DOI: https://doi.org/10.66118/MMP.2025.202

MATHEMATICAL BASES OF THE METHODOLOGY FOR BUILDING AN INTELLECTUAL SYSTEM BASED ON THE THEORY OF INTELLIGENCE AND THE APPARATUS OF PREDICATE ALGEBRA

Plehova G., Sharonova N., Neronov S., Plehov D., Fedorovych V.

Introduction

A method of comparative identification and tools for building an intelligent system are proposed. It is shown that finite predicate algebra (FPA) is a universal mathematical apparatus for the formal description of deterministic, discrete, finite objects of various nature. FPA makes it possible to interpret knowledge in a rigorous mathematical form, where different features and their meanings are related to each other by means of Boolean and predicate operations. It is shown that when constructing a feature system for a knowledge base, it is advisable to use an algebraic-logical model of the subject area. It is shown that in practical problems related to the interpretation of knowledge, which is mostly presented in textual form, it is not necessary to obtain all sets of values of semantic features, but it is necessary to obtain one or several essential sets of feature values (target variables), which are a fixed set of values of other features. When solving such problems, other variables that are not included in the initial conditions and are not target variables are excluded from the equation by linking them to existential quantifiers. An algorithm for excluding non-essential variables when solving algebraic-logical equations is proposed, which reduces the number of necessary calculations. Finite predicate equations of various types are considered. A description of classification problems based on predicate equations is given. The problem of classifying objects based on features that take discrete values is described mathematically as a solution of predicate equations. A broad class of predicates is described from which redundant variables can be removed and focus can be placed on the relationships between essential variables. A method for removing non-essential variables by applying an existential quantifier is proposed and demonstrated using a real-world example. An example is given of the use of the mathematical apparatus of intelligence theory, the method of comparator identification, and the tools of finite predicate algebra, which consists in constructing a model for the identification of medical diagnostic parameters in the form of a system of predicate equations, the solution of which provides an interpretation of medical knowledge in a given field.

1. Comparator identification – a method of intelligence theory

The need to consider the theory of intelligence as a ***formal doctrine*** is due to the fact that it requires a special mathematical language that is not sufficiently developed in existing branches of mathematics. Therefore, the theory of intelligence, along with a substantive study of the human mind, is also forced to develop the necessary formal apparatus. The possibility of teaching the theory of intelligence deductively, based solely on physically observable facts, is based on the method of axiomatic description of the human mind developed by the scientific school of Prof. Shabanov-Kushnarenko Yu.P. and his followers. This is a method of comparison or ***comparator identification method***. The essence of the method lies in the fact that the test subject (the person whose intellect is being studied) forms the meaning of certain predicates through their physical reactions in specially designed experiments P_1, P_2, ..., P_r.

These experiments reveal the properties of predicates P_1, P_2, ..., P_r, which are formally written as logical equations linking predicate variables X_1, X_2, ..., X_r. Some of these equations are used as axioms or initial postulates of intelligence theory. From the ***axioms***, as from equations, the values of the predicate variables X_1, X_2, ..., X_r are found, which are, respectively, the predicates P_1, P_2, ..., P_r. The internal structure of the found predicates characterizes certain details of the mechanism of intelligence.

The comparison method was first used by Newton in his physical study of human color vision. Acting as a test subject, he observed arbitrary light emissions X_1 and X_2 in the fields of comparison and recorded the equality or inequality of their colors. The predicate $P(X_1, X_2)$ formed in this way was first linked by logical equations-axioms by Grassmann. Based on Grassmann's postulates (laws), Schrödinger was the first to construct a deductive theory of human color vision.

When studying human intelligence using the comparison method, the researcher influences the subject's sensory organs with physical signals (stimuli) X_1, X_2, ..., X_n, which generate certain subjective experiences (states) Y_1, Y_2, ..., Y_n in the subject's consciousness. It is assumed that states Y_1, Y_2, ..., Y_n depend unambiguously on the corresponding stimuli X_1, X_2, ..., X_n. This means that there are (exist) functions

$$Y_1 = f_1(X_1),\ Y_2 = f_2(X_2),\ ...,\ Y_n = f_n(X_n)\ .$$

In experiments, the stimuli X_1, X_2, ..., X_n are taken from sets A_1, A_2, ..., A_n clearly defined by the researcher, so that always $X_1 \in A_1$, $X_2 \in A_2$, ..., $X_n \in A_n$.

The researcher selects the sets A_i $\left(i \in \{1, 2, .., n\}\right)$ arbitrarily, at his discretion, based on the scientific tasks he sets for himself. It is assumed that each of the stimuli $X_i \in A_i$ generates a specific state Y_i. The set of all values of the function $Y_i = f_i\left(X_i\right)$, defined on the set A_i, is denoted by the symbol B_i. Thus, each of the functions f_i is a surjection mapping the set A_i onto the set B_i. The functions f_i characterize the ability of the test subject to respond to external objects with corresponding subjective states.

2. Finite predicate algebra as a universal means of formal description for objects of different nature

It is believed that finite predicate algebra is a universal mathematical apparatus for the formal description of deterministic, discrete, finite objects [1–11]. In works [4, 5], it was proven that in order to develop the theory of intelligence, it is first necessary to have a formal language that can be used to mathematically describe the structures and functions of human intelligence, especially those related to the accumulation and interpretation of knowledge by humans. The scientific school of Professor Yu.P. Shabanov-Kushnarenko and his followers uses ***finite predicate algebra*** (FPA) as such a language.

It should be noted that *deterministic processes* are processes with a unique result, in which there is no element of chance. *Discrete processes* are processes in which information takes the form of separate portions or quanta – digits, letters, words, formulas, etc. – in which there is no factor of continuity. *Finite processes* are processes in which only a finite number of units of information are involved, in which there is no factor of infinity. By definition, the theory of intelligence must be limited to the study of purely machine-like functions of human intelligence, which are characterized by *determinism, discreteness, and finiteness*. This raises questions about the extent to which the imposed restrictions narrow the class of human intelligence functions that can be modeled within the framework of the theory of intelligence.

In order to be able to describe the functions of intelligence mathematically, it was necessary to create a formal language in which such a description could be made. The formal language had to be chosen so that any finite alphabet operator could be written in a convenient form. Such a language is the FPA described below [6–8].

The concept of a *finite predicate* was introduced as follows. Let A be a finite alphabet consisting of k letters a_1, $a_2, \ldots,$ a_k, Σ be a set consisting of two elements denoted by the symbols 0 and 1 and called *false* and *true*, respectively. A variable defined on the set A is called a *(letter) literal*, and a variable defined on the set Σ is called a *logical* variable.

A finite n-place predicate over an alphabet A is any function $f(x_1, x_2, \ldots, x_n) = t$ of n letter arguments $x_1, x_2, \ldots, x_n$, defined on the set A, and which takes the logical values t. Sometimes a finite predicate f is called *k-ary*, emphasizing that its alphabet A consists of k letters.

Each finite alphabet operator can be assigned a specific finite predicate. This can be done as follows. Let

$$F(x_1 x_2 \ldots x_m) = y_1 y_2 \ldots y_m$$

– is an arbitrarily chosen finite alphabet operator that converts input words $x_1 x_2 \ldots x_m$ of length m into output words $y_1 y_2 \ldots y_m$ of the same length, consisting of letters of the alphabet A. Let us construct a $2m$-place finite predicate $f(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m)$ over the alphabet A, using the following rule:

$$f(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m) = \begin{cases} 1, \text{ if } \rightleftarrows F(x_1 x_2 ... x_m) = y_1 y_2 ... y_m \\ 0, \text{ if } F(x_1 x_2 ... x_m) \neq y_1 y_2 ... y_m. \end{cases}$$

Let’s write down the equation:

$$f(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m) = 1 .$$

This equation links the variables $x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m$ by a certain relation. Substituting into $f(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m) = 1$. the letters $x_1, x_2, \ldots, x_m$ of the input word $x_1 x_2 \ldots x_m$ of the alphabet operator F, we obtain as a result of solving this equation the letters $y_1, y_2, \ldots, y_m$ of the output word $y_1 y_2 \ldots y_m$. Thus, the predicate f, constructed in this way, contains all the information about the alphabet operator F that interests us. With its help, we can determine the output word of the alphabet operator represented by this predicate for any input word.

In order to be able to write finite predicates in the form of formulas, a special algebraic system called FPA is introduced. More precisely, not one algebra is introduced, but a whole family of such algebras. *The type of finite predicate algebra* is characterized by *an alphabet of letters* A consisting of symbols $a_1, a_2, ..., a_k$, and an alphabet of variables consisting of n symbols $x_1, x_2, ..., x_n$. Using the finite predicate algebra with the alphabet $A = \{a_1, a_2, ..., a_k\}$ and an alphabet of variables $B = \{x_1, x_2, ..., x_n\}$, any n-place k-ary predicate $f(x_1, x_2, ..., x_n)$ defined over the alphabet A can be written. In this case, it is assumed that the letters and variables of the alphabets A and B are numbered.

In finite predicate algebra, there is a very special function, or more precisely, a unique predicate called the *recognition predicate*. Each formula of the form $a_i\left(x_j\right)$ can be conveniently regarded as a single-place predicate that depends only on the variable x_j and is defined as follows:

$$a_i\left(x_j\right)=\begin{cases}1, \; if \;\; x_j = a_i, \\ 0, \; if \;\; x_j \neq a_i.\end{cases}$$

The predicate $a_i\left(x_j\right)$ «recognizes» a single letter a_i among the possible letters of the alphabet A, and is therefore called the predicate of recognition of the letter a_i or even the recognition of the letter a_i, which depends on the variable x_j. There are k_n different letter recognitions in total. By applying disjunction and conjunction operations to different letter recognitions, repeatedly and in different orders, you can obtain different predicates. Letter recognition is considered to be an *elementary predicate of finite predicate algebra*. The set of elementary operations and elementary predicates forms the basis of *finite predicate algebra*.

It has been proven that the algebra of finite predicates is *complete*. Thus, any finite predicate can be written in the ASP language. The algebra of finite predicates is not only complete, but even *redundant* in a certain sense. The true predicate denoted by formula 1 can be expressed by other means of finite predicate algebra, for example, in the form of a disjunction of all conjunctions of the form $x_1^{\sigma_1}x_2^{\sigma_2}...x_n^{\sigma_n}$. In the case when $k \geq 2$, i.e., when the alphabet A contains at least two letters a_1 and a_2, one can do without formula 0. The predicate denoted by this formula, which is identically false, can be written, for example, in the form of the formula $x_1^{a_1}x_1^{a_2}$.

Most often, *disjunctive algebra* and its various conservative extensions are used as the language for recording finite predicates. The convenience of disjunctive algebra of finite predicates lies in the fact that its language allows for concise and convenient recording of *truth and falsity laws*. These laws play a special role in any finite predicate algebra. In essence, there are requirements that must be satisfied for the correct introduction of variables on finite sets.

The *truth law* specifies the domain of variation of a variable, and the *falsity law* ensures that all elements of the set in which the variable is defined are pairwise distinct. In any other algebra, the *truth* and *falsity laws* would be written in much more cumbersome expressions. Hereinafter, we will refer to the *disjunctive algebra* and its conservative extensions simply as the *finite predicate algebra*.

Thus, in this subsection, we have shown that the finite predicate algebra is a universal mathematical tool for the formal description of deterministic, discrete, finite objects of various kinds.

The convenience of the disjunctive algebra of finite predicates lies in the fact that its language allows for concise and convenient notation of *truth and falsity laws*, which play a special role because they set the necessary and sufficient requirements for the correct introduction of variable attributes on finite sets.

The truth law specifies the domain of variation of a variable, and the falsity law ensures that all elements of the set on which the variable is defined are pairwise different. In any other algebra, the truth and falsity laws would be written as much more cumbersome expressions.

3. Formal description of the subject space in the language of finite predicate algebra

A distinction should be made between the formal description of an object and its abstract description. The abstract description of an object always follows the formal description. This is a stage of more in-depth study of the object compared to its formal description. The formal description of an object is the replacement of the object with mathematical constructions. An abstract description of an object is an axiomatic construction of a theory of those mathematical structures that were obtained in the formal description of the object.

The representation and interpretation of knowledge play an important role in various areas of computer science. Various methods of discrete mathematics are used to formally represent information about objects and processes in knowledge bases. In cases where information about objects and processes in knowledge bases, represented by discrete information features, has a rather complex logical structure, various methods and models of discrete mathematics are used for its formal representation, including logical equations with Boolean variables. Thus, logical methods of image recognition involve the compilation and solution of logical equations with variables that take on values of 1 and 0, depending on whether a given object has a certain property or not. Solving such equations allows either to identify an object by the available sets of attribute variable values or to establish unknown properties of a given object [1].

4. Method for studying the relationships between discrete characteristics of objects

Logical classification methods usually involve compiling and solving logical equations with variables that take on values of 1 and 0, depending on whether a given object has a certain property or not. Solving such equations allows either to identify an object by the available sets of attribute variable values or to establish unknown properties of a given object. A natural generalization of Boolean algebra equations is finite predicate algebra equations, which allow you to operate with arbitrary attribute variables defined on different finite sets. The use of such equations for constructing logical conclusions in knowledge bases allows expanding the capabilities of Boolean logical methods for object recognition and classification. When classifying objects, we deal with sets of features, selecting some of whose values allows us to determine the membership of the object under consideration to a certain class. This section presents a method for investigating the relationships between discrete features of objects. Various types of predicate equations are considered.

When analyzing the relationships between essential features of data, we often encounter rather complex systems of logical equations, which, nevertheless, can be simplified due to their specific properties. To demonstrate the procedure for eliminating non-essential features, a real example from the field of medicine will be considered below.

Many practical problems require the use of logical classification methods. For example, in [16], binary feature vectors are classified. The proposed method can be used in various classification problems in different areas. The classification process boils down to studying logical-dynamic systems depending on some initial states. In [17], correction functions are constructed for logical object recognition methods. An interesting algorithm for logical classification using correction functions is proposed. In [18], logical data classification is used to analyze hyperspectral data. Some combinatorial issues of logical classification are discussed in [19]. The authors applied their research to the classification of proteomic expressions of Alzheimer's disease. In [20], logical algorithmic methods for constructing decision trees are considered. In [21], a logic-based classification method for text recognition is proposed. In [22], a logical approach to machine learning is proposed. The authors developed a special logical classifier. The logical design of a strong classifier using weak classifiers is considered in [23]. A combination of different methods, including logical classification, was tested in [24]. A method for classifying text messages based on logical extraction is considered in [25]. In [26], finite predicate networks for radar detection are investigated. A new approach to logical classification and recognition is proposed in [27]. Subsystems of Boolean equation systems are investigated in [28]. In [29], a method for distributed solving of logical equations is presented. In [30], some specific types of logical functions are considered. The main focus is on symmetric functions, which are widely used for classification purposes. The issue of entropy in Boolean networks is discussed in [31]. Logic-predicate networks are investigated in [32]. The use of predicates for classifying access problems in the Internet of Things is considered in [33]. Many works devoted to classification often use Boolean algebra, fuzzy logic, or neural networks as basic mathematical tools [19]. Logical approaches to fact-based analysis are discussed in [20].

A common generalization of Boolean algebra equations is finite predicate algebra equations [33], which allow operations on arbitrary attribute variables defined on finite sets. The use of such equations for constructing logical inferences in knowledge bases makes it possible to extend the capabilities of logical methods for object recognition [34]. There are many ways to represent data and the relationships between them. First, we will briefly consider Boolean algebra equations and some problems that arise when solving them.

The axioms of Boolean algebra are satisfied by some mathematical structures that are of great importance for information systems. Such structures include, for example, logic algebra, set algebra, and ASP. Often, the dependencies between elements of Boolean algebra can be described by an equation or a system of equations, where one equation (or a certain number of equations) can contain both unknown elements (variables) and known elements (parameters). It is interesting to find out under what conditions an equation (or a system of equations) has a solution, as well as the conditions under which this solution is unique. Finally, it is necessary to be able to find the solution to the equation.

In addition, if the conditions for uniqueness of the solution are not met, it is sometimes impossible to find all solutions to a given equation or system of equations, and it is practically impossible to try them all in turn, since their number can be very large. However, it would be desirable to be able to study these solutions, select the optimal ones (for specific problems), and consider the structure of these solutions in the same way as the structure of a set of differential equations is considered in the theory of differential equations, despite the fact that this set may be infinite. It is possible to find all solutions to Boolean equations without going through them one by one, but by describing them with formulas so that all solutions to a given equation or system of equations can be obtained from these formulas.

5. Analysis of logical connections between essential features of data in systems of predicate equations

A universal method for solving systems of finite predicate algebra equations is to reduce the predicate, given by the system of equations and initial conditions, to a perfect disjunctive normal form. However, this procedure involves searching through many intermediate solutions, and its practical implementation requires significant computer time. For some types of predicate equations, taking into account the peculiarities of their structure, simpler solution algorithms can be developed.

In many practical problems related to the semantic processing of medical data, natural language information, and customer data, it is not necessary to obtain all sets of values of semantic features, but it is necessary to obtain one or more sets of feature values (target variables) that are of interest to the user. It is often necessary to find the values of target variables under given initial conditions, which are a fixed set of values of other features. When solving such problems, other variables that are not included in the initial conditions and are not target variables are excluded from the equation by binding them with existential quantifiers.

Let us consider the solution of Boolean equations using the traditional method of reduction to PDNF (perfect disjunctive normal form). The essence of this method is that the logic algebra function on the left side of the equation is written in PDNF form, or more precisely, the sets of values of binary variables that satisfy this equation are written directly. For example, let us have the following equation

$$\left(X_1 \vee \bar{X}_2\right)\left(X_2 \vee \bar{X}_3\right)=1.$$

Writing down the left side as PDNF, we obtain the equation

$$X_1 X_2 X_3 \vee X_1 X_2 \bar{X}_3 \vee X_1 \bar{X}_2 \bar{X}_3 \vee \bar{X}_1 \bar{X}_2 \bar{X}_3 = 1,$$

which shows that the solutions are sets {1,1,1}, {1,1,0}, {1,0,0}, {0,0,0}.

For logic algebra, this approach is universal in a sense, since it is always possible to obtain all solutions to any equation. Suppose that in the above example, X_1, X_2, X_3 – are variable sets of some set algebra, and we interpret conjunction, disjunction, and negation as intersection, union, and addition in set algebra.

For logic algebra, this approach is universal in a sense, since it is always possible to obtain all solutions to any equation. Suppose that in the above example X_1, X_2, X_3 – variable sets of some set algebra, and we will interpret conjunction, disjunction, and negation as intersection, union, and complement in set algebra.

For clarity, let us consider the algebra of all subsets of a set $\{a,b\}$. Then 0 is an empty set, 1 is all sets. Obviously, the sets listed above will be solutions to the original equation in this case, but not only these sets. For example, if we substitute

$X_1 = \{a\}, X_2 = \{a\}, X_3 = \{a\}$, then the equation becomes an identity. From the above, it follows that the method of reduction to perfect disjunctive normal form is not universal for solving Boolean algebra equations in general.

The above considerations show the need for separate consideration of Boolean algebra as a mathematical tool for solving logical equations. Any results obtained for Boolean algebra equations are automatically transferred to FPA equations.

Let us move on to definitions. It is known that a Boolean algebra is any algebra for which the following axioms hold:

$$a \vee b = b \vee a, a \wedge b = b \wedge a,$$
$$(a \vee b) \vee c = a \vee (b \vee c),$$
$$(a \wedge b) \wedge c = a \wedge (b \wedge c),$$
$$(a \vee b) \wedge c = (a \wedge c) \vee (b \wedge c).$$
$$(a \wedge b) \vee c = (a \vee c) \wedge (b \vee c),$$
$$a \vee a = a, a \wedge a = a,$$
$$\underline{a} \vee 1 = 1, a \vee 0 = a, a \wedge 0, a \wedge 1 = a,$$
$$\overline{\overline{a}} = a, \overline{a \vee b} = \overline{a} \wedge \overline{b}, \overline{a \wedge b} = \overline{a} \vee \overline{b}$$
$$a \vee \overline{a} = 1, a \wedge \overline{a} = 0.$$

Let us assume that the algebra is defined on an abstract set, and the operations $"\vee","\wedge","-"$ will be called conjunction, disjunction, and negation, respectively. A Boolean formula is an arbitrary function $f: M^m \to M$, obtained by superposition of conjunction, disjunction, and negation. Further, for convenience, the conjunction sign will be omitted. The concept of a perfect disjunctive normal form can be defined in exactly the same way as in logic algebra, although it should be noted that in this case the variables in a perfect disjunctive normal form can take values other than 0 and 1.

To effectively solve equations with unknown finite predicates, it is advisable to first study equations written using Boolean algebra operations, since these operations are most commonly encountered when writing predicate equations.

The algebra of finite predicates allows us to interpret knowledge in a rigorous mathematical form, where different features and their meanings are related to each other by Boolean and predicate operations. The classical form of a logical equation with finite predicates looks like this:

$$f(x_1, x_2, \ldots, x_n) = 1,$$

where each variable takes values from a finite set of elements, and in general, these domains may be different. In practice, we may encounter problems with many equations.

In this case, the problem can be solved by solving a system of equations:

$$f_1(x_1, x_2, \ldots, x_n) = 1,$$
$$f_2(x_1, x_2, \ldots, x_n) = 1,$$
$$\ldots$$
$$f_m(x_1, x_2, \ldots, x_n) = 1.$$

If necessary, this system can be rewritten as a conjunction of the above equations and presented as a single equation:

$$f_1(x_1, x_2, \ldots, x_n) \wedge f_2(x_1, x_2, \ldots, x_n) \wedge \ldots \wedge f_m(x_1, x_2, \ldots, x_n) = 1.$$

Unlike Boolean variables, predicate variables provide greater flexibility in defining the necessary characteristics. For example, consider the following dependencies:

$$y^{a_1} \to x_1^{b_1} \vee x_1^{b_2},$$
$$y^{a_2} \to x_1^{b_3} \vee x_1^{b_4},$$
$$y^{a_3} \to x_1^{b_5} \vee x_1^{b_6},$$

where domains for y and x_1 is $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3, b_4, b_5, b_6\}$ accordingly. If $y = a_1$ then $x_1 = b_1$ or $x_1 = b_2$. On the other hand, it follows from the first expression that

$$\neg\left(x_1^{b_1} \vee x_1^{b_2}\right) \to \neg y^{a_1},$$

that means

$$x_1^{b_3} \vee x_1^{b_4} \vee x_1^{b_5} \vee x_1^{b_6} \to y^{a_2} \vee y^{a_3}.$$

Thus, if x_1 takes on a value from a set $\{b_3, b_4, b_5, b_6\}$ then the property of the object y acquires meaning a_2 or a_3.

The classic form of a logical equation with finite predicates looks like this:

$$f(x_1, x_2, \ldots, x_n) = 1,$$

where each variable takes values from a finite set of elements, and in general, these domains may be different. In practice, you may encounter problems with many equations. In this case, the problem can be solved by solving a system of equations:

$$f_1(x_1, x_2, \ldots, x_n) = 1,$$
$$f_2(x_1, x_2, \ldots, x_n) = 1,$$
$$\ldots$$
$$f_m(x_1, x_2, \ldots, x_n) = 1.$$

If necessary, this system can be rewritten as a conjunction of the above equations and presented as a single equation:

$$f_1(x_1, x_2, \ldots, x_n) \wedge f_2(x_1, x_2, \ldots, x_n) \wedge \ldots \wedge f_m(x_1, x_2, \ldots, x_n) = 1.$$

For such equations, we can identify some problems that can be solved:

1) find all possible sets of variable values that satisfy this equation. Obviously, this task is difficult because the number of calculations increases exponentially;

2) determine whether the system has a solution;

3) determine whether it has a unique solution;

4) find some important combinations of variable values that satisfy the system;

5) solve the system under some initial conditions.

Consider the following system of predicate equations:

$$y^{a_1} \to g_1(x_1, x_2, \ldots, x_n),$$
$$y^{a_2} \to g_2(x_1, x_2, \ldots, x_n),$$
$$\ldots$$
$$y^{a_m} \to g_m(x_1, x_2, \ldots, x_n).$$

This means that when the function y takes on the value a_i, the set of variable values $x_1, x_2, \ldots, x_n$ must satisfy the equation

$$g_i(x_1, x_2, \ldots, x_n) = 1,$$

which means that if an object has a property a_i, its characteristics $x_1, x_2, \ldots, x_n$ must satisfy the above formula.

Generally speaking, the opposite is not necessarily true. If $g_i(x_1, x_2, \ldots, x_n) = 1$, y does not necessarily acquire value a_i. Let's consider a stronger dependency:

$$y^{a_1} = g_1(x_1, x_2, \ldots, x_n),$$
$$y^{a_2} = g_2(x_1, x_2, \ldots, x_n),$$
$$\ldots$$
$$y^{a_m} = g_m(x_1, x_2, \ldots, x_n).$$

In this case, any set of feature values $x_1, x_2, \ldots, x_n$ either confirms the fact that y is equal to a_i or not (belongs to the corresponding class or not). If an object has a property a_i, then its features must satisfy $g_i(x_1, x_2, \ldots, x_n) = 1$. Thus, we can classify an object y by the values of its features $x_1, x_2, \ldots, x_n$. It can also be easily shown that the conjunction of any two different functions g_i and g_j is equal to zero. This follows from the basic properties of recognition predicates. It can be shown that the above system is equivalent to the following equation:

$$y^{a_1} g_1(x_1, x_2, \ldots, x_n) \vee y^{a_2} g_2(x_1, x_2, \ldots, x_n) \vee \ldots \vee y^{a_m} g_m(x_1, x_2, \ldots, x_n) = 1.$$

Based on the solution of such equations, it is possible to propose logical methods for classifying objects that can be applied to solve practical problems in various areas. Their features can be identified at the stage of constructing a mathematical model, taking into account the characteristics of the data. Usually, propositional logic is used for this purpose. We propose an approach based on finite predicate algebra. Let us construct a general form of such problems based on predicate equations.

6. Means of constructing a system of features for a knowledge base based on a formal algebraic-logical model of a subject area

Logical methods of object classification are used to solve practical problems in various areas: biology, physics, meteorology, etc. Their features can be identified at the stage of constructing a mathematical model, taking into account the characteristics of the data. Usually, propositional logic is used for this purpose. This paper proposes an approach based on finite predicate algebra.

Let us construct a general form of such problems based on predicate equations. Let the feature variables $y_1, y_2, \ldots, y_l$ denote certain properties of objects, for example, in the formal description of the symptoms of any disease (according to the diagnosis or treatment protocol) or the assignment of a patient to a certain risk class [15]. Each variable takes values from its domain. Unlike Boolean variables, predicate variables can take values from different arbitrary domains.

Let discrete variables $x_1, x_2, \ldots, x_n$ be attributes, based on which we can determine what values variable properties can take. Properties and attributes can be linked together in the form of complex logical dependencies, which can be represented as a predicate equation:

$$P(y_1, y_2, \ldots, y_l;\ x_1, x_2, \ldots, x_n) = 1,$$

where P is a finite predicate.

Classifying the object under consideration means determining, based on this predicate equation and experimental data on features $x_1, x_2, \ldots, x_n$, which properties (feature values $y_1, y_2, \ldots, y_l$) this object possesses and which properties it does not satisfy. For example, each elementary conjunction

$$x_1^{a_{11}} x_2^{a_{12}} \ldots x_n^{a_{1n}},$$

$$x_1^{a_{21}} x_2^{a_{22}} \ldots x_n^{a_{2n}},$$

$$\ldots$$

$$x_1^{a_{m1}} x_2^{a_{m2}} \ldots x_n^{mn}$$

corresponds to its own class of objects. Then, based on a priori dependence $P(y_1, y_2, \dots, y_l;\ x_1, x_2, \dots, x_n) = 1$ and experimental data on features $x_1, x_2, \dots, x_n$, it is possible to determine which class a given object belongs to. As can be seen from the above considerations, the values of features are grouped into a matrix.

Suppose that as a result of the experiment, we obtained some data related to the values of features $x_1, x_2, \dots, x_n$, describing the classified object and constructed the following predicate equation describing the relationships between them:

$$g(x_1, x_2, \dots, x_n) = 1.$$

The task of classifying objects can be formalized as solving the following predicate equation by finding the unknown predicate f :

$$g(x_1, x_2, \dots, x_n) \to f(y_1, y_2, \dots, y_l).$$

solving this functional equation, you can determine the values of the features $x_1, x_2, \dots, x_n$, that characterize objects $y_1, y_2, \dots, y_l$.

Unlike Boolean variables, predicate variables provide greater flexibility in defining the necessary characteristics. For example, consider the following dependencies:

$$y^{a_1} \supset x_1^{b_1} \vee x_1^{b_2},$$

$$y^{a_2} \supset x_1^{b_3} \vee x_1^{b_4},$$

$$y^{a_3} \supset x_1^{b_5} \vee x_1^{b_6},$$

where the domains for y and x_1 are $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3, b_4, b_5, b_6\}$ accordingly. If $y = a_1$, then $x_1 = b_1$ or $x_1 = b_2$. On the other hand, it follows from the first expression that

$$\neg\left(x_1^{b_1} \vee x_1^{b_2}\right) \supset \neg y^{a_1},$$

that means

$$x_1^{b_3} \vee x_1^{b_4} \vee x_1^{b_5} \vee x_1^{b_6} \supset y^{a_2} \vee y^{a_3}.$$

Thus if x_1 takes a value from the set $\{b_3, b_4, b_5, b_6\}$, the property of object y takes the value a_2 or a_3.

A universal method for solving FPA systems of equations is to reduce the predicate, given by the system of equations and initial conditions, to a perfect disjunctive normal form (PDNF). However, this procedure involves searching through a large number of intermediate solutions, and its practical implementation requires significant computer time. For some types of linguistic equations, given the peculiarities of their structure, simpler algorithms for solving them can be developed. For example, a heuristic algorithm is more practical, but it involves searching through all sets of semantic feature values, which slows down the solution process as the number of features increases, with the solution time growing exponentially.

In many practical problems related to knowledge interpretation, which are mostly presented in text form, i.e., related to the semantic processing of natural language information, it is not necessary to obtain all sets of values of semantic features, but it is necessary to obtain one or more sets of feature values (target variables) that are of interest to the user. It is often necessary to find the values of target variables under given initial conditions, which are a fixed set of values of other features. When solving such problems, other variables that are not included in the initial conditions and are not target variables are excluded from the equation by linking them with existential quantifiers.

In research related to logical inferences in knowledge bases, questions arise regarding the determination of the closeness of the relationships between the attributes of these objects, as well as questions about their essentiality and non-essentiality. Obviously, it can be assumed that the formal relationship between features is stronger when fewer sets of values of these variables satisfy the equation. At the same time, if any sets of values of these variables satisfy the original equation, it can be assumed that there is no relationship between these variables.

In addition, the following questions arise when solving practical problems:

1. How will the specific values of this feature, substituted into a logical equation, affect the relationships between other features?

2. How strong is the logical relationship between two (or more) given features?

To answer the first question, it is natural to single out those predicates (and, accordingly, equations) which, when a certain value of the attribute is substituted, turn into predicates that give a stronger relationship between the variables, as well as those predicates, the substitution of this value into which leads to a weakening of the logical relationship between the attributes.

To answer the second question, it is necessary to exclude from the original equation, using the existence quantifier, all variables except those under consideration, and examine the resulting equation with fewer variables, which describes all permissible sets of values of the studied features.

7. Development of an algorithm for eliminating insignificant variables when solving algebraic logic equations to reduce the amount of necessary calculations

Let us consider the feature selection procedure, where the number of features can be reduced. Here we may encounter the following problems.

1. We may need to find some sets of feature values that interest us, where there is at least one value of non-essential features for which there is at least one set of values of essential features. In this case, we apply an existence quantifier to the set of non-essential values.

2. We may need to find some sets of feature values where, for any set of non-essential features, there is at least one solution to the equation. In this case, we apply a universal quantifier to the non-essential variables.

3. We may need to find some sets of attribute values that satisfy the equation, provided that the non-essential attributes take certain specific values.

Let the predicate P depend on variables $x, y, \ldots, z$. Let us define the substitution operator $a(P)$ (a belongs to the domain of the variable x), which acts on the predicate P as follows

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

We will call a substitution operator restrictive if the following condition is met

$$P(a, y, \ldots, z) \to P(x, y, \ldots, z)$$

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

Let's determine the distribution of the substitution operator if the condition is true

$$P(a, y, \ldots, z) \leftarrow P(x, y, \ldots, z)$$

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

Interpreting the knowledge presented by this implication, we can say that substitution operators strengthen the logical connection between discrete features, while distributive substitution operators weaken this connection by shifting the relationship between features in an irrelevant manner.

Let us consider the predicate P as follows:

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

Then

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

Obviously, the predicate $a_1(P)$ will be shortened if $P_1 \to P_i \ \forall\, i = 1, 2, \ldots, n$.

The operator $a_1(P)$ will distribute if $P_1 \leftarrow P_i \ \forall \, i = 1, 2, \ldots, n$.

Let us consider examples of applying the operator a_1 to predicate $P(x, y)$ where variables x, y and z have domains of definition $\{a_1, a_2\}$, $\{b_1, b_2\}$ and $\{c_1, c_2\}$ accordingly.

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

Then $a_1(P) = y^{b_1} z^{c_1} = \left(x^{a_1} \vee x^{a_2} \right) \& \, y^{b_1} z^{c_1} = x^{a_1} y^{b_1} z^{c_1} \vee x^{a_2} y^{b_1} z^{c_1}$.

With the exception of disjunctions containing the predicate $a_1(P)$, it contains one more disjunction $x^{a_2} y^{b_1} c^{c_1}$, thus the operator a_1 is restrictive for the predicate P. According to the definitions given, in the example above $P_1 = y^{b_1} z^{c_1}$, $P_2 = y^{b_1} z^{c_2} \vee y^{b_1} z_1^{c_1}$. From this, it is clear that $P_1 \rightarrow P_2$. Let us now consider the predicate

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

$$a_1(P) = y^{b_1} z^{c_1} \vee y^{b_1} z^{c_2} = \left(x^{a_1} \vee x^{a_2} \right) \& \left(y^{b_1} z^{c_1} \vee y^{b_1} z^{c_2} \right) =$$

$$= x^{a_1} y_1 b_1 c_1 \vee x^{a_2} y^{b_1} z^{c_2} \vee x^{a_2} y^{b_1} z^{c_1} \vee x x^{a_2} b_1 b^{c_2}.$$

The operator a_1 for this predicate is obviously distributive. In this example

$$P_1 = y^{b_1} z^{c_1} \vee y^{b_1} z^{c_2} \text{ and } P_2 = y^{b_1} z^{c_2} \text{ thus } P_1 \leftarrow P_2.$$

To answer the second question, it is necessary to exclude all variables except those considered from the original equation and examine the resulting equation with fewer variables, which describes all permissible sets of attribute values. In [51], a fairly broad class of predicates is considered for which an effective algorithm for excluding variables without increasing the size of the original formula can be specified. Here, we extend this class by adding some additional properties. Consider the following properties of the existence quantifier:

1. $\exists x x^{a} = 1$.
2. $\exists x \neg x^{a} = 1$.
3. $\exists x \left(\neg \left(P(x) Q(x) \right) \right) = \exists x \neg P(x) \vee \exists x \neg Q(x)$.
4. $\exists x \left(P(x) \vee Q(x) \right) = \exists x P(x) \vee \exists x Q(x)$.
5. $\exists x (P(x) \& Q(y) = \exists x P(x) \& Q(y)$.
6. $\exists y \left(P(x) \rightarrow Q(y) \right) = P(x) \rightarrow \exists y Q(y)$.

7. $\exists y\left(P(x)\to Q(y)\right)=P(x)\to \exists yQ(y)$.

8. Assume $P_i(x)\,\&\,P_j(x)=0, i\neq j, i, j=1,2,\ldots,k$

9. Suppose:

$$\exists y\left(\left(P_1(x)\to Q_1(y)\right)\&\left(P_2(x)\to Q_2(y)\right)\&\ldots\&\left(P_k(x)\to Q_k(y)\right)\right)=$$
$$=\left(P_1(x)\to \exists yQ_1(y)\right)\&\left(P_2(x)\to \exists yQ_2(y)\right)\&\ldots\&\left(P_k(x)\to \exists yQ_k(y)\right).$$

10. If the identity $P_i(x)\equiv 0$ is not true for any $i=1,2,\ldots,k$ and $P_i(x)\,\&\,P_j(x)=0$ for $i\neq j, i, j=1,2,\ldots,k$, then

$$\exists x\left(\left(P_1(x)\to Q_1(y)\right)\&\left(P_2(x)\to Q_2(y)\right)\&\ldots\&\left(P_k(x)\to Q_k(y)\right)\right)=$$
$$=Q_1(y)\vee Q_2(y)\vee\ldots\vee Q_k(y)$$

The above properties allow us to describe a broad class of finite predicates (according to equations) defined on a set of variables $\{x, y,\ldots, z\}$, for which it is easy to find relationships between selected variables without increasing the size of the original formulas. Let us define this class recursively.

1. All «recognitions» for variable $x^a, x^b,\ldots, x^c$ ($a,b,\ldots,c$ – symbols belonging to the domain x) belong to Σ_x.

2. All objections $\neg x^a,\neg x^b,\ldots,\neg x^c$ belong to Σ_x.

3. If predicates $\neg P(x),\neg Q(x)$ belong to Σ_x then predicate $\neg\left(P(x)Q(x)\right)$ belongs to Σ_x.

4. Any predicate that does not depend on a variable x belongs to the set Σ_x.

5. If predicates P_1 та P_2 belong to Σ_x then predicate $P=P_1\vee P_2$ belongs to Σ_x.

6. If predicate P_1 belongs to Σ_x and predicate P_2 does not depend on x then predicate $P=P_1\,\&\,P_2$ belongs to Σ_x.

7. If predicate P_1 does not depend on x i predicate P_2 belongs to Σ_x then predicate $P=P_1\to P_2$ belongs to Σ_x.

8. Let predicates $P_1, P_2,\ldots, P_k$ do not depend on x; $P_i\,\&\,P_j=0$ for $i\neq j, i, j=1,2,\ldots,k$, predicates $Q_1, Q_2,\ldots, Q_k$ belong to Σ_x; then

$$P=\left(P_1\to Q_1\right)\&\left(P_2\to Q_2\right)\&\ldots\&\left(P_k\to Q_k\right)$$

belongs to Σ_x.

9. If predicates $P_1, P_2, \ldots, P_k$ depend only on x, $P_i \& P_j = 0$ for $i \neq j, i, j = 1, 2, \ldots, k$; for any $i = 1, 2, \ldots, k$ then identity $P_i \equiv 0$ is not true; predicates $Q_1, Q_2, \ldots, Q_k$ do not depend on x; then predicate

$$P = (P_1 \to Q_1) \& (P_2 \to Q_2) \& \ldots \& (P_k \to Q_k)$$

belongs to Σ_x.

It may also be necessary to eliminate redundant variables using a universal quantifier. In this case, we can use the following properties of this quantifier:

1. $\forall x x^a = 0$.
2. $\forall x \neg x^a = 0$.
   $\forall x \neg (P(x) \vee Q(x)) = \forall x \neg P(x) \& \forall x \neg Q(x)$
3. $\forall x (P(x) \& Q(x)) = \forall x P(x) \& \forall x Q(x)$.
4. $\forall x (P(x) \vee Q(y) = \forall x P(x) \vee Q(y)$.
5. $\forall y (P(x) \& Q(y)) = P(x) \& \forall y Q(y)$.
6. Suppose.

$$P_i(x) \& P_j(x) = 0,\ i \neq j, i, j = 1, 2, \ldots, k,$$

then:

$$\forall y \big((P_1(x) \& Q_1(y)) \vee (P_2(x) \& Q_2(y)) \vee \ldots \vee (P_k(x) \& Q_k(y)) \big) =$$
$$= (P_1(x) \& \forall y Q_1(y)) \vee (P_2(x) \& \forall y Q_2(y)) \vee \ldots \vee (P_k(x) \& \forall y Q_k(y)).$$

7. If identity $P_i(x) \equiv 0$ is not true for any $i = 1, 2, \ldots, k$ and $P_i(x) \& P_j(x) = 0$ for $i \neq j, j = 1, 2, \ldots, k$ then

$$\forall x \big((P_1(x) \& Q_1(y)) \vee (P_2(x) \& Q_2(y)) \vee \ldots \vee (P_k(x) \& Q_k(y)) \big) =$$
$$= Q_1(y) \& Q_2(y) \& \ldots \& Q_k(y).$$

We can recursively define a class of predicates Σ_x from which a variable x can be excluded without increasing the size of the formula:

1. All «recognitions» $x^a, x^b, \ldots, x^c$ belong to Σ_x.
2. All objections $\neg x^a, \neg x^b, \ldots, \neg x^c$ that do not depend on x belong to Σ_x.
3. If $\neg P_1$ and $\neg P_2$ belong to Σ_x then $\neg (P_1 \vee P_2)$ belongs to Σ_x.
4. If predicates P_1 and P_2 belong to Σ_x then predicate $P = P_1 \& P_2$ belongs to Σ_x.

5. If a P_1 belongs to Σ_x and predicate P_2 does not depend on x then predicate $P = P_1 \vee P_2$ belongs to Σ_x.

6. If predicate P_1 does not depend on x and predicate P_2 belongs to Σ_x then predicate $P = P_1 \,\&\, P_2$ belongs to Σ_x.

7. Let's assume that predicates $P_1, P_2, \ldots, P_k$ do not depend on x, $P_i \,\&\, P_j = 0$ for $i \neq j, i, j = 1, 2, \ldots, k$; predicates $Q_1, Q_2, \ldots, Q_k$ belong to Σ_x, then

$$P = (P_1 \,\&\, Q_1) \vee (P_2 \,\&\, Q_2) \vee \ldots \vee (P_k \,\&\, Q_k)$$

belongs to Σ_x.

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

An example of how this method is used is shown in the next paragraph based on information screening of medical records.

8. Application of algebraic-logical modeling for information screening in conditions of incomplete information

All areas that work with knowledge bases use information technology and produce or consume data. The quality of this data is crucial for the effectiveness of decision-making support processes. Therefore, the relevant areas of IT development are: interpretation of knowledge, its extraction from various sources, intelligent data processing, and the formation of high-quality information for making appropriate decisions. Information quality is an integral characteristic that shows the degree of suitability of data for decision-making. The process of improving information quality, or information screening, is a process of semantic processing of poorly structured, incomplete, and ambiguous data in order to analyze characteristics and identify patterns and inconsistencies [15].

For example, consider the task of analyzing certain medical records, namely data stored in patient medical records, and show that such data is generally characterized by the following features: incompleteness, obsolescence, inconsistency, etc. This situation leads to more than one interpretation, i.e., to ambiguity of information. The use of the results of processing ambiguous information contained in medical documentation with low data quality leads to incorrect management and medical decisions. In this regard, it is important to introduce models and methods for processing medical data that improve the quality of information for decision-making, namely: improve data completeness by extracting additional information, improve data accuracy by using only relevant information, remove contradictory information, and form a relevant data set.

Research methods for this type of task are based on the use of system analysis methods, decision-making theory, intelligent data analysis methods, and intelligence theory methods. Namely: the algebraic-logical modeling method and the comparator identification method are used in the intelligent processing of data from patient medical records for information screening of medical documentation; the mathematical apparatus of optimization methods is used in solving the problem of planning therapeutic and preventive measures based on the technology of information screening of medical documentation; a service-oriented approach was used to develop tools for solving the tasks set.

In works [15, 53], a method for information screening of medical documentation was proposed and developed. The use of the comparator identification method for solving the problem of information screening of medical documentation was also proposed. A model for identifying medical and diagnostic parameters has been developed.

For example, to improve the quality of information for medical decision-making, a method of information screening has been formulated and justified, which consists of using a set of data processing models (Fig. 1).

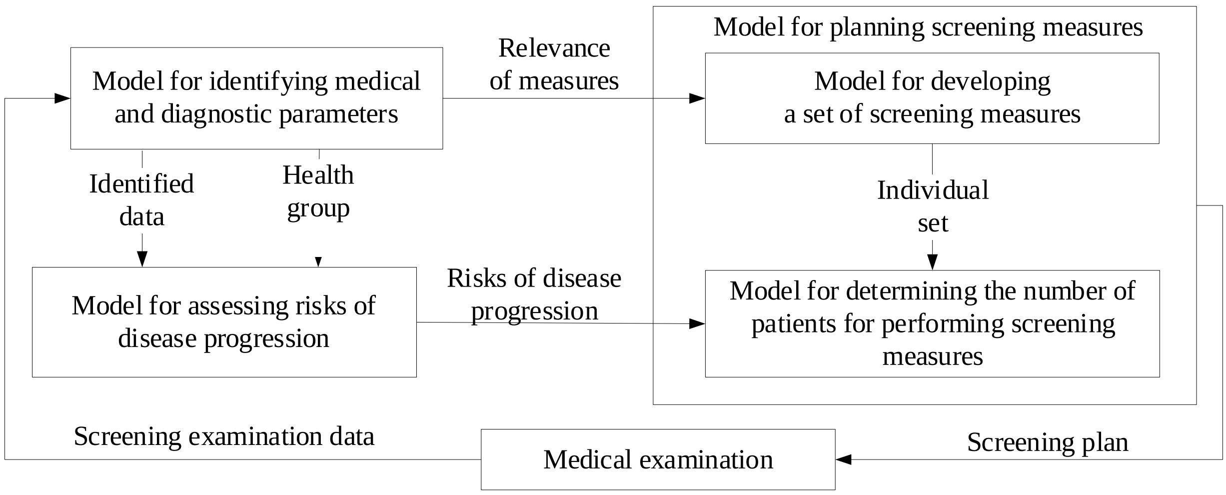

Fig. 1. Medical documentation information screening method

To solve the problem of information screening of certain medical documentation (for example, medical records filled out by a family doctor), the use of an algebraic-logical method of modeling a system of features based on finite predicate algebra is proposed as a method for modeling subject knowledge. The general scheme for using the proposed method is as follows: based on information from medical records and other sources of medical information, a set of features is formed, which is modeled by predicate equations. Depending on the results to be obtained, the reactions of the predicates are subjected to semantic processing and grouped into a set of aggregated signs, which subsequently form the basis for medical decisions (Fig. 1). In this formulation of the problem, the comparator is a system of predicate equations, the solution of which is described above.

One of the main conditions for the application of the comparator identification method is the discreteness, finiteness, and determinism of the objects of the subject space. In this case, the subject space U is a Cartesian product of a set of objects from the subject space $U_1 \times U_2 \times ... \times U_n$: medical data, patient records, clinical monitoring data, etc. With the help of a predicate $P(x_1, x_2, \ldots, x_m)$ any ratio that is defined at the subject space $U = U_1 \times U_2 \times ... \times U_n$ is described. In the language of finite predicate algebra, this predicate is formalized using the recognition predicate and the basic operations of conjunction and disjunction. The recognition predicate models the ability of a person to unambiguously assign the object under consideration to one of two classes. The predicate is equal to 1 when the object is recognized and 0 otherwise. The introduced predicate variables are linked by logical equations, the joint solution of which assigns the elements of the set of objects under consideration to a certain class, i.e., determines the patient’s risk group for the presence of certain diseases.

The following experiment plan is proposed. We use real medical data and encode it using predicate equations. Note that although some variables may take on the value «unknown», this is still a closed-world case, since «unknown» means only a value from the alphabet in which the variable is defined. Each domain for any variable is closed. After we have written a system of equations with the help of experts, we begin to remove variables that we consider irrelevant at the moment. This does not mean that in other cases other variables will be considered irrelevant. Important variables are those for which we want to determine logical relationships. The result is an equation from which irrelevant variables have been removed. The resulting equation is simpler than the original system, and it is easier to analyze the relationships between relevant variables.

If we consider information screening of medical data for the assessment of the development and prevention of cardiovascular diseases [22], we can identify a set of features for formalizing screening procedures. Let us consider the following features and their meanings:

Gender: $X_1 = \left\{x_1^1, x_1^2\right\}$, where x_1^1 means female, x_1^1 means male.

Age: $X_2 = \left\{x_2^1, x_2^2, x_2^3\right\}$, where x_2^1 under 40 years, x_2^2 from 40 to 50 years, x_2^3 over 50 years.

Diabetes mellitus: $X_3 = \left\{x_3^1, x_3^2, x_3^3, x_3^4\right\}$, where x_3^1 – yes, x_3^2 – no (actual diagnosis), x_3^3 – no (diagnosis not determined), x_3^4 – unknown.

Arterial hypertension: $X_4 = \left\{x_4^1, x_4^2, x_4^3, x_4^4\right\}$, where x_4^1 – yes, x_4^2 – no (actual diagnosis), x_4^3 – no (diagnosis not determined), x_4^4 – unknown.

Kidney problems: $X_5 = \left\{x_5^1, x_5^2, x_5^3\right\}$ where x_5^1 – yes, x_5^2 – no, x_5^3 – unknown.

Tachycardia: $X_6 = \left\{x_6^1, x_6^2, x_6^3, x_6^4, x_6^5\right\}$, where x_6^1 – yes (actual diagnosis), x_6^2 – yes (diagnosis not determined), yes (diagnosis not determined), x_6^3 – no (actual diagnosis), x_6^4 – no (diagnosis not determined), x_6^5 – unknown.

Heredity in relation to heart and vascular diseases: $X_7 = \left\{x_7^1, x_7^2, x_7^3\right\}$, where x_7^1 – yes, x_7^2 – no, x_7^3 – unknown.

Smoking: $X_8 = \left\{x_8^1, x_8^2, x_8^3\right\}$, where x_8^1 – yes, x_8^2 – no, x_8^3 – unknown.

Alcohol problems: $X_9 = \left\{x_9^1, x_9^2, x_9^3\right\}$, where x_9^1 – yes, x_9^2 – no, x_9^3 – unknown.

Hypodynamia: $X_{10} = \left\{x_{10}^1, x_{10}^2, x_{10}^3\right\}$, where x_{10}^1 – yes, x_{10}^2 – no, x_{10}^3 – unknown.

These features make it possible to develop a model for identifying diagnostic parameters, which can be used to determine the patient's health group $R=\{r_1,r_2,r_3,r_4\}$ where r_1 low risk of heart and vascular disease, r_2 medium risk, r_3 high risk, r_4 very high risk.

A set of aggregated characteristics Q_1-Q_3 is used to determine the health group, where Q_1 is denoted by X_1 and X_2, Q_2 is denoted by X_7 to X_{10}, Q_3 is denoted by $X_3 - X_6$.

The values for each health group and each aggregated indicator are divided into four classes in accordance with the relevant medical and technological documentation (standardized clinical protocol and local protocols relating to the prevention of heart and vascular diseases).

The system of equations for forming, for example, a feature Q_2 looks like this:

$$\begin{cases}
q_2^1 = x_7^2 x_8^2\left(x_9^2 \vee x_9^3\left(x_{10}^2 \vee x_{10}^3\right)\right) \vee x_7^2 x_8^3 x_9^2 x_{10}^2 \vee x_7^3 x_8^2 x_{10}^2\left(x_9^2 \vee x_9^3\right), \\
q_2^2 = x_7^2\left(x_8^1\left(x_9^1 x_{10}^2 \vee x_9^2\right) \vee x_9^3\left(x_8^1 x_{10}^2 \vee x_8^2 x_{10}^1\right)\right) \vee \\
\quad \vee\left(x_7^2\left(x_8^2 x_9^1 \vee x_8^3 x_9^2\right) \vee\left(x_7^2 x_9^3 \vee x_7^3 x_9^1\right) x_8^3 x_{10}^2 \vee\right. \\
\quad \vee\left(x_7^2 x_8^3 \vee x_7^3 x_8^2\right) x_9^1\left(x_{10}^2 \vee x_{10}^3\right) \vee x_7^3 x_8^2\left.\left(x_9^2 \vee x_9^3\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee \\
\quad \vee x_7^3 x_8^1\left(x_9^2 x_{10}^2 \vee x_9^3\right) \vee x_7^3 x_8^3 x_9^2\left(x_{10}^1 \vee x_{10}^2\right), \\
q_2^3 = x_7^1 x_{10}^2\left(x_8^1 x_9^2 \vee x_8^2\left(x_9^1 \vee x_9^2\right)\right) \vee \\
\quad \vee\left(x_7^1 x_9^3\left(x_8^1 \vee x_8^2\right) \vee\left(x_7^1 x_8^3 \vee x_7^3 x_8^1\right) x_9^1\right)\left(x_{10}^2 \vee x_{10}^3\right) \vee x_7^1 x_8^3\left(x_9^2 \vee x_9^3\right) \vee \\
\quad \vee\left(x_7^2\left(x_8^1\left(x_9^1 \vee x_9^3\right) \vee x_8^3 x_9^3\right) \vee\left(x_7^2 x_8^3 \vee x_7^3 x_8^2\right) x_9^1 x_{10}^1 \vee\right. \\
\quad \vee x_7^3\left.\left(x_8^1 x_9^2 \vee x_8^3 x_9^1\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee x_7^3 x_8^3\left(x_9^2 x_{10}^3 \vee x_9^3\right), \\
q_2^4 = x_7^1 x_9^3 x_{10}^1\left(x_8^1 \vee x_8^2\right) \vee\left(x_7^1 x_8^3 \vee x_7^3 x_8^1\right) x_9^1 x_{10}^1 \vee \\
\quad \vee\left(x_7^1 x_8^2 x_9^1 \vee x_7^1 x_9^2\left(x_8^1 \vee x_8^2\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee x_7^1 x_8^1 x_9^1.
\end{cases}$$

A model for identifying medical and diagnostic parameters using the example of risk class determination is presented in the form of a system of predicate equations:

$$
\begin{cases}
r_1 = q_1^1 q_2^1 (q_3^1 \vee q_3^2) \vee (q_1^1 q_2^2 \vee (q_1^2 \vee q_1^3) q_2^1) q_3^1, \\
r_2 = q_1^1 \left(q_2^1 q_3^3 \vee q_2^2 q_3^2 \right) \vee \\
\quad \vee \left(q_1^1 \left(q_2^3 \vee q_2^4 \right) \vee q_1^2 \left(q_2^2 \vee q_2^3 \right) \vee q_1^3 q_2^2 \vee q_1^4 \left(q_2^1 \vee q_2^2 \right) \right) \left(q_3^1 \vee q_3^2 \right) \vee \\
\quad \vee \left(q_1^2 \vee q_1^3 \right) q_2^1 \left(q_3^2 \vee q_3^3 \right) \vee \left(q_1^2 q_2^4 \vee \left(q_1^3 \vee q_1^4 \right) q_2^3 \right) q_3^1, \\
r_3 = q_2^1 q_3^4 \vee \left(q_1^1 \vee q_1^2 \vee q_1^3 \right) \left(q_2^2 \vee q_2^3 \right) \left(q_3^3 \vee q_3^4 \right) \vee \\
\quad \vee q_1^3 q_3^2 (\left(q_2^3 \vee q_2^4 \right) \vee \left(q_1^3 \vee q_1^4 \right) q_2^4 q_3^1 \vee \\
\quad \vee \left(q_1^1 q_2^4 \vee q_1^4 \left(q_2^1 \vee q_2^2 \right) \right) q_3^3 \vee \left(q_1^2 q_2^4 \vee q_1^4 q_2^3 \right) \left(q_3^2 \vee q_3^3 \right), \\
r_4 = \left(q_1^1 \vee q_1^2 \right) q_2^4 q_3^4 \vee q_1^3 q_2^4 \left(q_3^3 \vee q_3^4 \right) \vee \\
\quad \vee q_1^4 q_3^4 \left(q_2^2 \vee q_2^3 \right) \vee q_1^4 q_2^4 \left(q_3^2 \vee q_3^3 \vee q_3^4 \right).
\end{cases}
$$

The final classification can be described by the following system:

$$
\begin{cases}
r_1 = q_1^1 q_2^1 \left(q_3^1 \vee q_3^2 \right) \vee \left(q_1^1 q_2^2 \vee \left(q_1^2 \vee q_1^3 \right) q_2^1 \right) q_3^1 , \\
r_2 = q_1^1 \left(q_2^1 q_3^3 \vee q_2^2 q_3^2 \right) \vee \\
\quad \vee \left(q_1^1 \left(q_2^3 \vee q_2^4 \right) \vee q_1^2 \left(q_2^2 \vee q_2^3 \right) \vee q_1^3 q_2^2 \vee q_1^4 \left(q_2^1 \vee q_2^2 \right) \right) \left(q_3^1 \vee q_3^2 \right) \vee \\
\quad \vee \left(q_1^2 \vee q_1^3 \right) q_2^1 \left(q_3^2 \vee q_3^3 \right) \vee \left(q_1^2 q_2^4 \vee \left(q_1^3 \vee q_1^4 \right) q_2^3 \right) q_3^1 , \\
r_3 = q_2^1 q_3^4 \vee \left(q_1^1 \vee q_1^2 \vee q_1^3 \right) \left(q_2^2 \vee q_2^3 \right) \left(q_3^3 \vee q_3^4 \right) \vee \\
\quad \vee q_1^3 q_3^2 \left(q_2^3 \vee q_2^4 \right) \vee \left(q_1^3 \vee q_1^4 \right) q_2^4 q_3^1 \vee \\
\quad \vee \left(q_1^1 q_2^4 \vee q_1^4 \left(q_2^1 \vee q_2^2 \right) \right) q_3^3 \vee \left(q_1^2 q_2^4 \vee q_1^4 q_2^3 \right) \left(q_3^2 \vee q_3^3 \right), \\
r_4 = \left(q_1^1 \vee q_1^2 \right) q_2^4 q_3^4 \vee q_1^3 q_2^4 \left(q_3^3 \vee q_3^4 \right) \vee q_1^4 q_3^4 \left(q_2^2 \vee q_2^3 \right) \vee q_1^4 q_2^4 \left(q_3^2 \vee q_3^3 \vee q_3^4 \right)
\end{cases}
$$

Let's explore the logical connections between discrete features $x_1 - x_{10}$. First, let's rewrite the system of predicate equations in the following form:

$$
\begin{aligned}
&P\left(q_2, x_1, \ldots, x_{10}\right) = \\
&= q_2^1\left(x_7^2 x_8^2\left(x_9^2 \vee x_9^3\left(x_{10}^2 \vee x_{10}^3\right)\right) \vee x_7^2 x_8^3 x_9^2 x_{10}^2 \vee x_7^3 x_8^2 x_{10}^2\left(x_9^2 \vee x_9^3\right)\right) \vee \\
&\vee q_2^2\left(x_7^2\left(x_8^1\left(x_9^1 x_{10}^2 \vee x_9^2\right) \vee x_9^3\left(x_8^1 x_{10}^2 \vee x_8^2 x_{10}^1\right)\right) \vee \right. \\
&\vee \left(x_7^2\left(x_8^2 x_9^1 \vee x_8^3 x_9^2\right) \vee \left(x_7^2 x_9^3 \vee x_7^3 x_9^1\right) x_8^3 x_{10}^2 \vee \right. \\
&\vee \left.\left(x_7^2 x_8^3 \vee x_7^3 x_8^2\right) x_9^1\left(x_{10}^2 \vee x_{10}^3\right) \vee x_7^3 x_8^2\left(x_9^2 \vee x_9^3\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee \\
&\vee \left. x_7^3 x_8^1\left(x_9^2 x_{10}^2 \vee x_9^3\right) \vee x_7^3 x_8^3 x_9^2\left(x_{10}^1 \vee x_{10}^2\right)\right) \vee \\
&\vee q_2^3\left(x_7^1 x_{10}^2\left(x_8^1 x_9^2 \vee x_8^2\left(x_9^1 \vee x_9^2\right)\right) \vee \right. \\
&\vee \left(x_7^1 x_9^3\left(x_8^1 \vee x_8^2\right) \vee \left(x_7^1 x_8^3 \vee x_7^3 x_8^1\right) x_9^1\right)\left(x_{10}^2 \vee x_{10}^3\right) \vee x_7^1 x_8^3\left(x_9^2 \vee x_9^3\right) \vee \\
&\vee \left(x_7^2\left(x_8^1\left(x_9^1 \vee x_9^3\right) \vee x_8^3 x_9^3\right) \vee \left(x_7^2 x_8^3 \vee x_7^3 x_8^2\right) x_9^1 x_{10}^1 \vee \right. \\
&\vee \left. \left. x_7^3\left(x_8^1 x_9^2 \vee x_8^3 x_9^1\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee x_7^3 x_8^3\left(x_9^2 x_{10}^3 \vee x_9^3\right)\right) \vee \\
&\vee q_2^4\left(x_7^1 x_9^3 x_{10}^1\left(x_8^1 \vee x_8^2\right) \vee \left(x_7^1 x_8^3 \vee x_7^3 x_8^1\right) x_9^1 x_{10}^1 \vee \right. \\
&\vee \left. \left(x_7^1 x_8^2 x_9^1 \vee x_7^1 x_9^2\left(x_8^1 \vee x_8^2\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee x_7^1 x_8^1 x_9^1\right) = 1
\end{aligned}
$$

It is clear that this predicate belongs to the class Δ_{x7}. Let us examine the relationship between all variables except x_7. This exclusion will give us the relationship between the variables q_2, x_1, ..., x_6, x_8, x_9, x_{10}:

$$F = \exists x_7 P\left(q_2, x_1, \ldots, x_{10}\right) =$$
$$= q_2^1\left(x_8^2\left(x_9^2 \vee x_9^3\left(x_{10}^2 \vee x_{10}^3\right)\right) \vee x_8^3 x_9^2 x_{10}^2 \vee x_8^2 x_{10}^2\left(x_9^2 \vee x_9^3\right)\right) \vee$$
$$\vee q_2^2\left(x_8^1\left(x_9^1 x_{10}^2 \vee x_9^2\right) \vee x_9^3\left(x_8^1 x_{10}^2 \vee x_8^2 x_{10}^1\right)\right) \vee \left(\left(x_8^2 x_9^1 \vee x_8^3 x_9^2\right) \vee \left(x_9^3 \vee x_9^1\right) x_8^3 x_{10}^2 \vee\right.$$
$$\left.\vee\left(x_8^3 \vee x_8^2\right) x_9^1\left(x_{10}^2 \vee x_{10}^3\right) \vee x_8^2\left(x_9^2 \vee x_9^3\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee x_8^1\left(x_9^2 x_{10}^2 \vee x_9^3\right) \vee$$
$$\vee x_8^3 x_9^2\left(x_{10}^1 \vee x_{10}^2\right) \vee$$
$$\vee q_2^3\left(x_{10}^2\left(x_8^1 x_9^2 \vee x_8^2\left(x_9^1 \vee x_9^2\right)\right) \vee \left(x_9^3\left(x_8^1 \vee x_8^2\right) \vee \left(x_8^3 \vee x_7^3 x_8^1\right) x_9^1\right)\left(x_{10}^2 \vee x_{10}^3\right) \vee\right.$$
$$\vee x_8^3\left(x_9^2 \vee x_9^3\right) \vee \left(\left(x_8^1\left(x_9^1 \vee x_9^3\right) \vee x_8^3 x_9^3\right) \vee \left(x_8^3 \vee x_8^2\right) x_9^1 x_{10}^1 \vee\right.$$
$$\left.\left.\vee\left(x_8^1 x_9^2 \vee x_8^3 x_9^1\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee x_8^3\left(x_9^2 x_{10}^3 \vee x_9^3\right)\right) \vee$$
$$\vee q_2^4\left(x_9^3 x_{10}^1\left(x_8^1 \vee x_8^2\right) \vee \left(x_8^3 \vee x_8^1\right) x_9^1 x_{10}^1 \vee\right.$$
$$\left.\vee\left(x_8^2 x_9^1 \vee x_9^2\left(x_8^1 \vee x_8^2\right)\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee x_8^1 x_9^1\Big) = 1$$

It should be noted that the size of the output formula has not increased, which is explained by the fact that the predicate $P\left(q_2, x_1, \ldots, x_{10}\right)$ belongs to Σ_{x7}.

Suppose we are interested in the link between q_2, x_9, x_{10}. We remove other features $F\left(x_1, \ldots, x_{10}\right)$ from the predicate:

$$
\begin{aligned}
& G\left(q_2, x_9, x_{10}\right) = \exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists x_5 \exists x_6 \exists x_7 \exists x_8 P\left(q_2, x_1, \ldots, x_{10}\right) = \\
& = q_2^1\left(\left(x_9^2 \vee x_9^3\left(x_{10}^2 \vee x_{10}^3\right)\right) \vee x_9^2 x_{10}^2 \vee x_{10}^2\left(x_9^2 \vee x_9^3\right)\right) \vee \\
& \vee q_2^2\left(\left(x_9^1 x_{10}^2 \vee x_9^2\right) \vee x_9^3\left(x_{10}^2 \vee x_{10}^1\right)\right) \vee \left(\left(x_9^1 \vee x_9^2\right) \vee \left(x_9^3 \vee x_9^1\right) x_{10}^2 \vee \right. \\
& \left. \vee x_9^1\left(x_{10}^2 \vee x_{10}^3\right) \vee \left(x_9^2 \vee x_9^3\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee \left(x_9^2 x_{10}^2 \vee x_9^3\right) \vee \\
& \vee x_9^2\left(x_{10}^1 \vee x_{10}^2\right) \vee q_2^3 (x_{10}^2\left(x_9^1 \vee x_9^2\right) \vee x_9^3\left(x_{10}^2 \vee x_{10}^3\right) \vee \left(x_9^2 \vee x_9^3\right) \vee \\
& \vee\left(\left(x_9^1 \vee x_9^3\right) \vee x_9^1 x_{10}^1 \vee \left(x_9^2 \vee x_9^1\right)\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee \\
& \vee\left(x_9^2 x_{10}^3 \vee x_9^3\right) \vee q_2^4\left(x_9^3 x_{10}^1 \vee x_9^1 x_{10}^1 \vee \left(x_9^1 \vee x_9^2\right)\left(x_{10}^1 \vee x_{10}^3\right) \vee x_9^1\right) = 1
\end{aligned}
$$

We reduced the initial formula and obtained a simpler dependence between the selected characteristics. Once the necessary dependence has been obtained, we can solve the resulting equation with one or more target variables.

Depending on the solution of the system of predicate equations, the medical record will be assigned to a class, and the doctor will develop a set of therapeutic and preventive procedures and recommendations to maintain the patient’s health in good condition.

Although some variables in the medical example can take on the value «unknown», we are dealing with a closed world, since «unknown» only means an element from the domain for the variable. All domains are strictly defined and cannot be supplemented by any other elements. The main advantage of this method, based on the specific structure of predicate systems, is that after removing non-essential variables, the original system (or equation) is simplified, which is related to the special properties of quantifiers. Variables that are essential are not necessarily fixed forever. The researcher decides which logical relationships are important at a given moment.

As a further direction of research, it is possible to extend the classes of predicates from which non-essential variables can be easily removed. Such classes are much more complex than relational structures, but many practical tasks require such a representation and analysis of knowledge.

Conclusions

1. It has been shown that finite predicate algebra (FPA) is a universal mathematical apparatus for the formal description of deterministic, discrete, finite objects of various natures. FPA can interpret knowledge in a strict mathematical form, where different features and their meanings are linked by Boolean and predicate operations.

2. It has been shown that when constructing a feature system for a knowledge base, it is advisable to use an algebraic-logical model of the subject area. An approach based on the use of the comparator identification method and FPA is proposed.

3. It is shown that in practical tasks related to the interpretation of knowledge, which is mostly presented in textual form, it is not necessary to obtain all sets of values of semantic features, but it is necessary to obtain one or several essential sets of feature values (target variables), which are a fixed set of values of other features. When solving such problems, other variables that are not included in the initial conditions and are not target variables are excluded from the equation by binding them with existential quantifiers.

4. An algorithm for excluding non-essential variables when solving algebraic-logical equations is proposed, which reduces the number of necessary calculations. At the same time, it is proposed to single out those predicates (and, accordingly, equations) which, when a certain value of a feature is substituted, turn into predicates that give a stronger connection between variables, as well as those predicates, the substitution of this value into which leads to a weakening of the logical connection between features.

5. Finite predicate equations of various types are considered. A description of classification problems based on predicate equations is given. The problem of classifying objects based on features that take discrete values is described mathematically as a solution of predicate equations.

6. A broad class of predicates is described from which redundant variables can be removed and focus can be placed on the relationships between essential variables. A method for removing non-essential variables by applying an existential quantifier is proposed and demonstrated on a real medical example. The example demonstrates the use of the mathematical apparatus of intelligence theory, the method of comparator identification, and the tools of finite predicate algebra on a model of identification of medical diagnostic parameters in the form of a system of predicate equations, the solution of which provides an interpretation of medical knowledge in a specific field.

References

1. Shabanov-Kushnarenko, Yu.P. Theory of Intelligence. Mathematical Means. Kharkiv: Higher School. Published by Kharkiv University, 1984. 144 pp.
2. Shabanov-Kushnarenko, Yu.P. Theory of Intelligence. Technical Means. Kharkiv: Higher School. Published by Kharkiv University, 1986. – 136 p.
3. Shabanov-Kushnarenko Yu.P. Theory of Intelligence. Problems and Prospects. –Kharkiv: Higher School. Published by Kharkiv University, 1987. – 159 p.
4. Bondarenko M.F., Shabanov-Kushnarenko Yu.P. Theory of Intelligence: Textbook. – Kharkiv: LLC «SMIT Company», – 2006. – 576 p.
5. Bondarenko M.F., Shabanov-Kushnarenko Yu.P. Brain-like Structures. Reference Book. Volume One. Edited by Academician I.V. Sergienko, National Academy of Sciences of Ukraine. – Kiev, Naukova Dumka, 2011. – 460 pp.
6. Bondarenko M.F., Shabanov-Kushnarenko Yu.P., Sharonova N.V. Tools for Comparative Identification. – Bionics of Intelligence. – 2010, No. 2(73), pp. 74–86.
7. Bondarenko M.F., Shabanov-Kushnarenko Yu.P., Sharonova N.V. Situational-textual predicate. – Bionics of Intelligence. – 2010, No. 2(73), pp. 87–98.
8. Bondarenko M.F., Shabanov-Kushnarenko Yu.P., Sharonova N.V. Boolean structure of text. – Bionics of Intelligence. – 2010, No. 2(73), pp. 99-110.
9. Bulkin V.I., Sharonova N.V. Mathematical models of knowledge and their implementation using algebraic predicate structures: monograph // Bulkin V.I., Sharonova N.V. – NTU «KhPI», MEGI. – Donetsk, 2010. – 304 p.
10. Kanishcheva O.V., Sharonova N.V. Linguistic technologies for knowledge identification in information systems. Linguistic technologies for knowledge extraction and processing by intelligent systems. – LAP LAMBERT Academic Publishing? – 2013. – 164 p.
11. Khayrova N.F., Sharonova N.V. Information and linguistic technologies for extraction and identification of deep knowledge in texts: monograph // N.F. Khayrova, N.V. Sharonova. – Kharkiv: FLP Koryak S.F., 2016. – 205 p.
12. Karatayev O.A., Shubin I.Yu. Problems of knowledge reuse in software system design / O. A. Karatayev, I.Yu. Shubin // Current state of scientific research and technologies in industry, 2023. No. 2(24). – pp. 25–34. DOI: https://doi.org/10.30837/ITSSI.2023.24.025
13. Karataev O.A., Sitnikov D.E. Method for reusing knowledge in the form of logical equations / O.A. Karataev, D.E. Sitnikov // Current state of scientific research and technologies in industry, 2023. No. 3(25). – pp. 25–34. DOI: https://doi.org/10.30837/ITSSI.2023.24.025
14. Karataiev O., Sitnikov D., Sharonova N. A Method for Investigating Links between Discrete Data Features in Knowledge Bases in the Form of Predicate Equations. CEUR Workshop Proceedings, Vol. 3387. 2023. pp. 224–235. URL: https://ceur-ws.org/Vol-3387/paper17.pdf
15. Melnik K. Towards medical screening information technology: the healthgrid-based approach/ K. Melnik, O. Cherednichenko, V. Glushko // Information Systems: Methods, Models, and Applications. – Heidelberg : Springer, 2013. – P. 202–204.
16. Olga Cherednichenko, Nataliia Sharonova, Ganna Pliekhova, Nadiia Babkova: Intelligent Methods of Secure Routing in Software-Defined Networks. Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems. Volume I: Machine Learning Workshop, Lviv, Ukraine, April 12–13, 2024. CEUR Workshop Proceedings 3664, CEUR-WS.org 2024 Pp.342–351.

17. Victoria Vysotska, Kirill Smelyakov, Natalia Sharonova, Maksym Derenskyi, Ganna Pliekhova and Vadim Repikhov. Information System for Monitoring and Planning Maintenance of Offshore Wind Farms - Conference on Computational Linguistics and Intelligent Systems. Volume II: Modeling, Optimization, and Controlling in Information and Technology Systems Workshop 2024, Lviv, Ukraine, April 12-13, 2024. CEUR Workshop Proceedings 3664, CEUR-WS.org 2024 Pp. 63–82.
18. Cui, X., Tang, D., Zhu, H., & Yin, L. (2017). Knowledge representation and semantic inference of process based on ontology and swrl. Machine Building & Automation, 46(3), 6–10
19. Moshood, T. D., Rotimi, F. E., Rotimi, J. O. B. 2022. An Integrated Paradigm for Managing Efficient Knowledge Transfer: Towards a More Comprehensive Philosophy of Transferring Knowledge in the Construction Industry. Construction Economics and Building, 22:3, 65–98. DOI: https://doi.org/10.5130/ AJCEB.v22i3.8050
20. Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, Z. Duan, Knowledge graph completion: A review, IEEE Access 8 (2020) 192435–192456
21. Schacht, S., Maedche, A. (2016). A Methodology for Systematic Project Knowledge Reuse. Innovations in Knowledge Management. Razmerita, L. (Eds.), Springer (Berlin), 2016, pp. 19–44.
22. Ameri, F., Dutta, D. (2005). Product lifecycle management: closing the knowledge loops. Computer-Aided Design and Applications 2 (5), pp. 577–590.
23. Milton, N. R. (2007). Knowledge acquisition in practice: a step-by-step guide. Springer Science & Business Media.
24. Carro Saavedra C., Serrano Villodres T., Lindemann U. (2016). Review and Classification of Knowledge in Engineering Design. IcoRD2017. Technische Universität München
25. Philippe Martin and Jérémy Bénard. 2016. Top-level Ideas about Importing, Translating and Exporting Knowledge via an Ontology of Representation Languages. In Proceedings of the 12th International Conference on Semantic Systems (SEMANTiCS 2016). Association for Computing Machinery, New York, NY, USA, 89–92. DOI: https://doi.org/10.1145/2993318.2993344
26. Khudhair, A. T. (2017). The intelligence theory mathematical apparatus formal base. Advanced Information Systems, 1(1), 38–43. DOI: https://doi.org/10.20998/2522-9052.2017.1.07
27. Sharonova, Natalia et al. «Issues of Fact-based Information Analysis». International Conference on Computational Linguistics and Intelligent Systems (2018).
28. P. G. Omran, K. Wang, Z. Wang, An Embedding-based Approach to Rule Learning in Knowledge Graphs, TKDE 33 (2021) 1348–1359.
29. 13 T. Pellissier-Tanon, G. Weikum, F. Suchanek, YAGO 4: A Reasonable Knowledge Base, in: ESWC, 2020
30. Omran, P. G., Wang, Z., & Wang, K. (2022). Learning Rules With Attributes and Relations in Knowledge Graphs. In AAAI Spring Symposium: MAKE.
31. P. G. Omran, K. Wang, Z. Wang, Scalable rule learning via learning representation, in: IJCAI, 2018.
32. M.Svato, S.Schockaert, J.Davis, STRiKE: Rule-Driven Relational Learning Using Stratified k-Entailment, in: ECAI, 2020.
33. G. A. Oparin, V. G. Bogdanova, and A. A. Pashinin, «Classification in a binary feature space using logical dynamic models», 44th International Congress on Information, Communication and Electronic Technologies (MIPRO), Opatija, Croatia, 2021, pp. 1020–1025, DOI: 10.23919/MIPRO52101.2021.9596697

34. A. Kabulov, E. Urunboev, and I. Saimanov, «Method for object recognition based on logical correction functions,» 2020 International Conference on Information Sciences and Communication Technologies (ICISCT), Tashkent, Uzbekistan, 2020, pp. 1–4, doi: 10.1109/ICISCT50599.2020.9351473. pp. 1–4, DOI: 10.1109/ICISCT50599.2020.9351473
35. M. Ahmed, S. K. Ibrahim, and S. Yakut, «Classification of hyperspectral images based on logical data analysis,» 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019, pp. 1–9, DOI: 10.1109/AERO.2019.8742023
36. S. Kim, S. Noh and H. S. Ryoo, «Identifying Combinatorial Significance for Classification of Alzheimer's Disease Proteomics Expression with Logical Analysis of Data», 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, Texas, USA, 2021, pp. 1661–1663, DOI: 10.1109/BIBM52615.2021.9669835
37. Povkhan and M. Lupe, «Algorithmic Classification Trees,» 2020 IEEE Third International Conference on Intelligent Data Analysis and Processing (DSMP), Lviv, Ukraine, 2020, pp. 37–43, DOI: 10.1109/DSMP47368.2020.9204198
38. M. Tamilselvi, G. Ramkumar, G. Anita, P. Nirmal, and S. Ramesh, «A New Text Recognition Scheme Using a Classification-Based Digital Image Processing Strategy,» 2022 International Conference on Advances in Computing, Communication, and Applied Informatics (ACCAI), Chennai, India, 2022, pp. 1–6, DOI: 10.1109/ACCAI53970.2022.9752542
39. S. N. Veigl and B. Kovalerchuk, «An interactive visual approach to self-service data classification for democratizing machine learning,» 2020 24th International Conference on Information Visualization (IV), Melbourne, Australia, 2020, с. 280–285, DOI: 10.1109/IV51561.2020.00052
40. Б. Jean-Marc and O. Lafitte, «Combining weak classifiers: a logical analysis,» 2021 23rd International Symposium on Symbolic and Numerical Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 2021, pp. 178–181, DOI: 10.1109/SYNASC54541.2021.00038
41. Z. Chen, A. Xu, Y. Zhou, and Y. Gai, «Research on Pulsar Classification Based on Machine Learning,» 2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, 2020, pp. 14–18, DOI: 10.1109/ICoIAS49312.2020.9081836
42. Nwankwo, H. Wimmer, L. Chen, and J. Kim, «Text Classification of Digital Forensic Data,» 2020 11th Annual IEEE Conference on Information Technology, Electronics, and Mobile Communications (IEMCON), Vancouver, British Columbia, Canada, 2020, pp. 0661–0667, DOI: 10.1109/IEMCON51383.2020.9284913
43. Shubin, S. Snisar, and S. Litvin, «Categorical Analysis of Logical Networks in Application to Intelligent Radar Systems,» 2020 IEEE International Conference on Information and Communication Technologies. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2020, pp. 235–238, DOI: 10.1109/PICST51311.2020.9467893
44. Cenzer, V. W. Marek and J. B. Remmel, «On the complexity of index sets for finite predicate logic programs that allow function symbols,» Journal of Logic and Computation, vol. 30, no. 1, pp. 107–156, Jan. 2020, DOI: 10.1093/logcom/exaa005
45. A. Kabulov, E. Urunbaev, and A. Ashurov, «Logical method for finding maximum common subsystems of Boolean equation systems,» 2020 International Conference on Information Sciences and Communication Technologies (ICISCT), Tashkent, Uzbekistan, 2020, pp. 1–5, DOI: 10.1109/ICISCT50599.2020.9351394

46. H. Qi, B. Li, R.-J. Jing, A. Proutiere and G. Shi, «Distributed Solving Boolean Equations over Networks,» 2020 59th IEEE Conference on Decision and Control (CDC), Jeju, Korea (South), 2020, pp. 560–565, DOI: 10.1109/CDC42340.2020.9304144
47. F. N. Castro, O. E. González and L. A. Medina, «Diophantine equations with binomial coefficients and perturbations of symmetric Boolean functions», in IEEE Transactions on Information Theory, vol. 64, no. 2, pp. 1347–1360, February 2018, DOI: 10.1109/TIT.2017.2750674
48. L. Gong, J. Zhang, L. Sang, H. Liu and Y. Wang, «The Unpredictability Analysis of Boolean Chaos,» in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 10, pp. 1854–1858, Oct. 2020, DOI: 10.1109/TCSII.2019.2949571
49. T. Kosovska, «Implementation of a partial sequence of formulas for coarse-grained solution of AI problems within a logic-predicate approach,» 2019 Computer Science and Information Technologies (CSIT), Yerevan, Armenia, 2019, pp. 65–68, DOI: 10.1109/CSITechnol.2019.8895153
50. Y.-F. Tseng and S.-J. Gao, «Efficient encryption of predicate subsets for the Internet of Things,» 2021 IEEE Conference on Dependable and Secure Computing (DSC), Aizu-Wakamatsu, Fukushima, Japan, 2021, pp. 1–2, DOI: 10.1109/DSC49826.2021.9346245
51. K. Smelyakov, A. Chuprina, O. Bogomolov, and N. Gunko, «Effectiveness of neural network models for face detection and recognition,» 2021 IEEE Open Conference on Electrical, Electronic, and Information Sciences (eStream), 2021, pp. 1–7, DOI: 10.1109/eStream53087.2021.9431476
52. Sharonova N. Issues of fact-based information analysis [Electronic resource] / N. Sharonova, A. Doroshenko, O. Cherednichenko // Computational Linguistics and Intelligent Systems (COLINS 2018): Vol. 1: Lviv, 2018. P. 11–19. URL: http://ceur-ws.org/Vol-2136/10000011.pdf
53. D.E. Sitnikov, B. D'Kruz, P.E. Sitnikova, «Identification of essential data features by constructing and manipulating logical equations,» Data Mining II, WIT Press, 2000, 241–248.
54. Melnik K. Towards information technology for medical screening: an approach based on healthgrid / K. Melnik, O. Cherednichenko, V. Glushko // Information Systems: Methods, Models and Applications. – Heidelberg: Springer, 2013. P. 202–204.