \neq j$ ,  $i, j = 1, 2, \ldots, k$ ; for any  $i = 1,2,...,k$  the identity  $P_i \equiv 0$  is not true; the predicates  $Q_1, Q_2,..., Q_k$  do not depend on x; then the predicate  $P = (P_1 \rightarrow Q_1) \land (P_2 \rightarrow Q_2) \land ... \land (P_k \rightarrow Q_k)$  belongs to  $\Delta_x$ .

Eliminating features with the help of the existence quantifier gives us all ordered sets of possible feature values for which there *exists at least one allowable set* of values of the other features. If we wish to obtain sets of values of the target features that satisfy the equation *irrespectively of* which values we have for the other features, the variables should be eliminated with the help of the universal quantifier.

The property of predicates to be splitable simplifies significantly the procedure of eliminating variables. When applying this procedure, we will use the following properties of the existence quantifier:

1. Addictiveness property:

 $\exists x_i(A_1 \vee A_2 \vee \ldots \vee A_k) = \exists x_i A_1 \vee \exists x_i A_2 \vee \ldots \vee \exists x_i A_k;$ 

2. The application of the existence quantifier to the conjunction of predicates when only one predicate depends on the variable to be eliminated is identical to the application of this quantifier to the given predicate, whereas the other predicates do not change:

$$
\exists x_i (A_1(x_1,...,x_{k_1}) \land ... \land A_l(x_{k_{l-1}+1},...,x_i,...,x_{k_l}) \land ... \land A_n(x_{k_{n-1}+1},...,x_{k_n})) =
$$
  
=  $(A_1(x_1,...,x_{k_1}) \land ... \land A_n(x_{k_{n-1}+1},...,x_{k_n})) \land \exists x_i A_l(x_{k_{l-1}+1},...,x_i,...,x_{k_l}).$ 

Thus, owing to the described structure of splitable predicates and the above properties of the existence quantifier, the method of eliminating variables is significantly simplified and can be split into the following steps:

Step 1. Split the original predicate into addends.

Step 2. Split the obtained addends into multipliers.

Step 3. Select the multiplier that depends on the variable to be eliminated.

Step 4. If this multiplier is a disjunctive normal form, eliminate the variable from it. This process can be split into the following steps:

a) find the recognition of this variable in every elementary conjunction;

b) if the elementary conjunction consists of a single recognition of the given variable, then the entire disjunctive normal form is replaced with 1;

c) if not, replace this recognition predicate with 1, which is identical to eliminating it from the formula.

Step 5. If the addend is represented not in the disjunctive normal form but has a complex structure, then perform all actions starting from step 1.

Let us illustrate how the method works on the above example.

Let

$$
P = (x_8^M(x_7^6 \vee x_7^B \vee x_7^A \vee x_7^A \vee x_7^A \vee x_7^M \vee x_7^B \vee x_7^B \vee x_7^C \vee x_7^C \vee x_7^F \vee x_7^A \vee x_7^B \ve
$$

Find  $\exists x_2(P)$ .

a) select in the predicate  $P$  all the addends (in this example there is a single addend)

 $(x^{\scriptscriptstyle{\mathrm{M}}}_8(x^{\scriptscriptstyle{\mathrm{G}}}_7\vee x^{\scriptscriptstyle{\mathrm{B}}}_7\vee x^{\scriptscriptstyle{\mathrm{A}}}_7\vee x^{\scriptscriptstyle{\mathrm{M}}}_7\vee x^{\scriptscriptstyle{\mathrm{M}}}_7\vee x^{\scriptscriptstyle{\mathrm{R}}}_7\vee x^{\scriptscriptstyle{\mathrm{R}}}_7\vee x^{\scriptscriptstyle{\mathrm{C}}}_7\vee x^{\scriptscriptstyle{\mathrm{R}}}_7\vee x^{\scriptscriptstyle{\mathrm{R}}}_7)\vee x^{\scriptscriptstyle{\mathrm{A}}}_7\vee x^{\scriptscriptstyle{\mathrm{R}}}_7\vee x^{\scriptscriptstyle{\mathrm{R}}}_7\ve$ V x<sup>y</sup> V x<sup>{b}</sup> V x<sup>{0}</sup> V x<sup>{1}</sup> ) ∧ x<sup>{</sup>{(x<sup>{1}</sup> x<sup>{2}</sup> V x<sup>{1}</sup> V x<sup>{2}</sup> V x<sup>{1}</sup> X<sup>{2}</sup> V x<sup>{1}</sup> V b) select multipliers in the obtained addend:

$$
(x_8^M(x_7^6 \vee x_7^B \vee x_7^A \vee x_7^A \vee x_7^M \vee x_7^M \vee x_7^H \vee x_7^H \vee x_7^B \vee x_7^C \vee x_7^T \vee x_7^A \vee x_7^A)) \vee \vee x_7^3 \vee x_7^2 \vee x_7^M \vee x_7^M \vee x_7^O \vee x_7^V \vee x_7^M \vee x_7^M \vee x_7^M
$$

 $(x_3^e(x_1^u x_2^w \vee x_1^p(x_2^w \vee x_2^c) \vee x_1^B x_2^M x_4^o) \vee x_3^M((x_1^u \vee x_1^B x_4^u)x_2^c \vee x_1^A \vee x_1^T \vee x_1^n)).$ 

c) it is obvious that only the second multiplier depends on  $x_2$ . It has a complex structure, therefore we should split it into addends:

$$
x_3^e(x_1^u x_2^w \vee x_1^p(x_2^w \vee x_2^c) \vee x_1^p x_2^w x_4^o) x_3^w((x_1^u \vee x_1^p x_4^u) x_2^c \vee x_1^q \vee x_1^r \vee x_1^r)
$$

d) split every addend into multipliers:

- $x_3^e$ ;
- $(x_1^{\text{N}}x_2^{\text{N}} \vee x_1^{\text{p}}(x_2^{\text{N}} \vee x_2^{\text{c}}) \vee x_1^{\text{B}}x_2^{\text{N}}x_4^{\text{o}});$
- $\chi_3^M$ ;
- $((x_1^M \vee x_1^B x_4^H)x_2^C \vee x_1^A \vee x_1^T \vee x_1^T).$
- e) the first and third multipliers do not depend on  $x_2$ . Split the rest into addends to get:
- $x_1^{\mu} x_2^{\mu}$ ;
- $x_1^{\bar{p}}(x_2^M \vee x_2^c);$
- $x_1^B x_2^M x_4^0$ ;
- $(x_1^H \vee x_1^B x_4^H)x_2^C;$
- $x_1^{\pi} \vee x_1^{\pi} \vee x_1^{\pi}$ .

f) the first and third addends are disjunctive normal forms. Apply the operation  $\exists x_2(P)$  to get:  $\exists x_2(x_1^{\text{N}}x_2^{\text{N}}) = x_1^{\text{N}}; \exists x_2(x_1^{\text{B}}x_2^{\text{N}}x_4^{\text{O}}) = x_1^{\text{B}}x_4^{\text{O}}.$ 

g) the second and fourth addends should be split into multipliers to get:

- $x_1^p$ ;
- $(x_2^M \vee x_2^C);$
- $(x_1^H \vee x_1^B x_4^H);$
- $x_2^c$ .
- h) apply the operation  $\exists x_2$ :

 $\exists x_2(x_2^M \vee x_2^C) = 1; \ \exists x_2(x_2^C) = 1.$ 

i) taking into consideration the fact that all the other multipliers have not changed, we get the result in the following form:

$$
\exists x_2(P) = (x_8^M(x_7^6 \vee x_7^B \vee x_7^A \vee x_7^3 \vee x_7^M \vee x_7^M \vee x_7^H \vee x_7^H \vee x_7^H \vee x_7^W \vee x_7^H \vee x_
$$

We can see that eliminating variables from splitable predicates simplifies their structure, whereas universal methods lead to an increase in the size of the original formula.

Thus, a generalized algorithm of finding values of target variables under predefined initial conditions from the system (5), where predicates on the right side are splitable can be described as follows:

Step 1. Check the correctness of the model, i.e., whether conditions (6) for right sides hold.

Step 2. Transform the system to a single equation in accordance with formula (7).

Step 3. Substitute the initial values.

Step 4. Eliminate the non-salient variables with the help of the existence quantifier.

Step 5. Find the values of the target variables that satisfy the resulting equation.

In the second example it is demonstrated how it is possible to find the dependence between some linguistic variables if a problem is described in the form of a linguistic variable system. The second example shows how you can find the relationship between some variables if the problem is described by a system of equations. For example, it is necessary to express this dependence between noun gender and particular case and number values. We have considered a relation between a noun gender and particular values for case and number. In the proposed example we have considered the following initial values: accusative case and singular form:  $x_3 = e$  and  $x_1 = B$ .

For simplification of complex deductions only two equations have been considered, although the method itself is universal. The main advantage of this method lies in the fact that the original formulae are simplified at every step.

Consider this example. Suppose we have the following model:

$$
\begin{cases}\ny_1^{bl} = (x_8^m(x_7^6 \vee x_7^e \vee x_7^2 \vee x_7^2 \vee x_7^m \vee x_7^m \vee x_7^n \vee x_7^p \vee x_7^c \vee x_7^m \vee x_7^d) \vee x_7^u) \wedge \\
& \wedge \left(x_3^e x_1^p x_2^{yc} \vee x_3^m(x_4^u \vee x_1^e x_4^u)(x_2^w \vee x_2^{yc})\right), \\
y_1^{\prime o} = (x_8^M(x_7^6 \vee x_7^e \vee x_7^o \vee x_7^2 \vee x_7^m \vee x_7^w \vee x_7^u \vee x_7^u \vee x_7^v \vee x_7^c \vee x_7^m \vee x_7^d) \vee x_7^a \vee x_7^e \vee x_7^w \vee x_7^d \vee x_7^u \vee x_7^u \vee x_7^u \vee x_7^u \vee x_7^v \vee x_7^u \vee x_7^u) \vee x_7^a \vee x_7^e \vee x_7^u \vee x_7^u \vee x_7^u \vee x_7^u \vee x_7^u \vee x_7^u \vee x_7^u) \wedge x_8^e(x_1^{\lambda}(x_2^M \vee x_2^c) \vee x_1^e x_2^{yc}).\n\end{cases}
$$

Find values of the target variable  $x_2$  under the initial conditions: 1)  $x_3 = e$ ; 2)  $x_1 = e$ . Check the correctness of the model and transform the system of equations to a single equation:

 $(y_{1}^{\text{\tiny{bl}}}) \wedge \big( (x_{8}^{\text{\tiny{T}}} (x_{7}^{6} \vee x_{7}^{\text{\tiny{B}}} \vee x_{7}^{\text{\tiny{A}}} \vee x_{7}^{\text{\tiny{B}}} \vee x_{7}^{\text{\tiny{M}}} \vee x_{7}^{\text{\tiny{M}}} \vee x_{7}^{\text{\tiny{H}}} \vee x_{7}^{\text{\tiny{B}}} \vee x_{7}^{\text{\tiny{C}}} \vee x_{7}^{\text{\tiny{C}}} \vee x_{7}^{\text{\tiny{A}}} \vee x_{7}^{\text{\tiny{A}}}) \vee x_{7}^{\text{\tiny{H}}}$  $\wedge$   $(x_3^e x_1^p x_2^w ∨ x_3^w (x_1^w ∪ x_1^w x_4^w)(x_2^w ∨ x_2^w))) ∨ (y_1^{ro}) ∧ ((x_8^w (x_7^6 ∨ x_7^8 ∨ x_7^q ∨ x_7^3 ∨ x_7^w ∨ x_7^w ∨ x_7^w ∨ x_7^w ∨ x_7^w)$ V  $x_7^p$  V  $x_7^c$  V  $x_7^{\phi}$  V  $x_7^{\phi}$ 

$$
\wedge x_3^e(x_1^A(x_2^M \vee x_2^c) \vee x_1^B x_2^W)) = 1.
$$

Substitute the initial values to get:

$$
(y_1^{10}) \wedge ((x_8^{10}(x_7^5 \vee x_7^8 \vee x_7^7 \vee x_7^3 \vee x_7^7 \vee x_7^8 \vee x_7^8 \vee x_7^8 \vee x_7^8 \vee x_7^2 \vee x_7^2 \vee x_7^2 \vee x_7^4 \vee x_7^8 \vee x_7^8
$$

Eliminate all the non-salient values, i.e.,  $y_1, x_7, x_8$ :

 $\exists y ((y_1^{\text{no}}) \wedge ((x_8^{\text{M}} (x_7^{\text{G}} \vee x_7^{\text{B}} \vee x_7^{\text{A}} \vee x_7^{\text{A}} \vee x_7^{\text{M}} \vee x_7^{\text{M}} \vee x_7^{\text{B}} \vee x_7^{\text{B}} \vee x_7^{\text{C}} \vee x_7^{\text{C}} \vee x_7^{\text{A}} \vee x_7^{\text{A}} \vee x_7^{\text{A}} \vee x_7^{\text{A}} \vee x_7^{\text{A}})$ V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  ∧  $(x_2^{\text{K}}))$  ) = (( $x_8^{\text{M}}(x_7^6$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$  V  $x_7^{\text{H}}$ V  $x_7^p$  V  $x_7^c$  V  $x_7^p$  V  $x_7^q$  V  $x_7^e$  V  $x_7^p$  V  $x_7^p$  V  $x_7^p$  V  $x_7^p$  V  $x_7^p$  V  $x_7^m$  V  $x_7^m$ ) ∧  $(x_2^w)$ ) = 1.  $\exists x_7 (((x_8^M(x_7^6 \vee x_7^8 \vee x_7^{\pi} \vee x_7^3 \vee x_7^{\pi} \vee x_7^{\$ V  $x_7^{\text{N}}$  V  $x_7^{\text{O}}$  V  $x_7^{\text{N}}$  V  $x_7^{\text{N}}$  V  $x_7^{\text{N}}$  V  $x_7^{\text{N}}$   $\wedge$   $(x_2^{\text{N}}))$  =  $((x_8^{\text{N}}) \wedge (x_2^{\text{N}}))$  = 1.  $\exists x_8((x_8^M) \wedge (x_2^m)) = (x_2^m) = 1.$ 

Thus, the result is  $x_2^m = 1$ . Hence, for the given initial values the variable  $x_2$  takes on the value { $\mathcal{H}$ }.

### **7. Conclusions**

Logic inferences in a variety of knowledge bases can be done with the help of logic equations. The main advantage of such models is the absence of a predefined input or output. The input and output depend on the problem under consideration. Also, logic equations allow describing much more complex data structures than relational databases or decision trees. The main problem is algorithmic difficulties in solving such equations. Eliminating feature variables sometimes becomes quite a time-consuming procedure. In this paper we have tried to show that there are quite large classes of equations that allow us to eliminate variables without an increase in the size of the original formula. Real-world linguistic problems very often can be solved using methods described in this paper.

This research demonstrates the fact that for a large class of finite predicates eliminating non-salient feature variables with the help of quantifiers. Also, the tightness of links between discrete features has been investigated. It should be noted that the results obtained can be used not only for linguistic problems but also for any knowledge bases with a complex structure.

## **8. References**

- [1] Y. P. Shabanov-Kushnarenko, Teoria intellekta. Matematicheskie sredstva [The theory of intelligence. Mathematical tools]. Visha Shkola Publs., Kharkiv,1984.
- [2] Y. P. Shabanov-Kushnarenko Teoria intellekta. Tehnicheskie sredstva [The theory of intelligence. Technical tools]. Visha Shkola Publs., Kharkiv,1986.
- [3] Y. P. Shabanov-Kushnarenko, Teoria intellekta. Problemy i perspectivy, [The theory of intelligence. Problems and prospects]. Visha Shkola Publs., Kharkiv,1987.
- [4] D.E. Sitnikov, B. D'cruz, P.E. Sitnikova, Extracting Salient Discrete Object Features Based On Composing And Manipulating Logical Equations. Vestnik kharkivskogo gоsydarstvennogo politechnicheckogo yniversiteta [Bulletin of Kharkiv State Polytechnic University] 51 (1999) 186– 192.
- [5] M. Kracht, G. Pullum, The Mathematics of Language. The Mathematical Intelligencer 28 (2006) 74–78.
- [6] S. Gebus, E. Juuso, K. Leiviskä, Knowledge-based linguistic equations for defect detection through functional testing of printed circuit boards. Expert Systems with Applications 36.1 (2009) 292–302.
- [7] A. Isokangas, E. Juuso, Fuzzy Modelling with Linguistic Equations. University of Oulu: Control Engineering Laboratory Department of Process Engineering Publs (2000).
- [8] K. J. Chen, P. H. Dedhia, J. R. Imbus, F. David, Schneider Thyroid Ultrasound Reports: Will the Thyroid Imaging, Reporting, and Data System Improve Natural Language Processing Capture of Critical Thyroid Nodule Features? Journal of Surgical Research 256 (2020) 557–563.
- [9] A. Okhotin, O. Yakimova, Language equations with complementation: Expressive power. Theoretical Computer Science 416 (2012) 71–86.
- [10] A. Okhotin, Decision problems for language equations. Journal of Computer and System Sciences 76, Issues 3–4 (2010) 251–266.
- [11] S. Ganji, K. Dhawan, R. Sinha, Novel textual features for language modelling of intra-sentential code-switching data. Computer Speech & Language 64 (2020) 101099.
- [12] C.A. MacArthur, A. Jennings, Z.A. Philippakos, Which linguistic features predict quality of argumentative writing for college basic writers, and how do those features change with instruction? Reading and Writing 32 (2019) 1553–1574.
- [13] A. Ionescu, E. L. Leiss, On the role of complementation in implicit language equations and relations. Journal of Computer and System Sciences 80.2 (2014) 457–467.
- [14] G. Weikum et al, Big Data Methods for Computational Linguistics. IEEE Data Eng. Bull. 35 (2012) 46–64.