Problems of Solvability and Choice of Algorithms for Decision Making by Precedence

V. V. Krasnoproshin and V. A. Obraztsov

Belarussian State University, Minsk, Belarus

e-mail: krasnoproshin@bsu.by

DOI: 10.1134/S1054661806020027

INTRODUCTION

A wide range of practical and theoretical problems are reduced to decision making, which is interpreted rather ambiguously from the viewpoint of its statement. This is mainly related to the fact that the concept of *decision* cannot be formalized. From the methodological point of view, decision making is a process whose result is a decision. Even if the concept of *decision* is not defined, such a methodological view of the problem highlights its key points in a different way. In this case, the problem is transformed into a sequence of clearer and unambiguously defined problems. In this sequence, there certainly exists the problem of calculating the property of objects, and the decision itself can be associated with such a property [21]. Then, the problem of decision making can be reduced to calculating (determining) the properties of the information analyzed. This is undoubtedly one version of the problem of decision making, which is based on the problem of determining the above properties. Decision making allows one to easily find out if all property determination problems have a common nature and to reduce them to a single statement on a formal level. What problems are meant by the determination of properties? These are well-studied problems of sentential calculus and predicate calculus [1, 2]; problems of logical diagnostics [3, 4], which are less studied but have a larger number of practical interpretations; and, finally, just poorly formalized problems in pattern recognition theory (PRT) [5, 9]. It is obvious that, formally, all the above problems contain sets divided into subsets. Each subset is characterized by a certain property. The problem consists in calculating this property for each element of the original set.

Why is it necessary to unify all the above problems? First, because all of them have essentially similar statements, and, second, because the solution of poorly formalized problems (such as logical diagnostics and problems of PRT) requires a certain methodology. The point is that the concepts of a *solvable problem* and a *validated algorithm* are naturally introduced into problems of sentential calculus (and certain other problems). For PRT problems, similar concepts are either impossible in the framework of the theory alone or require additional information. However, it is also required that one can demonstrate the solvability of a particular problem or the validity of the choice of algorithms for PRT problems, because many problems important in practice are reduced to PRT problems [9, 18]. This unification is in fact the goal of the present study. It provides a means for considering and analyzing PRT problems. Among a variety of PRT problems, we focus on the algorithmic aspects of the *problem of learning recognition* [5, 9]. In the context of the aforesaid, this problem is related to the class of problems of decision making by precedence. Note once again that this term only reflects our attempt to unify problems of the same type, while the terminology may be different.

The main results of the study are as follows:

(i) the concept of a solvable problem is introduced and solvability conditions are obtained for the case when information is given by precedence;

(ii) a class of algorithms satisfying the necessary conditions of solvability is studied in detail, and the results obtained are shown to be unimprovable;

(iii) a class of inductive inference algorithms is introduced and studied; this class can be applied to solving the whole class of decision making problems and is comparable with the known algorithms of deductive inference.

The results given in this work are closely related to the results of the studies [6, 7], which contain the proof of the main propositions and theorems.

1. FORMULATION OF THE PROBLEM AND GENERAL CONDITIONS OF ITS SOLVABILITY

1.1. Formal Statement of the Problem

Consider the following problem:

Given a certain, possibly infinite, number of subsets (classes) $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_l$ *on a set of objects* $\boldsymbol{X}$ *of arbitrary nature, find an algorithm* A *(possibly, the best, in a certain sense) that is defined on the whole set* $\boldsymbol{X}$ *and whose operation for each* $\boldsymbol{x} \in \boldsymbol{X}$ *can be interpreted in terms of membership of the latter in the subsets* $\boldsymbol{X}_i$.

Received January 26, 2006

Let us agree that we will consider this problem on the formal, rather than meaningful, level. Therefore, an arbitrary nature of objects of the set $\boldsymbol{X}$ means that some coding function has already been chosen and the corresponding spaces have been fixed. The space in which the results of the algorithms are formed is also fixed. It is clear that this leads to the problem of choosing the functions of coding and interpretation. This problem is closely related to the solvability of the above-formulated problem. The character of this problem and the methodology for solving it are described in [6].

Denote the problem formulated by $\mathbf{Z}$. According to the statement, this problem (or, more precisely, the class of problems) has the maximal level of generality. It is easily seen that a large number of problems are reduced to this statement. They differ in the way in which information on the set $\boldsymbol{X}$ and the subsets $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_l$ is defined.

It can be seen even in such a general formulation that one must answer a number of questions in order to solve problem $\mathbf{Z}$. The main question is that concerning the solvability of problem $\mathbf{Z}$ and the possibility of validating this solution. Without a satisfactory answer to this question, the solution of any applied problems that are reduced to some variants of problem $\mathbf{Z}$ will always be inadequate. Unfortunately, in the general case, the possibility of validation is doubtful, even if it is a matter of mathematical formalism [2]. However, there exist problems for which this possibility has been proved. These include, for example, the problem of classification of formulas in sentential calculus that employs the resolution method as the algorithm A, and a large number of property recognition problems, which are technically related to the former problem [8].

Regarding the solvability of problem $\mathbf{Z}$, the possibility of its validation is one of a number of sufficient conditions. The necessary conditions are mainly defined by the expressive power of the formalization language and the formulation of a specific problem. In order to describe such conditions, it is necessary to classify the problems corresponding to this statement.

Classification can be performed, in particular, according to the following criteria:

(i) whether the number of classes l and the set $\boldsymbol{X}$ are finite or infinite;

(ii) whether the classes $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_l$ are *fully described*, and whether they *cover* the whole set $\boldsymbol{X}$;

(iii) what method is chosen for describing the *objects* and a *membership function*;

(iv) what is known about the objects that form the classes $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_l$.

These criteria are of information character, which is in agreement with the logic of formulation of any problem; according to this logic, algorithms should result from the formalization and be completely determined by the character and amount of known information.

Let us consider some examples of problems.

Problem $\mathbf{Z}_A$ (classification of formulas in sentential calculus [2]).

The number of classes is finite ($l = 3$), *and the set* $\boldsymbol{X}$ *is infinite. The classes are described by axioms and inference rules without quantifiers. The classes* $\boldsymbol{X}_i$ *are fully described and cover the whole set* $\boldsymbol{X}$. *Finally, the membership of axioms is defined explicitly.*

Such a choice problem is solved by the resolution method [2]. The solution algorithm is denoted by A_0.

Problem $\mathbf{Z}_B$ (logical diagnostics [3, 4]).

The number of classes l is finite, and the set $\boldsymbol{X}$ *can be either finite or infinite. The classes* $\boldsymbol{X}_i$ *are not fully described and/or do not cover the whole set* $\boldsymbol{X}$. *The classes are described by predicates* (*rules*), *which are simultaneously used to describe the objects and the membership function. Information about the membership of the objects that satisfy the rules is assumed to be specified.*

Depending on the formalization language, one uses one version of the algorithm A_0 (a parameterized version) for solving this problem.

Problem $\mathbf{Z}_C$ (learning pattern recognition [5]).

The number of classes l is finite, and the set $\boldsymbol{X}$ *may be either finite or infinite. The classes* $\boldsymbol{X}_i$ *are not fully described and/or do not cover the whole set* $\boldsymbol{X}$. *The classes are described by precedence* (*i.e., by explicitly indicating objects that belong to each of the classes* $\boldsymbol{X}_1, \ldots \boldsymbol{X}_l$).

There exists a wide variety of algorithms for this problem. The construction of these algorithms depends on the formalization language, on the information available, and on certain additional assumptions.

From the viewpoint of statement, problems $\mathbf{Z}_{A-C}$ undoubtedly have a common nature: all of them are versions of the above-formulated problem $\mathbf{Z}$. However, from the technical (algorithmic) point of view, they are different; the same can be said of the problem of solvability. In problem $\mathbf{Z}_A$, the solvability is equivalent to the validity of the algorithm. In problem $\mathbf{Z}_B$, the scheme of the algorithm is similar. However, due to the parameterization, which is necessary for the algorithm to work beyond the classes $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_l$ or with objects with a priori unknown membership, only the scheme can be considered validated. Even in the case when information is reduced to the conditions of $\mathbf{Z}_A$, the algorithm itself is reduced to A_0 only in special cases. Finally, in the problem $\mathbf{Z}_C$, the corresponding algorithm has nothing to do with A_0. Therefore, it is hardly possible to say anything about the solvability of the problems $\mathbf{Z}_B$ and $\mathbf{Z}_C$.

Let us introduce the notion of a solvable problem $\mathbf{Z}$. For this purpose, we will need some additional notations and definitions.

The formulation of the problem suggests that a certain mechanism is required for distinguishing between objects of different classes. Introduce a system of predicates $P = (P_1, \ldots, P_l)$ and define it as follows:

$$\forall x \in \boldsymbol{X} \;\; (P_i(x) \in \{0, 1\} \wedge (P_i(x) = 1 \Rightarrow x \in \boldsymbol{X}_i)).$$

Now, if we denote by $I(\boldsymbol{X})$ some information about the set $\boldsymbol{X}$ in problem $\mathbf{Z}$, then any algorithm A that solves the problems $\mathbf{Z}_{A-C}$ can be represented as

$$\forall x \in \boldsymbol{X} \;\; (\mathrm{A}\colon x \times I(\boldsymbol{X}) \longrightarrow (P_1^{\mathrm{A}}(x), \ldots, P_l^{\mathrm{A}}(x))), \quad (1)$$

where $P^{\mathrm{A}} = (P_1^{\mathrm{A}}, \ldots, P_l^{\mathrm{A}})$ can be called an *algorithmic* predicate. Unlike P, this predicate is calculated by the algorithm A, and its values are chosen in the interval [0, 1].

It is clear that one can define an order on the set of all possible algorithms of the form (1). To formalize the requirements on the algorithms A that solve the problem $\mathbf{Z}$, we define l subsets,

$$\boldsymbol{X}_i^{\mathrm{A}} = \{x \in \boldsymbol{X}\colon P_i^{\mathrm{A}}(x) = 1\}.$$

Obviously, $\boldsymbol{X}_i^{\mathrm{A}}$ may differ from $\boldsymbol{X}_i$. To characterize this difference, we introduce monotonic functions

$$\mu\colon \boldsymbol{X}_i \times \boldsymbol{X}_i^{\mathrm{A}} \longrightarrow [0, 1] \;\text{ and }\; \psi\colon [0, 1]^l \longrightarrow [0, 1]$$

and require that these functions satisfy the conditions

$$\mu(\boldsymbol{X}_i, \boldsymbol{X}_i^{\mathrm{A}}) = \begin{cases} 1, \text{ if } \boldsymbol{X}_i = \boldsymbol{X}_i^{\mathrm{A}}, \\ 0, \text{ if } \boldsymbol{X}_i \cap \boldsymbol{X}_i^{\mathrm{A}} = \varnothing, \end{cases}$$

$$\psi(a_1, \ldots, a_l) = \begin{cases} 1, \text{ if } a_1 = \ldots = a_l = 1, \\ 0, \text{ if } a_1 = \ldots = a_l = 0. \end{cases}$$

Then, $\Phi_{\mathrm{A}}(\boldsymbol{X}) := \psi(\mu(\boldsymbol{X}_1, \boldsymbol{X}_1^{\mathrm{A}}), \ldots, \mu(\boldsymbol{X}_l, \boldsymbol{X}_l^{\mathrm{A}}))$ can be called a *quality functional* for the algorithm A since its values are easily interpreted in terms of a mismatch between P and P^{A}.

The problem $\mathbf{Z}$ for which $\Phi_{\mathrm{A}}(\boldsymbol{X}) = 1$ is said to be *solvable*. The corresponding algorithm for which the condition of solvability is satisfied, is naturally called a *validated* algorithm.

Among the problems $\mathbf{Z}_{A-C}$, only the problem $\mathbf{Z}_A$ is solvable. In particular, it is solvable by the algorithm A_0, which is validated in the context of the concepts introduced. The same algorithm A_0, when applied to solving a certain problem $\mathbf{Z}_B$, obviously becomes invalidated. Therefore, the notions of solvability and validity are closely related.

In the general case, it can be said that the goal of solving any problem from the class $\mathbf{Z}$ is the proof of its solvability, where validated algorithms can be used as a means.

1.2. Solvability of the Problem $\mathbf{Z}_C$

As mentioned in the Introduction, the main objective of this work is to investigate the problems $\mathbf{Z}_C$ for their validation and algorithmization. For this purpose, one should specify the formulation of the problem.

In order to exclude an appeal to the meaningful aspects of the problem (e.g., the *set of objects* $\boldsymbol{X}$ *of an arbitrary nature* in the formulation of the problem $\mathbf{Z}$), we will consider the problem statement only at a formal level. We will denote the set of objects by X and the classes by $X_1, \ldots, X_l$. It is clear that this change in the notation does not influence the formulation of the problem and the definitions introduced in Section 1.1. Next, we specify in more detail what is meant by explicit indication of objects that belong to each of the classes $X_1, \ldots, X_i$ in the formulation of the problem $\mathbf{Z}_C$. Usually, such a set of objects implies a finite sample of objects $x \in X$ that provides information $I(X)$ about X. Denote this sample by X^0. There are various terms for such samples in the literature [5, 9, 20]; most often, they are called *learning* samples, since they are used for constructing algorithms of the form

$$\forall x \in X \; (\mathrm{A}\colon x \times X^0 \longrightarrow (P_l^{\mathrm{A}}(x), \ldots, P_l^{\mathrm{A}}(x))). \quad (2)$$

It is easily seen that, in the general case, the functional Φ_{A} cannot be calculated on the whole set X for the problem $\mathbf{Z}_C$, since $I(X) = X^0$ according to the formulation of the problem. At the same time, a value of this functional is easily calculated on the learning sample X^0 (by definition). However, this is not sufficient, since the solvability is related to the condition

$$\Phi_{\mathrm{A}}(X) = 1. \quad (3)$$

In other words, algorithm A of the form (2) in the problem $\mathbf{Z}_C$ should satisfy condition (3), or should be the closest to it in a certain sense, if we are interested in the solvability of a particular problem $\mathbf{Z}_C$. This brings us to the following question: Is it possible to solve problem $\mathbf{Z}_C$ and simultaneously require that algorithm (2) should satisfy, say, condition (3)?

It is easily seen that the answer to this question consists in determining the relation between the values of the functionals $\Phi_{\mathrm{A}}(X)$ and $\Phi_{\mathrm{A}}(X^0)$. This relation is obvious:

$$\forall \mathrm{A}(\Phi_{\mathrm{A}}(X) < 1 \Leftrightarrow \exists X^0 \Phi_{\mathrm{A}}(X^0) < 1). \quad (4)$$

Relation (4) can be considered an axiom (which can naturally be called a *reduction axiom*).

This axiom implies the conditions for the quality functional Φ_{A} under which the axiom is consistent and can be used to determine the corresponding conditions for algorithms (2). To find these conditions, we formulate the following proposition.

Proposition 1. *Relation* (4) *is valid if and only if the functional* Φ_A *satisfies the conditions*

$$\forall X^0 \subseteq X \forall \tilde{X}^0 \subset X^0 (\Phi_A(X^0) = 1 \Rightarrow \Phi_A(\tilde{X}^0) = 1), \quad (5)$$

$$\forall X^0, \tilde{X}^0 \subseteq X(\Phi_A(X^0) = 1 \wedge \Phi_A(\tilde{X}^0) = 1 \Rightarrow \Phi_A(X^0 \cup \tilde{X}^0) = 1) \quad (6)$$

irrespective of the choice of the algorithm Φ_A. *Henceforth, we will assume without additional reservations that* Φ_A *satisfies the conditions of Proposition* 1.

Actually, conditions (5) and (6) imposed on the functional Φ_A imply that this functional is invariant with respect to some operations on the set X. In the case of condition (5), these are the operations of intersection of subsets, and, in the case of condition (6), the operations of union of subsets. These operations define conditions on the set of quality functionals under which the question of solvability can be raised.

Now, we apply equivalent transformations to transform the reduction axiom to the disjunctive representation

$$\forall A((\Phi_A(X) = 1 \wedge \forall X^0 \Phi_A(X^0) = 1) \vee (\Phi_A(X) < 1 \wedge \exists X^0 \Phi_A(X^0) < 1)). \quad (7)$$

This representation implies the dual form of the axiom,

$$\forall A(\Phi_A(X) = 1 \Leftrightarrow \forall X^0 \Phi_A(X^0) = 1). \quad (8)$$

The disjunctive representation shows that, in solving problem $\mathbf{Z}_C$, there are only two possibilities: either one or the other conjuncts in (7) takes place. Each of the algorithms that solve the above-formulated problem satisfies one and only one of the two conditions: either (3) or, say,

$$\sup_{\{A\}} \Phi_A(X). \quad (9)$$

In this case, the problem of constructing algorithm (3) cannot be considered as a limiting case of problem (9). Moreover, the similarity between the algorithms can be defined in a different way.

To solve each of these problems, we have the finite sample X^0 at our disposal, which allows us to find constraints on the choice of the corresponding algorithm A. In most cases, we use the so-called limit schemes in solving (9); these schemes are based on the proof (more frequently, on the assumption) that the following relation holds:

$$\Phi_A(X) = \lim_{X^0 \to X} \Phi_A(X^0).$$

Then, problem (9) is replaced by the problem of finding an algorithm that is extremal on sample X^0, and the algorithm obtained is taken as a solution to (9). The applicability of this method is due to the fact that the difference between the constructed and the required algorithms is at least checked and can be made arbitrarily small. Moreover, such a limit dependence between the values of the functionals $\Phi_A(X)$ and $\Phi_A(X^0)$ provides typical sufficient conditions for the consistency of the extended reduction axiom. Note that the application of limit schemes has reached a certain level of methodological completeness in the framework of the statistical approach [5, 10] and pattern theory [11] (it is also statistical in its principles and methods). This is quite obvious because mathematical statistics is based on frequency concepts, which contain an apparatus for investigating various limit schemes.

Now, let us return to problem (3). Unlike (9), its solution can be obtained by using the reduction axiom in the form (8). Consider certain sufficient conditions for the consistency of the latter axiom. For this purpose, we need the following definition:

A sample $\tilde{X}^0 \subseteq X$ is said to be *representative* with respect to the algorithm A for $\mathbf{Z}_C$ if

$$\forall x_1 \notin \tilde{X}^0 \exists x_2 \in \tilde{X}^0((P(x_1) = P(x_2)) \Leftrightarrow (P^A(x_1) = P^A(x_2))).$$

One can easily prove the following proposition.

Proposition 2. *Let* $\tilde{X}^0 \subseteq X$ *be an arbitrary finite sample in X that is representative with respect to a certain algorithm* A. *Then,*

$$\Phi_A(\tilde{X}^0) = 1 \Rightarrow \Phi_A(X) = 1.$$

Such a sufficient condition is not unique. Let us consider another condition, which is as obvious as the previous one.

The algorithm A is said to be *competent* as applied to the sample $X^0 \subseteq X$ if

$$\forall x \notin X^0 (P_1^A(x) = \ldots = P_l^A(x) = 0).$$

One can prove the following proposition for such algorithms.

Proposition 3. *Suppose that* $X_1^0, X_2^0, \ldots$ *is a certain, possibly infinite, sequence of samples from* $\mathbf{X}$ *such that*

$$\forall i, j\ (i \neq j \Rightarrow X_i^0 \cap X_j^0 = \varnothing), \quad \bigcup_i X_i^0 = X,$$

and $A_1, A_2, \ldots$ *is the corresponding sequence of competent algorithms. Then, the algorithm* $A := (\sum_i P_1^{A_i}, \ldots, \sum_i P_l^{A_i})$ *satisfies the following relation:*

$$\forall X_i^0 (\Phi_{A_i}(X_i^0) = 1) \Rightarrow \Phi_A(X) = 1.$$

The following proposition is a direct generalization of, and simultaneously a corollary to, Propositions 2 and 3.

Proposition 4. *Suppose that* $X_1^0, X_2^0, \ldots$ *is a certain, possibly infinite, sequence of samples from X such that*

$$\forall i, j\ (i \neq j \Rightarrow X_i^0 \cap X_j^0 = \varnothing), \quad \bigcup_i X_i^0 = X,$$

and $X_1^0, X_2^0, \ldots$ *are samples such that* $\forall i(\tilde{X}_i^0 \subseteq X_i^0)$ *and each* $\tilde{X}_i^0$ *is representative with respect to an algorithm* A_i *for* X_i^0. *Then, if the algorithms* $A_1, \mathrm{A}_2, \ldots$ *are competent as appled to the samples* $X_1^0, X_2^0, \ldots$, *respectively, then the algorithm* $\mathrm{A} := (\sum_i P_1^{\mathrm{A}_i}, \ldots, \sum_i P_l^{\mathrm{A}_i})$ *satisfies the relation*

$$\forall \tilde{X}_i^0(\Phi_{\mathrm{A}_i}(\tilde{X}_i^0) = 1) \Rightarrow \Phi_{\mathrm{A}}(X) = 1.$$

Now, let us discuss the results obtained. Apparently, there exist other sufficient conditions for the solvability of problem (3). In our case, the following condition for the algorithm A is common to all the results: $\Phi_{\mathrm{A}}(\tilde{X}^0) = 1$ (or some of its variants). Such algorithms have been used in PRT for quite a long time [9, 19, 20]. They are called *correct* algorithms. Thus, the property of correctness on a given sample turns out to be one of the sufficient conditions. This condition can serve as a basis for choosing the algorithm for solving problem (3). Relation (8) also implies that the correctness of the algorithm irrespective of the sample represents a necessary condition. Therefore, whatever sufficient conditions are obtained in the future, they will inevitably be connected with this property.

The sufficient conditions obtained above can also be considered from the viewpoint of requirements on information. In this sense, the representative character of the sample is a "stronger" requirement than the existence of a partition of the set X. Probably, this is why the correctness of the algorithm is supplemented with the condition of competence in the latter case; i.e., the requirements imposed on the algorithms become stronger at the expense of weakening the requirements on information; however, the total complexity of the problem seems to remain the same.

Now, let us examine the problem of constructing representative samples. Note that this is the most complicated problem not only of PRT but also of inductive inference on the whole [8, 12–16]. This is the reason why problem (3) preserves its inductive nature even in spite of the fact that the problem of choosing the algorithms for a fixed sample is solved by deductive means. The problem under discussion can be solved by the characterization of such samples. For example, the following property can readily be proved.

Property. *A sample* $\tilde{X}^0 \subseteq X$ *is representative with respect to an algorithm* A *for X only if it has the property*

$$\forall i \in \{1, \ldots, l\}(\tilde{X}^0 \cap X_i \neq \varnothing).$$

This property implies that a sample can be representative only under the condition that it includes precedents from all classes. Some other properties of such samples can be obtained just as easily. However, the problem cannot be solved completely until the inductive inference is proved in a certain sense [12].

1.3. Some Drawbacks of Formalization

The notions introduced above allow one to formalize the solvability conditions for problem $\mathbf{Z}_C$ and to understand the validation of algorithms (2). However, as often occurs in mathematics, "excessive" formalization aimed at an accurate representation inevitably leads to ambiguity.

Let us analyze the formal language introduced above. For this purpose, we restrict ourselves to the case $l = 2$ for the present. To each algorithm of the form (2), we assign an algorithm A',

$$\forall x \in X\ (\mathrm{A}'\colon x \times X^0 \longrightarrow (P_1^{\mathrm{A}'}(x), P_2^{\mathrm{A}'}(x))),$$

and determine its results as follows:

$$P_i^{\mathrm{A}'}(x) = 1 - P_i^{\mathrm{A}}(x).$$

It is easily seen that the algorithms A and A' are inverse. Now, consider the relation between the corresponding values of the functionals $\Phi_{\mathrm{A}}(X)$ and $\Phi_{\mathrm{A}'}(X)$. Obviously, irrespective of the set X and the conditions imposed on the functional Φ (except conditions (5) and (6)), the relation has the form

$$\Phi_{\mathrm{A}}(X) = 1 - \Phi_{\mathrm{A}'}(X), \text{ for } \Phi_{\mathrm{A}}(X) \in \{0, 1\}.$$

If Φ is additive (for example, is such as the *percentage of correct predictions* [9]), the above relation holds even under the condition $\Phi_{\mathrm{A}}(X) \in [0, 1]$. This suggests that problems (3) and (9) can be solved in the inverse formulation as well, and the result concerning the solvability of the problem and the validity of the algorithms will be the same.

Now, let A be an arbitrary algorithm of the form (2). Divide the set X into disjoint subsets X_1 and X_2 such that $X_1 \cup X_2 = X$. Assign the algorithm A to the algorithms A_1 and A_2 as follows:

$$\forall x \in X_i(\mathrm{A}_i(x) = \mathrm{A}(x)) \wedge \forall x \in X \backslash X_i(\mathrm{A}_i(x) = (0, 0)),$$
$$i = 1, 2.$$

It is easily seen that A_1 and A_2 represent "projections" of the algorithm A onto the corresponding subsets X_1 and X_2 that are defined over the whole set X. Obviously, under these conditions, the algorithm A can be represented in the form $A = A_1 + A_2$. Under the additional condition that the functional Φ is additive, the following relation is valid: $\Phi_A(X) = \Phi_{A_1}(X) + \Phi_{A_2}(X)$.

Let us return again to the problem $\mathbf{Z}_C$. Suppose that we can calculate the value of the functional $\Phi_A(X) \in [0, 1]$ under the conditions of problem $\mathbf{Z}_C$. Suppose also that this means that one can point out samples X_1 and X_2, where X_1 is a sample on which algorithm A satisfies condition $P(X_1) = P_A(X_1)$ and X_2 is set equal to $X \backslash X_1$. With such a partition, one can easily construct an algorithm A' such that $\Phi_{A'}(X) = 1$ irrespective of the result of $\Phi_A(X)$.

If samples X_1 and X_2 cannot be indicated, we can still construct a procedure (possibly, a stochastic procedure) that will result in an appropriate algorithm A'. Note, however, that in this case it is impossible to dispense with additional assumptions on the structure of information X and/or on the relation between the sets X and X_0.

We do not know whether all the above considerations remain valid when, instead of calculating the value of $\Phi_A(X) \in [0, 1]$, we can only estimate it, like in the case when limit schemes are used in problem (9). Even if all the above considerations are not valid in problem (9), they nevertheless make us less optimistic about the possibility of calculating and/or estimating $\Phi_A(X) \in [0, 1]$.

It remains to note that all the above calculations are generalized to the case of $l > 2$. The existence of the inverse algorithm A' is easily proved. However, its construction is associated with certain algorithmic difficulties. The simplest way to construct it is to use a dichotomic partition of set X^A, which represents a union of the subsets $X_1^A, \ldots, X_l^A$.

2. REALIZATION OF CORRECT ALGORITHMS

The results obtained above upon analyzing the solvability of problem $\mathbf{Z}_C$ are based on the condition of correctness of algorithm A: $\Phi_A(\tilde{X}^0) = 1$. It is also clear that the validated algorithms are also contained in the set of correct algorithms. Therefore, in a formal approach to solving problem $\mathbf{Z}_C$, one should construct the whole set of correct algorithms for each such problem. It is desirable that the complexity of such a construction should be minimal.

As pointed out in the Introduction, the majority of the results presented in this section are based on or related to the results of [6, 7, 17]. Therefore, the description is largely schematic and simplified. Many details used in further constructions and notations can be found in the above-cited works.

2.1. General Correctness Conditions of Algorithms

First, let us describe the set of recognition algorithms that will be analyzed below. For this purpose, we return to the assumptions made in the formulation of the problem. These assumptions imply that, in the most general form, a recognition algorithm can be considered as a map A: $X \longrightarrow \mathbf{B}_2^l$, where $\mathbf{B}_2^l$ is the lth Cartesian degree of the set $\mathbf{B}_2 = \{0, 1\}$. In this case, the result of this algorithm on all $x \in X$ is easily interpreted in the system of predicates P^A. In the set of such maps, we choose a subset (class) of algorithms generated by a natural superposition of the form $A = \mathbf{c} \circ B$, which consists of a *recognition operator* B: $X \longrightarrow \mathbf{R}^l$ (here, $\mathbf{R}$ is the space of real numbers) and a *decision rule* $\mathbf{c}$: $\mathbf{R}^l \longrightarrow \mathbf{B}_2^l$. The set of recognition operators that are of the same type in a certain sense are called a model and denoted by $\mathfrak{M}$. Then, $A_{\mathfrak{M}} = \mathbf{c} \circ \mathfrak{M}$ represents a certain model of recognition algorithms. Among all $A_{\mathfrak{M}}$ admitting such a representation, we restrict the analysis to models with surjective maps c whose decision rules satisfy the condition $\forall b \in \mathbf{B}_2^l \; \exists r \in \mathbf{R}^l \; (\mathbf{c}(r) = b)$. This is associated with the fact that the original formulation of the problem does not contain any constraints on the structure of classes nor on the relationship between them. Therefore, it is easily seen that (3) necessarily implies the restriction imposed on the model $A_{\mathfrak{M}}$.

Now, consider the conditions for the existence of algorithms in the model $A_{\mathfrak{M}}$ that are correct on a fixed finite sample X^0 (or, in other words, consider the correctness conditions for the model $A_{\mathfrak{M}}$). For this purpose, we associate the decision rule $\mathbf{c}$ on $\mathbf{X}$ with the following set: $R_c(X) = \{R \subseteq R^l | \mathbf{c}(R) = P(X)\}$, where $P(X) \subseteq \mathbf{B}_2^l$ is a system of all possible values of the predicate P on X. The surjectivity of $\mathbf{c}$ implies that this definition is justified. However, for the reasons indicated above, it is impossible to calculate the set $R_c(X)$. The situation is different for the set X^0, although there are some difficulties in this case as well since $R_c(X^0)$ may be infinite. Anyway, we structure the latter set by fixing a certain order in $\{1, \ldots, l\}$ and $\{1, \ldots, |X^0|\}$. Now, if we introduce the notation $|X^0| = q$, then $R_c(X^0)$ can be considered as matrices in the space $\mathbf{R}^{ql}$ of dimension $q \times l$. Obviously, $B(X^0)$ are elements of the same matrix space for all recognition operators $B \in \mathfrak{M}$. Then, the following proposition is easily proved.

Proposition 5. *The model* $A_{\mathfrak{M}}$ *is correct on the sample* X^0 *if and only if*

$$\mathfrak{M}(X^0) \cap R_c(X^0) \neq \varnothing. \tag{10}$$

Indeed, under the assumptions made, it is easily verified that condition (10) is equivalent to

$$\exists A \in A_{\mathfrak{M}}(\Phi_A(X^0) = 1).$$

This condition allows us to further specify the set of models $A_{\mathfrak{M}}$, since it necessarily implies the surjectivity of decision rule c, although on the set corresponding to the sample X^0 (such c are sometimes called *correct* or *consistent*). This means that, in constructing and investigating $A_{\mathfrak{M}}$, we can restrict ourselves to a certain fixed surjective decision rule, because $R_c(X^0) \neq \varnothing$ in this case. Of course, this does not imply that the whole set of possible models $A_{\mathfrak{M}}$ can be generated in this way. The correct conclusion is that all complications of the realization are shifted to the modeling of recognition operators $\mathfrak{M}$. Note also the following important circumstance: the investigation of $A_{\mathfrak{M}}$ is considerably simplified since the space $\mathbf{R}^{\mathbf{q}l}$ is much "stronger" from the viewpoint of its possibilities than $\mathbf{B}_2^{\mathbf{q}l}$ (in which only logical operations are admissible) and is certainly not "weaker" than any set X. On the basis of the aforesaid, we will henceforth consider the models $A_{\mathfrak{M}}$ with the so-called *linear threshold* decision rules that are defined in the space $\mathbf{R}^{\mathbf{q}l}$ (along with the space $\mathbf{R}^l$),

$$\forall R \in \mathbf{R}^{\mathbf{q}l} \;\; (\mathbf{c}(R) = \|\mathbf{c}(r_{ij})\| \in \mathbf{B}_2^{\mathbf{q}l}),$$

and act according to the rule (where $(i, j) \in I = \{1, \dots, q\} \times \{1, \dots, l\}$, $c_0 \in \mathbf{R}$)

$$c(r_{ij}) = \begin{cases} 1, & \text{if} \;\; r_{ij} > c_0, \\ 0, & \text{if} \;\; r_{ij} \le c_0. \end{cases}$$

Thus, we will focus on the correctness conditions of the models $A_{\mathfrak{M}}$ generated by model $\mathfrak{M}$ and a certain fixed linear threshold decision rule $c(c_0)$. Let us choose the threshold from the condition $c_0 > 0$, which, as will be seen below, does not lead to any loss in generality of the results. Assume, for simplicity, that all models $\mathfrak{M}$ examined below satisfy the following condition:

$$\forall B \in \mathfrak{M} \;\; \forall x \in X$$
$$(B(x) = (b_1(x), \dots, b_l(x)), b_i(x) \ge 0).$$

First, consider the most general case, when no additional conditions are imposed on model $\mathfrak{M}$. It is easily seen that $R_c(X^0)$ forms a convex subset in the space $\mathbf{R}^{\mathbf{q}l}$ and, due to the choice of the decision rule, is a solution to a system of nonstrict linear inequalities. The set of such solutions can be characterized in terms of the separability of specially constructed subspaces. We need some notations to describe these subspaces. Divide the set of indices $\mathbf{I}$ into the subsets

$$M_t = \{(i, j) | (i, j) \in \mathbf{I}, P_j(x_i) = t\}$$

and, by analogy with the vector case, introduce a scalar product of matrices in the space $\mathbf{R}^{\mathbf{q}l}$:

$$\forall R_1, R_2 \in \mathbf{R}^{\mathbf{q}l} \left(\langle R_1, R_2 \rangle := \sum_{(i, j) \in \mathbf{I}} r_{ij}^1 r_{ij}^2 \right),$$
$$\text{where} \;\; R_k = \|r_{ij}^k\|, \quad k = 1, 2.$$

Denote by $\{E_{ij}\}$ a canonical basis (with the unit element $(i, j) \in \mathbf{I}$ and other elements zero) of the space $\mathbf{R}^{\mathbf{q}l}$. Using these notations, one can easily formulate and prove the following correctness criterion for the model $A_{\mathfrak{M}}$.

Theorem 1. *The model* $A_{\mathfrak{M}}$ *is correct on* $\mathbf{X}^0$ *if and only if*

$$\begin{aligned} &\exists B \in \mathfrak{M}(\min_{(i, j) \in M_1} \{ \langle B(X^0), E_{ij} \rangle \} \\ &> c_0 \ge \max_{(i, j) \in M_0} \{ \langle B(X^0), E_{ij} \rangle \}). \end{aligned} \tag{11}$$

The proof of this theorem is straightforward and is implied by the equivalence of (10) and (11).

One can also obtain other correctness conditions, including sufficient conditions for correctness. In practice, however, such conditions prove inefficient, since it is possible to obtain appropriate algorithms only by solving systems of matrix inequalities. This is a rather complicated task, even for the simplest models $\mathfrak{M}$. Such a nonconstructive property of condition (11) is explained by its too general character, which is attributed to the absence of any requirements on the recognition operators in $\mathfrak{M}$.

2.2. Correct Algorithms in Algebraic Extensions

Now, consider some possible requirements on the model $\mathfrak{M}$ in order to reduce the complexity of realization of correct algorithms. For this purpose, we will apply the methods based on the following definition. Denote by F a certain class of functions of the same type of the form

$$\forall f \supseteq \mathbf{F} \;\; (f \colon (\mathbf{R}^l)^m \longrightarrow \mathbf{R}^l), \quad m = 1, 2, \dots.$$

A model of recognition operators $\mathfrak{M}$ will be called $\mathbf{F}$-unextendable on X and denoted by $\mathfrak{M}_F$ if the superposition $F \circ \mathfrak{M}$ satisfies the condition $F \circ \mathfrak{M}(X) \subseteq \mathfrak{M}(X)$.

It is easily seen that this condition is in a sense "self-complementary": if, for a class of functions F and a model $\mathfrak{M}$, one fails to show that the latter is $\mathbf{F}$-unextendable, then, augmenting $\mathfrak{M}$ with functions from $\mathbf{F}$ in a standard way, one obtains a model that certainly possesses the above-mentioned property. The model constructed will always be superior to the original one both in the number of operators and in the power of the set of values (this fact underlies the idea of correcting or two-level recognition algorithms). However, a more important fact is that such a model can be expected to be correct under "weaker", compared to (10), conditions (i.e., conditions that are sufficient for (10)). Obviously, the same applies to the case when the **F**-unextendability of the model $\mathfrak{M}$ is proved. Therefore, let us introduce one more definition. The model $\mathfrak{M}$ is called *complete* on the set X^0 if $\mathfrak{M}(X^0) = \mathbf{R}^{ql}$.

It is easily seen that the properties of completeness and correctness are related in the way indicated above; i.e., the completeness implies (10). This provides a methodological basis for obtaining the conditions of completeness.

Now, introduce three types of possible functions F and consider the correctness and completeness conditions for the corresponding models with the recognition operators $\mathfrak{M}_F$. It should be noted that a subsequent choice of F is mainly due to the tradition and is far from being exhaustive. The form of the functions will be determined directly in the space $\mathbf{R}^{\mathbf{q}l}$, which is quite justified because X^0 is assumed to be fixed.

Define a class $\mathbf{F}_1$ as a set of maps of the form

$$\mathbf{F}_1 := \left\{ f \,|\, f(R_1, \ldots, R_m) = \sum_{i=1}^{m} \beta_i R_i \right\}, \quad m = 1, 2, \ldots,$$

where $\beta_i \in \mathbf{R}$, $(\beta_i \geq 0)$, $R_i \in \mathbf{R}^{\mathbf{q}l}$. A class $\mathbf{F}_2$ is defined as a set

$$\mathbf{F}_2 := \left\{ f \,|\, f(R_1, \ldots, R_m) = \sum_{i=1}^{m} \beta_i R_i^{t_i} \right\}, \quad m = 1, 2, \ldots,$$

where $\beta_i \in \mathbf{R}$, $(\beta_i \geq 0)$, $R_i^{t_i} := \left\| (r_{uv}^i)^{t_i} \right\|$ is the t_ith degree of the matrix $R_i = \left\| (r_{uv}^i) \right\| \in \mathbf{R}^{\mathbf{q}l}$ $(t_i \in \mathbf{N})$ defined in the matrix algebra that is commutative with respect to multiplication (i.e., actually defined in a vector algebra).

In order to describe the next class $\mathbf{F}_3$, we will need some additional constructions. First, introduce the set of maps

$$\mathbf{G} := \{ g \,|\, g\colon (\mathbf{R}^l)^m \longrightarrow \mathbf{R}^q \}.$$

Then, using the isomorphism between the space $\mathbf{R}^{\mathbf{q}l}$ and the tensor product $\mathbf{R}^{\mathbf{q}} \otimes \mathbf{R}^l$ of the vector spaces $\mathbf{R}^{\mathbf{q}}$ and $\mathbf{R}^l$ and an arbitrary bilinear map

$$\mathbf{H} := \{ h \,|\, h\colon \mathbf{R}^{\mathbf{q}} \times \mathbf{R}^l \longrightarrow \mathbf{R}^{\mathbf{q}l} \},$$

we can easily construct the required map **F**.

Methods for constructing such maps and their properties (including general conditions of correctness and completeness of the corresponding models of algorithms) are described in [17] in more detail. In the present work, we fix the class $\mathbf{F}_3$, which was first introduced and examined in [17]. For this purpose, we must specify the map **G**. Introduce two more sets of parameterized functions $\mathbf{G}_1\colon \mathbf{R}^l \longrightarrow \mathbf{R}$ and $\mathbf{G}_2\colon \mathbf{R}^m \longrightarrow \mathbf{R}$ and define their superposition $\mathbf{G}_2 \circ (\mathbf{G}_1)^m$ in the space $(\mathbf{R}^l)^m$. Then, it is obvious that one can choose a system of all possible sets with power q in the constructed superposition as **G** (recall that q is related to the dimension of the sample X^0). In [9], it was shown that, for such a **G**, one can take ordinary linear operators from $\mathbf{R}^{\mathbf{q}}$ into $\mathbf{R}^l$ (i.e., matrices in the space $\mathbf{R}^{\mathbf{q}l}$) as **H**. Denoting this matrix space by $\mathbf{L}(\mathbf{R}^{\mathbf{q}}, \mathbf{R}^l)$, we can represent **F** as $\mathbf{F} = \mathbf{L}(\mathbf{R}^{\mathbf{q}}, \mathbf{R}^l) \circ \mathbf{G}$. Finally, define $\mathbf{F}_3$ as a subset of the maps F in which the functions $\mathbf{G}_1$ and $\mathbf{G}_2$ satisfy the conditions

$$\forall \mathbf{g} \in \mathbf{G}_1 \quad \forall y \in \mathbf{R}^l \quad (\mathbf{g}(y) := \eta(y, y_\eta)),$$

where $y_\eta \in \mathbf{R}^l$ is a parameter (depending on η) and η is a metric in $\mathbf{R}^l$;

$$\forall \mathbf{g} \in \mathbf{G}_2 \quad \forall y \in \mathbf{R}^m \quad \left(\mathbf{g}(y) := \mu\left(\sum_{i=1}^{m} y_i \right) \right),$$

where μ is a monotonically decreasing function such that

(i) $\mu(0) = c_1 > 0$ (c_1 is a numerical constant) and

(ii) $\lim_{y \to \infty} \mu(y) = 0$.

Now, we can formulate and prove the following results for the models $\mathfrak{M}_{F_i}$ $(i = 1, 2, 3)$.

Theorem 2. (i) *Model* $\mathfrak{M}_{F_i}$ *is complete on* $\mathbf{X}^0$ *if and only if there are* $q \cdot l$ *operators* $B_{11}, \ldots, B_{ql}$ *such that the set* $(B_{11}(X^0), \ldots, B_{ql}(X^0))$ *forms a basis in* $\mathbf{R}^{\mathbf{q}l}$;

(ii) *Model* $\mathfrak{M}_{F_i}$ *satisfies condition* (10) *on* $\mathbf{X}^0$ *if and only if there exists a finite set of operators* $B_1, \ldots, B_m$ *that satisfy*

$$\begin{cases} \forall k \in \{1, \ldots, m\} \; \exists (i_0, j_0) \\ \in M_1 \; \forall (u, v) \in M_0 (b_{i_0 j_0}^k > m b_{uv}^k) \\ \forall (i, j) \in M_1 \; \forall (u, v) \in M_0 \; \exists k_0 \\ \in \{1, \ldots, m\} (b_{ij}^{k_0} > m b_{uv}^{k_0}). \end{cases} \tag{12}$$

Theorem 3. (i) *Model* $\mathfrak{M}_{F_i}$ *is complete on* X^0 *if and only if there are* $q \cdot l$ *operators* $B_{11}, \ldots, B_{ql}$ *such that the set* $(B_{11}(X^0), \ldots, B_{ql}(X^0))$ *satisfies*

$$\begin{cases} \forall (i, j) \in \mathbf{I} \; \exists (i_0, j_0) \\ \in \mathbf{I} \; \forall (u, v) \neq (i_0, j_0) (b_{i_0 j_0}^{ij} > b_{uv}^{ij}) \\ \forall (i, j) \in \mathbf{I} \; \forall (u, v) \neq (i, j) \; \exists (i_0, j_0) \\ \in \mathbf{I} (b_{ij}^{i_0 j_0} > b_{uv}^{i_0 j_0}); \end{cases} \tag{13}$$

(ii) *Model* $\mathfrak{M}_{F_i}$ *satisfies condition* (10) *on* X^0 *if and only if there exists a finite set of operators* $B_1, \ldots, B_m$ *that satisfy*

$$\begin{cases} \forall k \in \{1, \ldots, m\}\ \exists (i_0, j_0) \\ \in M_1\ \forall (u, v) \in M_0 (b^k_{i_0 j_0} > b^k_{uv}) \\ \forall (i, j) \in M_1\ \forall (u, v) \in M_0\ \exists k_0 \\ \in \{1, \ldots, m\} (b^{k_0}_{ij} > b^{k_0}_{uv}). \end{cases} \quad (14)$$

Theorem 4. (i) *Model* $\mathfrak{M}_{F_3}$ *is complete on* X^0 *if and only if there is a finite set of operators* $B_1, \ldots, B_m$ *such that* $(B_1(X^0), \ldots, B_m(X^0))$ *satisfies*

$$((b^1(x_i) \oplus \ldots \oplus b^m(x_i) \neq b^1(x_j) \oplus \ldots \oplus b^m(x_j)) \Leftrightarrow (i \neq j)); \quad (15)$$

(ii) *Model* $\mathfrak{M}_{F_3}$ *satisfies condition* (10) *on* X^0 *if and only if there exists a finite set of operators* $B_1, \ldots, B_m$ *such that*

$$((b^1(x_i) \oplus \ldots \oplus b^m(x_i) \neq b^1(x_j) \oplus \ldots \oplus b^m(x_j)) \Leftrightarrow (P(x_i) \neq P(x_j))). \quad (16)$$

Let us make a few remarks concerning the notations used in the formulation of the theorems. We used two types of sets for indexing the sets of operators, **I** and $\{1, \ldots, m\}$. The first set is used to indicate both the position of an element in the matrix $B(X^0) = \|b_{ij}\| \in \mathbf{R}^{ql}$ and the number of an operator in the corresponding set, while the second set is used for the latter purpose only. In all cases, the number of an operator is indicated by the upper index. By $b^k(x_i)$, we denote the ith row (corresponding to $x_i \in X^0$) of the matrix of the values of the recognition operator B_k $(k = 1, \ldots, m)$. The symbol $\oplus$ denotes the standard operation of direct summation of vectors. Finally, $P(x_i)$ is the value of the predicate P on $x_i \in X^0$, which is also defined and, according to the formulation of the problem, is known.

Remark. *The sufficiency of the conditions similar to those given in Theorems* 2 *and* 3 *was first demonstrated in* [9, 19, 20]. *The constraints on the model* $\mathfrak{M}$ *and the maps* $\mathbf{F}_2$ *and* $\mathbf{F}_1$ *introduced above are essentially used when proving the necessity of these conditions. It is interesting that such a formulation of the results illustrates the role of each operation from the sets* $\mathbf{F}_2$ *and* $\mathbf{F}_1$ *in the best way. We would like to point out one more operation—a componentwise "inversion" of matrices in the space* $\mathbf{R}^{ql}$*—which can be used for constructing* $\mathbf{F}$ *and reducing the complexity of the realization of algorithms satisfying* (10).

Let us briefly discuss the results obtained. First, note that the conditions obtained can be used to determine the requirements on information in $\mathbf{X}(\mathbf{X}^0)$ under which there exists a correct algorithm in $A_{\mathfrak{M}}$. In our case, such requirements are determined indirectly, in the set $\mathfrak{M}(X^0)$. In order to transfer these properties to information, one should specify the model $\mathfrak{M}$. In most cases, the determination of such requirements is technically rather complicated. Nonetheless, for a number of known models $\mathfrak{M}$, it was shown in [9] that these models satisfy either (13) or (12) and (14), unless very strong conditions are imposed on X^0. As for the conditions obtained in Theorem 4, the situation is different, or even opposite. It is rather difficult to find a reasonably formulated problem and a model $\mathfrak{M}$ for which conditions (15), or at least (16), are not satisfied even for $m = 1$. Moreover, one can see that there is a relation between (16) and the following condition:

$$\forall x_1, x_2 \in X (x_1 \neq x_2 \Leftrightarrow P(x_1) \neq P(x_2)). \quad (17)$$

This relation can be formulated in the form of the following meaningful proposition:

a model $A_{\mathfrak{M}}$ *satisfies the condition* $\Phi_A(X^0) = 1$ *if* (17) *holds*; *when* $\mathfrak{M}$ *is* $\mathbf{F}_3$*-unextendable and* (16) *holds on* X^0, *the model* $A_{\mathfrak{M}}$ *satisfies the above condition if and only if* (17) *holds*.

In spite of its obvious importance, we will not formalize and prove this proposition, because this requires that $\mathfrak{M}$ should be specified. In our opinion, the importance is associated with the fact that such models $A_{\mathfrak{M}}$ are unimprovable in the sense that one can hardly obtain simpler conditions on information for any other model.

In the context of the classes of functions $\mathbf{F}_i$ $(i = 1, 2, 3)$ considered above, the unimprovability can also be imparted a comparative character. This can easily be done by formalizing the following well-known hypothesis: *the stronger the model, the wider the set of solutions obtained, and the weaker the conditions of solvability for any problem.* One can easily prove the following corollary.

Corollary. *The following relation holds irrespective of the choice of* $\mathfrak{M}$:

$$\forall X^0 \ \ (F_1 \circ \mathfrak{M}(X^0) \subseteq F_2 \circ \mathfrak{M}(X^0) \subseteq F_3 \circ \mathfrak{M}(X^0)).$$

Consider the complexity of the realization of correct algorithms. This problem can be divided into two interrelated parts: the complexity of the verification of the corresponding conditions and, if they are satisfied, the complexity of the realization of the appropriate algorithms. Concerning the first part, not many results have been obtained. For instance, we may refer to [9], where conditions of the type (14) were examined for the first time, and it was shown that such operators can be constructed as a result of solving a properly formulated problem on covers, whose complexity is obvious. Apparently, operators that satisfy conditions (15) and (16) can also be constructed by a similar method. After that, the complexity of the realization of correct algorithms is not very high. It was shown in [17] that in the

$\mathbf{F}_3$-unextendable model considered above, such complexity does not exceed the complexity of inverting matrices in the space $\mathbf{R}^{\mathbf{qq}}$, i.e., no higher than $O(|X^0|^3)$. At the same time, this complexity is much higher in all other cases.

3. ALGORITHMS FOR SOLVING THE PROBLEM Z

Comparing the problems $\mathbf{Z}_{A-C}$ described above, we can see that the first problem has a theoretical character, while the second and the third are clearly applied problems. Comparing these problems, we can draw the following alternative propositions concerning the results of their solution: they represent *either a validated result without any practical interpretation, or a result about which the question of validation cannot be raised at all* (*at least until the inductive inference is not validated* [8]). The situation is rather complicated for practical problems because they must be solved and it is desirable to obtain maximally validated results. It is better if this is achieved by the statement alone, rather than by plausible additional assumptions. Is this possible? The answer can be yes, at least in the following sense. The problems $\mathbf{Z}_{A-C}$ can be solved in the framework of a unified set of algorithms. This allows the separation of algorithmic and information aspects of the problem of validation. In this case, the latter problem reduces to proving a certain generality of problem statements and to validating the application of the algorithms constructed to solving at least one of the problems. The validity of the application of the algorithm A_0 to solving the problem $\mathbf{Z}_A$ was proved mathematically rigorously. If it turns out that the results coincide, this will mean that the constructed algorithm is validated, or at least its application is valid, including the application to solving the problems $\mathbf{Z}_C$.

Remark. *Even if the approach described above is realized, there still remains the information aspect of the problem with respect to which the given formulations will never be unified. Therefore, the algorithmic aspects of the problem of validation do not exhaust the problem on the whole. Moreover, not all variants of the problem* $\mathbf{Z}$ *are included in the list considered. Based on the aforesaid, one can state that the view of the problem developed here confirms that it is justified that questions should arise concerning the validation of the algorithms and the solvability of problem* $\mathbf{Z}_C$.

3.1. More Specific Formulation and General Requirements on Algorithms

Let us begin with the description of the set X in which the comparison will be made. In contrast to the problems $\mathbf{Z}_B$ and $\mathbf{Z}_C$, this set is always infinite in the case of problem $\mathbf{Z}_A$. Moreover, in $\mathbf{Z}_A$ and $\mathbf{Z}_B$, along with the spaces with fixed dimensions, the constructions of super- and subspaces are admitted in $\mathbf{Z}_A$ and $\mathbf{Z}_C$, which correspond to redundancy or insufficiency of information. In order to make problems $\mathbf{Z}_{A-C}$ comparable, it is necessary to provide a correct operation of the algorithm on all possible information. To simplify the account and to make it clearer, we will restrict the analysis to the case of Boolean spaces. In other words, the set X is identified with the following construction:

$$\bigcup_{i=1}^{\infty}\mathbf{B}_2^i, \quad \mathbf{B}_2 = \{0, 1\}. \tag{18}$$

In this case, with respect to the space $\mathbf{B}_2^i$, a vector from $\mathbf{B}_2^i$ will be interpreted as a vector whose $(i+1)$th coordinate is not defined, and vice versa.

Now, let us discuss the ways in which objects in space (18) are obtained. In problem $\mathbf{Z}_C$, the classes X_i are described by the learning sample X^0. In problems $\mathbf{Z}_A$ and $\mathbf{Z}_B$, the predicate is known. In $\mathbf{Z}_B$, a finite number of objects from X provides the truth-value predicate P. In $\mathbf{Z}_A$, the number of such objects is infinite and covers the whole set X. Suppose that there exists a constructive method that assigns sets of objects with the true value of the predicate P_i to each class X_i. Combining these objects, we obtain an analog of the learning sample X^0. Thus, we assume that the learning sample is defined for all problems.

Finally, let us discuss the requirements on the algorithms. We formulate two principles that follow immediately from the operation conditions of the corresponding algorithms on the problems $\mathbf{Z}_{A-C}$. The first one is as follows.

1. The correctness of A.

$$\forall x \in X^0\ (\mathrm{A}(x) = (P_1(x), \ldots, P_l(x))). \tag{19}$$

This principle follows from the role of the correctness condition in the solvability of the problem $\mathbf{Z}$, although, from the formal point of view, it does not coincide with the definition of correctness given above.

In order to formulate the second principle, we introduce the following definition. Suppose given objects $x_1 \in \mathbf{B}_2^i$ and $x_2 \in \mathbf{B}_2^{i+1}$ (i is an arbitrary natural number). Fix the coordinate $j \in [1, i+1]$ by which the spaces $\mathbf{B}_2^i$ and $\mathbf{B}_2^{i+1}$ differ from each other. The object x_1 is said to be a subobject of x_2 if, upon elimination of the coordinate j, the remaining part of the vector x_2 coincides with x_1. Obviously, this definition has its inverse (i.e., x_2 can be called a superobject of x_1) and its elementary extension (to the case $i + 2, \ldots$). Now, we obtain the following principle.

2. The object monotonicity of A.

Assume that, for a certain class X_i, the corresponding learning sample X_i^0 consists of a single object x. Suppose, in addition, that certain objects x_1 and $x_2 \in X$

are given. Then, if x_1 is a subobject of x and x_2 is a subobject of x_1, then the following condition is satisfied:

$$P_i^{\mathrm{A}}(x_l) \geq P_i^{\mathrm{A}}(x_2). \quad (20)$$

The principles formulated above are corollaries to the general concepts related to the solution of problems $\mathbf{Z}_{A-C}$. For instance, the principle of correctness applies to problem $\mathbf{Z}_A$ and is preferred for $\mathbf{Z}_B$. The principle of object monotonicity is characteristic of all approximation problems: *the less information is known, the smaller any comparative quantitative estimates should be.* This principle is obvious for $\mathbf{Z}_C$. It is realized in $\mathbf{Z}_B$ as well, but either by parametrizing predicates or by axiomatizing the inference. This is not a drawback, it is just the way the inductive inference works [12, 13, 15]. The principle of object monotonicity is a particular case since $|X_i^0| = 1$; however, such a formulation is sufficient for a generalization to $|X_i^0| > 1$. The character of the generalization depends on the scheme of the algorithm A.

3.2. Description of Algorithms for Solving the Problem $\mathbf{Z}$

Let us describe a parametric family of algorithms $\mathfrak{A}$. First, introduce some agreements and notations. We assume that

(i) The features of the space X, whose number may be infinite, are enumerated, and their order is fixed. Denote the set of these numbers by I.

(ii) The set X^0 is divided into subsets X_i^0 $(i = 1, \ldots, l)$. Objects in X_i^0 are also enumerated, and their order is fixed. Denote the sets of these numbers by $\mathbf{J}_k$ $(k = 1, \ldots, l)$.

(iii) The set $\{1, \ldots, l\} \times I$ is associated with the matrix of real numbers $\|a_{ij}\|$, where $i \in \{1, \ldots, l\}$, $j \in \{1, \ldots, |\mathbf{I}|\}$, $a_{ij} \in \mathbf{R}$, and

$$\forall i, j \ \ (a_{ij} \geq 0), \ \ \forall i \left(\sum_j a_{ij} > 0\right). \quad (21)$$

Now, the algorithm A can be described as a sequence of the following steps.

Step 1. Fix an object $x \in X$ and go to step 2.

Step 2. For each $i \in \{1, \ldots, l\}$ and all $x_j \in X_i^0$ $(j \in \mathbf{J}_i)$, calculate

$$\mu_{\mathbf{A}}^{i,j}(x, x_j) = \max\left\{0, \left(\sum_{u \in \mathbf{I}} (-1)^t a_{iu}\right)\left(\sum_{u \in \mathbf{I}} a_{iu}\right)^{-1}\right\},$$

where

$$t = \begin{cases} 1, \text{ if } \ x_u \neq x_{ju}, \\ 2, \text{ otherwise.} \end{cases}$$

Step 3. If all $x_j \in X_i^0$ are exhausted, calculate

$$P_i^{\mathbf{A}}(x) = \max_{x_j \in X_i^0} \{\mu_{\mathbf{A}}^{i,j}(x, x_j)\}.$$

If the numbers of classes $i \in \{1, \ldots, l\}$ are not exhausted, return to step 2. Otherwise, go to the next step.

Step 4. If the set of objects x is not exhausted, return to step 1. Otherwise, the algorithm ends.

End of algorithm.

It is easily seen that the algorithm A is uniquely determined by the choice of specific parameters $\|a_{ij}\|$. This is mainly associated with nonalgorithmic considerations, namely, with the choice of the description space of objects in the set X, with the desire to impart some meaningful interpretation to the numbers $a_{ij} \in \mathbf{R}$ and the values $P_i^{\mathbf{A}}(x)$, with the formalization of the concept of the "similarity" of objects, etc. For better understanding of the problem and for the analysis of the initial information, it is desirable to interpret the quantities $P_i^{\mathbf{A}}(x)$ in terms of the application domain. Therefore, the way of choosing the parameters $\|a_{ij}\|$ is essential, although, as will be shown below, the operation of algorithm A does not depend on this choice.

Let us describe one possible scheme for determining the parameters $\|a_{ij}\|$.

Step 1. For given $\mathbf{I}$ and X_i^0, execute the following sequence of steps.

Step 2. Fix the feature number $j \in \{1, \ldots, |\mathbf{I}|\}$ and, for each $i \in \{1, \ldots, l\}$, calculate

$$b_{ij} = \left(\sum_{x_u \in X_i^0} x_{uj}\right)\left(\left|X_i^0\right|\right)^{-1}.$$

Step 3. If not all the features $j \in \{1, \ldots, |\mathbf{I}|\}$ are exhausted, return to step 2. Otherwise, calculate

$$b_j = \left(\sum_{i=1}^{l} b_{ij}\right) l^{-1}, \quad a_{ij} = |b_{ij} - b_j|.$$

End of algorithm.

It can easily be verified that the parameters a_{ij} thus constructed satisfy condition (21) if $l > 1$ and $X_i^0 \neq \varnothing$ for all i. Moreover, one can specify the limits of the parameters: $a_{ij} \in [0, 1)$. Denote a family of algorithms thus parametrized by $\mathfrak{A}(a)$.

Let us consider one more question related to the choice of the space X in the form (18). Here, objects of both infinite and finite dimensions are admissible. In the latter case, there are the following variants:

(i) Objects in X^0 may belong to different subspaces and therefore be incomparable from the point of view of the scheme for calculating the parameters $\|a_{ij}\|$.

(ii) The dimension of the object x under study may be incomparable with the dimensions of objects in any finite sample X^0.

Both these cases refer to the way of constructing the samples X^0. In $\mathbf{Z}_C$, no problems arise since the objects are assumed to be absolutely comparable. In $\mathbf{Z}_A$ and $\mathbf{Z}_B$, one must first obtain a system of predicates $P = (P_1, \ldots, P_l)$. For $\mathbf{Z}_B$, this is clear, even if the predicates are not defined explicitly. Therefore, we can assume that, for all $i \in \{1, \ldots, l\}$, there exist $P_i(x)$, which, at worst, are defined in different subspaces. $X_i^0 = \{x|P_i(x) = 1\}$ can be constructed, for example, by reducing $P_i(x)$ to a *disjunctive normal form* (DNF). The definition of the predicate can be extended with regard to the meaningful sense of information, and one faces no problems with calculating the parameters. If such an extension is impossible, it is sufficient to modify step 3 of the scheme for calculating the parameters. For example, in calculating b_j, the summation is performed only over those $i \in \{1, \ldots, l\}$ for which the value of b_{ij} (step 2) is determined, whereas, for all other (i, j), it is assumed that $a_{ij} = 0$. A similar procedure can be applied when a certain feature j is not defined in the predicates $P_i(x)$ for any $i \in \{1, \ldots, l\}$. If the object x under study has incomparable dimensions, then there is no such problem in the modification proposed. When calculating t, it is sufficient to consider a logical, rather than an algebraic, mismatch between the values of the feature u.

In problem $\mathbf{Z}_A$, the situation is quite different. Even if it were possible to obtain a system of predicates $P = (P_1, \ldots, P_l)$, the problem of the infinite sample X^0 would remain. However, we consider algorithmic aspects of the problem $\mathbf{Z}$. Therefore, we still assume that such a sample exists for the problem $\mathbf{Z_A}$ as well. The method of constructing the parameters $\|a_{ij}\|$ is identical to that described above for problem $\mathbf{Z}_B$.

3.3. Analysis of the Family of Algorithms $\mathfrak{A}(a)$

Let us show that any algorithm in $\mathfrak{A}(a)$ is a solution to the problem $\mathbf{Z}$. For this purpose, we formulate the following propositions.

Proposition 6. *For any* $\mathrm{A} \in \mathfrak{A}(a)$, $\forall x \in X\ (P_i^{\mathrm{A}}(x) \in [0, 1])$ *and the correctness condition* (19) *is satisfied.*

Proposition 6 implies several obvious corollaries related to the algorithmic aspects of monotonicity. They will be formulated in the form of the properties of the algorithm A. First, recall that the algorithm A_0 satisfies the following condition by definition:

$$P_i^{\mathrm{A}_0}(x) = \begin{cases} 1, & \text{if } x \in X_i^0, \\ 0, & \text{otherwise.} \end{cases}$$

Property 2. *For any* $\mathrm{A} \in \mathfrak{A}(a)$, *we have*

$$\forall x \in X\ \forall i \in \{1, \ldots, l\}\ \ (P_i^{\mathrm{A}}(x) \ge P_i^{\mathrm{A}_0}(x)).$$

This property can be interpreted as a majorization relation between algorithms A and A_0.

Property 3. *For any* $\mathrm{A} \in \mathfrak{A}(a)$, *we have*

$$\forall i \in \{1, \ldots, l\}\ \forall x \notin X_i^0 (1 > P_i^{\mathrm{A}}(x))$$
$$\Leftrightarrow \forall j \in \{1, \ldots, |\mathbf{I}|\}\ (a_{ij} > 0).$$

This property determines the condition under which the natural difference (separability) of the sets X^0 and $X \backslash X^0$ "is transformed" by algorithm A into the difference between the values of the predicate $P_i^{\mathbf{A}}$. The latter difference also has the character of separability and can be interpreted as the categorical monotonicity on the boundary of the set. The condition in Property 3 should be determined by meaningful considerations (i.e., by the formulation of the problem) and the scheme of calculating the parameters $\|a_{ij}\|$.

Now, consider the principle of object monotonicity. Let us extend condition (20), whose validity for $\mathfrak{A}(a)$ is almost obvious, and prove the following proposition.

Proposition 7. *Given a sample* X^0, *suppose that, for certain* $x \in X$ and $i \in \{1, \ldots, l\}$, *x is a subobject of all objects included in the corresponding sample* X_i^0. *Then, any* $\mathrm{A} \in \mathfrak{A}(a)$ *has the following properties*:

(a) *for any x' that is a subobject of x, the following condition is satisfied*:

$$(P_i^{\mathbf{A}}(x) \ge P_i^{\mathbf{A}}(x')). \tag{22}$$

(b) *for any x' that is a superobject of x*, *condition* (22) *is satisfied if and only if x' is a subobject of at least one object from* X^0.

In this proposition, condition (20) is somewhat extended. In the general case, analysis of the estimate for $P_i^{\mathbf{A}}(x)$ is rather complicated. This is mainly related to the properties of the relation "object–subobject" which is nonsymmetric, transitive, and has indefinite reflexivity. The complexity is of technical character; therefore, complete analysis of the principle of object monotonicity cannot be carried out in the framework of this study.

Thus, we have nearly demonstrated that the family of algorithms $\mathfrak{A}(a)$ solves problem $\mathbf{Z}$ (but we have not shown that sample X^0 can be constructed for $\mathbf{Z}_A$). Obviously, if we overcome this limitation, we will obtain the required solution.

For this purpose, we invoke a conventional scheme for proving the solvability of formal theories. First, we choose the formalization language and formulate the basic principles (axioms and inference rules). Then, we define a coding function and show that any statement of the theory can be associated with a unique number (a natural number). This number is connected to the proof (course of inference). In this case, the problem is reduced to the analysis of the structure of the sets, which can be obtained by some characterization of statements, and to the investigation of their properties. If it turns out that the characteristic function of the subsets is calculable (or recursive), then the original theory is solvable. One can construct an appropriate algorithm that identifies the properties of each new statement by the initial information.

For $\mathbf{Z}_A$, the structure is very simple: the set of statements (formulas) splits into three disjoint subsets: tautologies, identically false statements, and neutral statements. The corresponding algorithm is A_0. It remains to choose appropriate coding in order to obtain elements of space (18) and in order that the number of such elements be finite. In the context of the above reasoning, it is sufficient to show the existence of such coding and the subsequent applicability of algorithms $A \in \mathfrak{a}(a)$.

Now, the following theorem can be formulated.

Theorem 5. *For $\mathbf{Z}_A$, there exists a finite sample X^0 such that, for all algorithms* A *from $\mathfrak{a}(a)$, we have*

$$\forall x \in X \ \ (A(x) = A_0(x)). \tag{23}$$

Proof. Consider a scheme for constructing the required sample X^0. We will show that all elements of this scheme at least exist.

Thus, each object (formula) in $\mathbf{Z}_A$ can be coded. Suppose that the procedure of Gödel enumeration of the objects has already been performed. As a result, we obtain disjoint subsets of the set of natural numbers $\mathbf{N}$. Denote these subsets by $\mathbf{N}_i$ ($i = 1, 2, 3$). For the solvability of $\mathbf{Z}_A$, it is essential that a union of the subsets $\mathbf{N}_i$ either is a subset of or coincides with the set $\mathbf{N}$, and the subsets $\mathbf{N}_i$ themselves are closed. Now, continue the coding to obtain elements of space (18). For this purpose, we choose a function f: $\mathbf{N} \longrightarrow \mathbf{N}$ that simply reorders the elements of the subsets $\mathbf{N}_i$ in such a way that

$$f(n_1) = f(n_2) - 1 = f(n_3) - 2 \geq 3 \tag{24}$$

at least for three $n_i \in \mathbf{N}_i$. At the next stage of coding, we convert the elements obtained to binary representation. Include the binary representation of the corresponding element $f(n_i)$ into the sample X_i^0. Condition (24) implies that these elements differ only in the two last digits; therefore, the following condition is satisfied in calculating the parameters $\|a_{ij}\|$: $\forall i\ \forall j > 2\ (a_{ij} = 0)$. It is also easy to calculate the corresponding values for the lower order digits. Now, if we require that f should characterize the subsets $\mathbf{N}_i$ in the sense (24), then the validity of (23) becomes obvious.

Let us show that such characterization is possible. For this purpose, we construct f with the use of the functions f_1 and f_2 in such a way that $\forall n \in \mathbf{N}$ ($f(n) = f_1(n) + f_2(n)$) and require that the functions f_1 and f_2 satisfy the following conditions:

$$f_1(n_1) = f_1(n_2) = f_1(n_3)$$

for n_1, n_2, and n_3, which appear in condition (24), and

$$f_2(n) = \begin{cases} 1, \text{ if } n \in \mathbf{N}_1, \\ 2, \text{ if } n \in \mathbf{N}_2, \\ 3, \text{ if } n \in \mathbf{N}_3 \end{cases}$$

for any $\forall n \in \mathbf{N}$. The existence of the appropriate function f_1 is obvious. As regards f_2, its existence is ensured by the well-known topological lemma by Uryson. The applicability of this lemma is ensured by the conditions imposed on the subsets $\mathbf{N}_i$.

It remains to note that one can apply the same algorithm A_0 to construct the formulas corresponding to the numbers n_1, n_2, and n_3. Since these numbers are arbitrary, it is sufficient to choose any three of them and obtain their Gödel code.

Remarks.

1. Condition (24) is not unique. Other similar conditions are also possible for the same scheme of proof.

2. Some other objects whose codes satisfy condition (24) can also be included in the learning sample X^0. For the algorithms from $\mathfrak{a}(a)$, (23) is satisfied in this case as well.

3. The function f_2 is equivalent to the algorithm A_0; therefore, if such a function is available, there is no need to construct A_0. This is not quite so in the context of assumptions made in the statement, since f_2 is a part of some coding f. Constructively, the latter can be obtained in quite various ways. However, something can be said about A_0 only if the existence of f_2 is proved; but this is a question of the analysis of f, or, more precisely, of the structure of the codes obtained. Moreover, a *solution to any problem is related to the choice of appropriate coding*.

4. Examples of problems that can be solved by the scheme designed for $\mathbf{Z}_A$ can be easily represented. These are, for example, a parity recognition problem on the set of natural numbers and certain other problems.

CONCLUSIONS

One can draw the following conclusions from the results obtained. The *first* conclusion concerns the problem formulated in the title of this work: *the choice of algorithms for solving a specific practical problem must be validated, because the statement of a pattern recognition problem is nondegenerate, at least in the* *algorithmic sense.* We would like to note in addition that the above statement holds in the informational sense as well; however, this result follows from other works cited in the text. The *second* conclusion concerns the correctness condition, or, more precisely, to its role in PRT: *the correctness of at least one model always testifies to the consistency of the problem in the sense of formulation and, under certain conditions, to the completeness of information as well.* Note that this role of the correctness condition is not as insignificant as may seem at first sight. To verify this, it suffices to consider, e.g., the problem of validation of axiomatizable systems in set theory. A validity criterion for such systems is their consistency and completeness, but in a stronger sense than those in the problem of recognition. In this context, it is relevant to recall that, in such axiomatic systems, the method of resolutions is used for inference, and the role of this method is comparable with the role of correct algorithms for PRT. The *third* conclusion is related to the applicability of correct algorithms: *for practical application, this condition must be supplemented with a certain condition on information.* This means that the correctness (as a property of algorithms) will never guarantee the solvability of the problem on a global level. For this purpose, it is necessary to supplement it with a certain condition that would guarantee finiteness of choice for an infinite number of objects. For example, this may be the finite capacity of a set of algorithms, the convergence to the limit with respect to information, stability, etc. Any such condition should narrow down the set of correct algorithms applicable to solving the corresponding class of problems.

The following conclusions are not so indisputable and seemingly do not follow directly from the results obtained. However, we will formulate them because they reflect the authors' experience in solving various practical problems in the field of recognition and are of great methodological importance. Thus, the *fourth* conclusion is as follows: *in the state of the art in PRT, the development of one more model of algorithms will have virtually no effect either on theory or on practice.* In other words, the time has come for the qualitative interpretation of the problems of recognition on the basis of the experience and technical results accumulated over recent decades. One may easily verify this statement: it suffices to try to solve the problems mentioned in the Introduction. In this case, it can be stated almost for certain that, for any model and any algorithms, there exists information that leads to the necessity of rejecting an algorithm irrespective of the methods used. This result is related to the *fifth* conclusion: *solving a recognition problem is a process whose goal is an exact analytical characterization of classes.* Most probably, such a characterization with the help of models and tools of PRT (as it is understood today) is impossible in principle. If we assume that the inductive nature of the problem can somehow be overcome, there still remains too wide a variety of models and solutions. The latter fact is always indicative of a poor formulation of the problem. Finally, the last, *sixth*, conclusion, which seems to need no comment, is as follows: *a specific problem and its solution are always more important than any theoretical constructions and, even more so, preferences, whatever be their arguments.*

Thus, we have proved that the algorithms $\mathfrak{a}(a)$ solve any of the problems $\mathbf{Z}_{A-C}$; moreover, for $\mathbf{Z}_A$, any $\mathbf{A} \in \mathfrak{a}(a)$ is equivalent to $\mathbf{A}_0$. Therefore, these algorithms behave, in a certain sense, correctly for other problems as well; at least, they satisfy the principle of correctness on $\boldsymbol{X}^0$ and are monotonic on $\boldsymbol{X} \backslash \boldsymbol{X}^0$. Due to the general character of the formulations, the applicability of the algorithm itself (as a scheme for converting the initial information into a result) is beyond doubt. This is quite sufficient from the viewpoint of the goals set forth in this study.

However, there still remain informational aspects of the problem. Certain conclusions can be drawn from the results obtained in this direction as well. The most important conclusions concern the way of constructing, as well as the structure of, $\boldsymbol{X}^0$. It is obvious that the main difficulties in solving problems (not only practical problems) are associated with the choice of the appropriate method of coding information. If the problem has a solution, then there exist finite coding and a finite sample for which the problem of constructing an algorithm of type $\mathbf{A}_0$ is of a technical nature. One can determine the parameters of this coding, the structure of the set $\boldsymbol{X}^0$, etc. This can be done as a result of experiments, which are an essential part of problems $\mathbf{Z}_B$ and $\mathbf{Z}_C$.

REFERENCES

1. K. Gödel, "Uber formal unentscheidbare Satze der Principia Mathematica und vervandter System. I," Monatsh. Math. Phys. **38**, 173–198 (1931).
2. J. Barwise, *Handbook of Mathematical Logic* (North-Holland, Amsterdam, 1977).
3. A. Thayse and P. Gribomont, *Approche logique de l'intelligence artificielle.1 De la logique classique a la programmation logique* (Bordas, Paris,1988).
4. J.-L. Lauriere, *Intelligence artificielle. Resolution de problemes par l'homme et la machine* (Eyrolles, Paris, 1987).
5. N. J. Nilsson, *Learning Machines* (McGraw-Hill, New York, 1965).
6. V. V. Krasnoproshin and V. A. Obraztsov, "The Problem of Algorithms Choosing in Pattern Recognition," Pattern Recognit. Image Anal. **6** (2), 188–199 (1996).
7. V. V. Krasnoproshin and V. A. Obraztsov, "Learning Recognition as a Problem of Choice," in *Digital Processing* (Minsk, 1988), Vol. 2, pp. 80–94.
8. B. N. Pyatnitsyn, "A Problem on the Relation between Induction and Deduction," in *Methods of Logical Analysis* (Nauka, Moscow, 1977).
9. Yu. I. Zhuravlev, "Algebraic Methods in Problems of Recognition and Classification," in *Recognition, Classification, and Prediction: Mathematical Methods and*

Their Application, Ed. by Yu. I. Zhuravlev (Nauka, Moscow, 1988), Vol. 1, pp. 9–17.

10. V. N. Vapnik, "Inductive Principles of Seeking Empirical Regularities," in *Recognition, Classification, and Prediction: Mathematical Methods and Their Application*, Ed. by Yu. I. Zhuravlev (Nauka, Moscow, 1988), Vol. 1, pp. 17–82.
11. U. Grenander, *Lectures in Pattern Theory*, Vol. 1: *Pattern Synthesis* (Springer, New York, 1976).
12. G. Polya, *Mathematics and Plausible Reasoning* (Princeton Univ. Press, New Jersey, 1954).
13. H. E. Kyburg, *Probability and Inductive Logic* (Macmillan, London, 1970).
14. J. Hintikka, *Towards a Theory of Inductive Generalization* (Reading in the Philosophy of Science) (New York, 1970).
15. K. Popper, *The Logic of Scientific Discovery* (London, 1959).
16. G. Kreisel, "Foundations of Intuitionistic Logic," in *Proc. 1960 Int. Congr. "Logic, Methodology and Philosophy of Science"* (Stanford, Stanford Univ. Press, 1962), pp. 198–210.
17. V. V. Krasnoproshin and V. A. Obraztsov, "Two-Level Models of Recognition Algorithms," Zh. Vychisl. Mat. Mat. Fiz. **25** (10), 1534–1547 (1985).
18. K. V. Rudakov, "Algebraic Theory of Universal and Local Constraints for Classification Problems," in *Recognition, Classification, and Prediction: Mathematical Methods and Their Application*, Ed. by Yu. I. Zhuravlev (Nauka, Moscow, 1988), Vol. 1, pp. 176–201.
19. Yu. I. Zhuravlev, "Extremal Algorithms in Algebra over Incorrect Algorithms," Dokl. Akad. Nauk SSSR **237** (3), 509–512 (1977).
20. Yu. I. Zhuravlev and I. V. Isaev, "Construction of Recognition Algorithms That Are Correct for a Given Sample," Zh. Vychisl. Mat. Mat. Fiz. **19** (3), 726–738 (1979).
21. Yu. I. Zhuravlev, "An Algebraic Approach to Solving Recognition or Classification Problems," Problemy Kibernetiki **33**, 5–68 (1987).

Viktor Vladimirovich Krasnoproshin. Born 1947. Graduated from Belarussian State University in 1974. Received candidate's degree in 1979. Scientific interests: decision-making theory, pattern recognition, and artificial intelligence. Author of more than 130 papers, including 4 monographs.

Vladimir Alekseevich Obraztsov. Born 1953. Graduated from the Belarussian State University in 1979. Received candidate's degree in 1986. Scientific interests: pattern recognition, artificial intelligence, and inductive logic. Author of more than 50 papers.