Modeling Processes of Inferring Good Maximally Redundant Tests

Xenia Naidenova and Vladimir Parkhomenko

***Abstract*—Good test analysis is considered. Two kinds of classification subtasks are defined: attributive and object ones. Some ideas of modeling and optimization of inferring good maximally redundant tests are formalized. An algorithm of inferring good maximally redundant tests based on the decomposition into attributive subtasks is given, where good maximally redundant tests are regarded as concepts of formal concept analysis. An approach to incremental inferring good maximally redundant tests is considered.

***Keywords*—Good classification test, modeling, Galois mappings, decomposition, implications, inferring rules, incremental, formal concept analysis.

TABLE I
EXAMPLE OF CLASSIFICATION

No	Height	Color of Hair	Color of Eyes	KL
1	Low	Blond	Blue	$k(+)$
2	Low	Brown	Blue	not $k(+)$
3	Tall	Brown	Hazel	not $k(+)$
4	Tall	Blond	Hazel	not $k(+)$
5	Tall	Brown	Blue	not $k(+)$
6	Low	Blond	Hazel	not $k(+)$
7	Tall	Red	Blue	$k(+)$
8	Tall	Blond	Blue	$k(+)$

I. Introduction

THE aim of the paper is to develop the Good Test Analysis (GTA), which is one of machine learning methods based on the notion of good diagnostic (classification) test (GCT). GCT is understood as an approximation of a given classification on a given set of objects [1] and serve as a basis for inferring implicative, functional dependencies and association rules from datasets.

Denote by M the set of attribute values such that $M = \{\cup \mathrm{dom}(\mathrm{attr}), \mathrm{attr} \in U\}$, where $\mathrm{dom}(\mathrm{attr})$ is the set of all values of attr and U is a given set of attributes, see an example in Tab. I. Let G be the set of objects; $G = G_+ \cup G_-$, where G_+ and G_- are the sets of positive and negative objects, respectively. Denote by $I(B), B \subseteq M$, the set of all the objects in description of which B appears. $I(B)$ is called the interpretation of B in the power set 2^G. If $I(B)$ contains only G_+ objects and the number of these objects more than 2, then we call B (as a subset of attribute values) a description of some positive objects and $(I(B), B)$ a **test** for G_+. A pair $(I(B), B)$ is a **good test** for G_+ if and only if it is a test and no such subset $C \subset M$ exists that $I(B) \subset I(C) \subseteq G_+$.

It is not difficult to see that if $(I(B), B)$ is a test for positive objects, then the following implicative dependency is satisfied: $B \rightarrow G_+$. In Tab. I, $(I(\{\text{Blond, Hazel}\}) = \{4,6\}, \{\text{Blond, Hazel}\})$ is a test for class "not $k(+)$", but not good one, and test $(I(\{\text{Hazel}\}) = \{3,4,6\}, \{\text{Hazel}\})$ is a good test for the same class.

Since, however, the task of inferring all good tests for a class of objects is NP-complete, we introduce the decomposition of the main task into subtasks for which searching for good tests is greatly simplified. Some ideas of limiting the search for association rules by selecting instances that have at least one common attribute value with all instances in a specific class in data set are advanced in [2]. Some results in clustering via partial implications are in [3]. This method significantly reduce the target data set. A theoretical study of algorithms for decomposition of numerical semigroup is given in [4].

It would be expedient to include the algorithms of good classification test mining in the experts reasoning process, so as this process would be governed by some consecutively formed goals and sub-goals.

The rest of the paper is organized as follows. Sec.II is devoted to defining the characteristic properties of classification tests and a notion of good maximally redundant tests (GMRTs) as concepts of the formal concept analysis [5]. Sec.III gives the decomposition of the main task of inferring GMRTs into two kinds of subtasks. Sec.IV is devoted to describing the main algorithm for solving any kinds of considered subtasks. Some ideas of modeling and optimization of inferring GCT proposed in [6] are formalized and supplied with an algorithm based on one of the kinds of subtasks in Sec. V. In Sec. V and Sec. VI we introduce some effective procedures for selecting and ordering subtasks, in particular, the evaluation of the number of subtasks to be solved and the depth of recursion [6]. Sec. VII describes the application of two kinds of subtasks for an approach to incremental constructing good maximally redundant classification tests.

II. Good Maximally Redundant Tests as Formal Concepts

Let $G = \overline{1, N}$ be the set of objects indices and $M = \{m_1, m_2, \ldots, m_j, \ldots m_m\}$ be the set of attributes values (objects and values, respectively). Each object is described by a set of values from M. The object descriptions are represented by rows of a table the columns of which are associated with the attributes taking their values in M (see, please, Tab.I).

Xenia A. Naidenova obtained Ph.D. in Computer Science from the St.Petersburg Electrotechnical University. Xenia is a senior researcher of the Group of Psycho Diagnostic Systems Automation at the Military Medical Academy, St.Petersburg, Russia. Email: ksennaidd@gmail.com

Vladimir A. Parkhomenko is a software engineer in the St.Petersburg State Polytechnical University, St.Petersburg, Russia. Email: parhomenko.v@gmail.com

Assume $A \subseteq G$, $B \subseteq M$. Denote by B_i, $B_i \subseteq M$, $i = \overline{1, N}$ the description of object with index i. The Galois connection between the ordered sets $(2^G, \subseteq)$ and $(2^M, \subseteq)$ is defined by the following mappings called derivation operators [7], [8]: for $A \subseteq G$ and $B \subseteq M$, $A' = \mathrm{val}(A) = \{$intersection of all $B_i |\, B_i \subseteq M, i \in A\}$ and $B' = \mathrm{obj}(B) = \{i |\, i \in G, B \subseteq B_i\}$.

Two generalization operations that are closure operators [9] are introduced in [10]: $\mathrm{generalization\_of}(B) = B'' = \mathrm{val}(\mathrm{obj}(B))$ and $\mathrm{generalization\_of}(A) = A'' = \mathrm{obj}(\mathrm{val}(A))$. A set A is closed if $A = \mathrm{obj}(\mathrm{val}(A))$. A set B is closed if $B = \mathrm{val}(\mathrm{obj}(B))$. For $A \subseteq G$ and $B \subseteq M$, a pair (A, B), $A \subseteq G$, $B \subseteq M$, $A'' = A$ is a formal concept, left (A) and right (B) parts of which are called extent and intent, respectively. Let us recall the main definitions of GTA [1].

A Diagnostic Test (DT) for the positive examples G_+ is a pair (A, B) such that $B \subseteq M$, $A = B' \neq \emptyset$, $A \subseteq G_+$, $B \not\subseteq g'$ $\forall g \in G_-$. Equivalently, $B' \cap G_- = \emptyset$. A diagnostic test (A, B) for G_+ is **irredundant** if any narrowing $B_* = B \setminus m$, $m \in B$ implies that $(obj(B_*), B_*))$ is not a test for G_+. A diagnostic test (A, B) for G_+ is **maximally redundant** if $obj(B \cup m) \subset A$ for all $m \notin B$ and $m \in M$. In general case, a set B is not closed for diagnostic test (A, B), i. e., a diagnostic test is not obligatory a concept of FCA. This condition is true only for the maximally redundant tests.

A diagnostic test (A, B) for G_+ is **good** if and only if any extension $A_* = A \cup i$, $i \notin A$, $i \in G_+$ implies that $(A_*, val(A_*))$ is not a test for G_+. If a good test (A, B) for G_+ is maximally redundant (GMRT), then any extension $B_* = B \cup m$, $m \notin B$, $m \in M$ implies that $(\mathrm{obj}(B_*), B_*)$ is not a good test for G_+.

Any object description $d = g'$, $g \in G$ in a given classification context is a maximally redundant set of values because for any value $m \notin d$, $m \in M, \mathrm{obj}(d \cup m)$ is equal to $\emptyset$. Denote by D_+ and D_- a set of descriptions for $g \in G_+$ and $g \in G_-$, respectively.

In Tab.I, $((1, 8)$, Blond Blue$)$ is a GMRT for $k(+)$ but it is irredundant one, simultaneously; $((2, 5), \{$Brown, Blue$\})$ is a test for not $k(+)$ but not a good one; and $((2, 3, 5)$, Brown$)$ is a GMRT for not $k(+)$.

In the paper, we deal only with GMRTs. The fact that every good irredundant test (GIRT) (as minimal generator) [1] is contained in one and only one GMRT implies one of the possible methods of searching for GIRTs for a given class of objects:

1) find all GMRTs for a given class of objects;
2) for each GMRT, find all GIRTs contained in it.

III. The Decomposition of Inferring Good Maximally Redundant Tests into Subtasks

Two kinds of subtasks of Inferring GMRTs [7] are described in the section, for a given set G_+ :

1) given a set of values $B \subseteq M$, $\mathrm{obj}(B) \neq \emptyset$, B is not included in any description of negative object, find all GMRTs $(\mathrm{obj}(B_*), B_*)$ such that $B_* \subset B$;
2) given a non-empty set of values $X \subseteq M$ such that $(\mathrm{obj}(X), X)$ is not a test for positive objects, find all GMRTs $(\mathrm{obj}(Y), Y)$ such that $X \subset Y$, $Y \subseteq M$.

The first subtask is useful to find all GMRTs intents of which are contained in the description of an object g. This subtask is considered in [11] for fast incremental obtaining formal concepts.

The **object projection** of a positive object description t on the set D_+ is defined as follows: $\mathrm{proj}(t) = \{z |\, z = t \cap g' \neq \emptyset, g \in G_+$ and $(\mathrm{obj}(z), z)$ is a test for $G_+\}$.

The **value projection** $\mathrm{proj}(m)$ of a given value m on a given set D_+ is defined as follows: $\mathrm{proj}(m) = \{g', g \in G_+ |\, (\mathrm{obj}(g') \cap \mathrm{obj}(m) \cap G_+) \neq \emptyset)\}$.

Algorithm ASTRA, based on value projections, has been advanced in [12]. Algoritm DIAGaRa, based on object projections, has been proposed in [13].

The projections define the methods to construct two kinds of sub-contexts of the main classification context in accordance with two kinds of subtasks determined in this section. The following **theorem** gives the foundation of reducing subcontexts [12].

Theorem 1: Let $X \subseteq M$, $(\mathrm{obj}(X), X)$ be a maximally redundant test for positive objects and $\mathrm{obj}(m) \subseteq \mathrm{obj}(X), m \in M$. Then m can not belong to any GMRT for positive objects different from $(obj(X), X)$.

Proof: Case 1. Suppose that m appears in Y and $(\mathrm{obj}(Y), Y)$ is a GMRT for positive objects different from $(\mathrm{obj}(X), X)$. Then $\mathrm{obj}(Y)$ is a proper subset of $\mathrm{obj}(m)$. However we have that $obj(m) \subseteq obj(X)$ and hence $\mathrm{obj}(Y)$ is a proper subset of $\mathrm{obj}(X)$. However it is impossible because the set of GMRTs is a Sperner system and hence $\mathrm{obj}(Y)$ and $\mathrm{obj}(X)$ does not contain each other [14].

Case 2. Let $(\mathrm{obj}(X), X)$ be the maximally redundant test for positive objects but not a good one. Suppose that there exists a GMRT $(\mathrm{obj}(Y), Y)$ such that m appears in Y. Next observe that $\mathrm{obj}(Y)$ is a proper subset of $\mathrm{obj}(m)$ and $\mathrm{obj}(Y)$ is a proper subset of $\mathrm{obj}(X)$. Then $X \subset Y$ and X is not a maximally redundant test. We have a contradiction. ■

Consider some example of reducing subcontext (see, please, Tab.I). Let $\mathrm{splus}(m)$ be $\mathrm{obj}(m) \cap G_+$ or $\mathrm{obj}(m) \cap G_-$ and SPLUS be $\{\mathrm{splus}(m) |\, m \in M\}$. In Tab.I, we have for values "Hazel, Brown, Tall, Blue, Blond, and Low", respectively, $\mathrm{SPLUS} = \mathrm{obj}(m) \cap G_- = \{\{3, 4, 6\}, \{2, 3, 5\}, \{3, 4, 5\}, \{2, 5\}, \{4, 6\}, \{2, 6\}\}$.

We have $\mathrm{val}(\mathrm{obj}(\mathrm{Hazel})) = \mathrm{Hazel}$, hence $((3, 4, 6), \mathrm{Hazel})$ is a test for G_-. Then value "Blond" can be deleted from consideration, because $\mathrm{splus}(\mathrm{Blond}) \subset \mathrm{splus}(\mathrm{Hazel})$. Delete values Blond and Hazel from consideration. After that the description of object 4 is included in the description of object 8 of G_+ and the description of object 6 is included in the description of object 1 of G_+. Delete objects 4 and 6. Then for values "Brown, Tall, Blue, and Low", respectively, $\mathrm{SPLUS} = \{\{2, 3, 5\}, \{3, 5\}, \{2, 5\}, \{2\}\}$. Now we have $\mathrm{val}(\mathrm{obj}(\mathrm{Brown})) = \mathrm{Brown}$ and $((2, 3, 5), \mathrm{Brown})$ is a test for G_-. All values are deleted and all GMRTs for G_- have been obtained.

IV. The background Algorithm ASTRA for GMRTs construction

The initial information for finding all the GMRTs contained in a positive object description is the projection of it on current set positive objects. It is essential that the projection is a subset of object descriptions defined on a certain restricted subset t_* of values. Let s_* be the subset of indices of objects the descriptions of which produce the projection. In the projection, $\mathrm{splus}(m) = \mathrm{obj}(m) \cap s_*$, $m \in t_*$.

Let STGOOD be the partially ordered set of elements s satisfying the condition that $(s, \mathrm{val}(s))$ is a good test for D_+. The basic recursive procedure for solving any kind of subtask consists of the following steps:

1) Check whether $(s_*, \mathrm{val}(s_*))$ is a test and if so, then s_* is stored in STGOOD if s_* corresponds to a good test at the current step; in this case, the subtask is over. Otherwise the next step is performed.
2) The value m can be deleted from the projection if $\mathrm{splus}(m) \subseteq s$ for some $s \in \mathrm{STGOOD}$.
3) For each value m in the projection, check whether $(\mathrm{splus}(m), \mathrm{val}(\mathrm{splus}(m)))$ is a test and if so, then value m is deleted from the projection and $\mathrm{splus}(m)$ is stored in STGOOD if it corresponds to a good test at the current step.
4) If at least one value has been deleted from the projection, then the reduction of the projection is necessary. The reduction consists in checking, for each element t of the projection, whether $(\mathrm{obj}(t), t)$ is not a test (as a result of previous eliminating values) and if so, this element is deleted from the projection. If, under reduction, at least one element has been deleted, then Step 2, Step 3, Step 4, and Step 5 are repeated.
5) Check whether the subtask is over or not. The subtask is over when either the projection is empty or the intersection of all elements of the projection corresponds to a test (see, please, Step 1). If the subtask is not over, then an object (value) in this projection is selected and the new subtask is formed. The new subsets s_* and t_* are constructed and the basic algorithm runs recursively.

The set TGOOD of all the GMRTs is obtained as follows: $\mathrm{TGOOD} = \{\mathrm{tg} |\ \mathrm{tg} = (s, \mathrm{val}(s)),\ s \in \mathrm{STGOOD}\}$. Main recursive algorithm of `ASTRA-1` based on the decomposition of the main problem of inferring GMRTs into subtasks of the second kind is given in Fig. 1. It consists of `DelVal` (Fig. 2), `DelObj` (Fig. 3), `ChoiceOfSubTask` and `FormSubTask` (Fig. 4). Here, in pseudocode, $m \in M$ and $g \in G$ are denoted by M_i and $G(i)$, respectively. `ChoiceOfSubTask` generates M_{na}, which is the attribute value chosen by means of statistical investigations. In a simple case, it can be the value appearing in average.

Algorithm `GenAllGMRTs`
  Input: G, M
  Output: STGOOD `// to be modified`
  $flag := 1$;
  while *true* **do**
    while *flag=1* **do**
      `DelVal` ; `// modify` M
      if *flag=1* **then**
        `DelObj` ; `// modify` G_+
    if $G_+ = \emptyset$ **or** $M = \emptyset$ **or** $M' \not\subseteq G_+$ **or** $G_+ \subseteq s, \exists s \in \mathrm{STGOOD}$ **then**
      return ; `// exit`
    M_{SUB}, G_{SUB},na:=$\emptyset$;
    `ChoiceOfSubTask` ; `// form` na
    `FormSubTask` ; `// form` G_{SUB}, M_{SUB}
    `GenAllGMRTs(`G_{SUB}, M_{SUB}`)` ;
    `// recursion`
    $M := M \setminus M_{na}$;
    `DelObj`;

Fig. 1. Algorithm `GenAllGMRTs` of `ASTRA-1`

Algorithm `DelVal`
  Input: G_+, M
  Output: $M, flag$
  $flag$:= 0, i := 1;
  while $i \leq \|M\|$ **do**
    if $M'_i \subseteq s, \exists s \in \mathrm{STGOOD}$ **then**
      $M := M \setminus M_i$;
      $flag$:= 1;
    else if $(M'_i \cap G_+)'' \subseteq G_+$ **then**
    `//` $((M'_i \cap G_+), (M'_i \cap G_+)')$ `is a test`
      j :=1 ;
      while $j \leq \|\mathrm{STGOOD}\|$ **do**
        if $\mathrm{STGOOD}_j \subseteq (M'_i \cap G_+)$ **then**
          $\mathrm{STGOOD} := \mathrm{STGOOD} \setminus \mathrm{STGOOD}_j$
      $\mathrm{STGOOD} := \mathrm{STGOOD} \cup (M'_i \cap G_+)$;
      $M := M \setminus M_i$;
      $flag$:= 1;
    $i := i + +$;
  return;

Fig. 2. Algorithm `DelVal` of `GenAllGMRTs`

V. Extracting prior knowledge of GMRTs and decreasing the number of subcontexts

Algorithms of GMRTs inferring are constructed by the rules of selecting and ordering subcontexts of the main classification context. We need the following additional definitions.

Let t be a set of values such that $(\mathrm{obj}(t), t)$ is a test for G_+. We say that **the value** $m \in M, m \in t$ **is essential** in t if $(\mathrm{obj}(t \setminus m), (t \setminus m))$ is not a test for a given set of objects. Generally, we are interested in finding the maximal subset $\mathrm{sbmax}(t) \subset t$ such that $(\mathrm{obj}(t), t)$ is a test but $(\mathrm{obj}(\mathrm{sbmax}(t)), \mathrm{sbmax}(t))$ is not a test for a given set of positive objects. Then $\mathrm{sbmin}(t) = t \setminus \mathrm{sbmax}(t)$ is a minimal set of essential values in t.

Let $s \subseteq G_+$, assume also that $(s, \mathrm{val}(s))$ is not a test for G_+. **The object** with index $j \in s$ **is said to be an essential** in s if $(s \setminus j, \mathrm{val}(s \setminus j))$ proves to be a test for a given set of positive objects. Generally, we are also interested in finding the maximal subset $\mathrm{sbmax}(s) \subset s$ such that $(s, \mathrm{val}(s))$ is not a test but $(\mathrm{sbmax}(s), \mathrm{val}(\mathrm{sbmax}(s)))$ is a test for a given set of positive objects. Then $\mathrm{sbmin}(s) = s \setminus \mathrm{sbmax}(s)$ is the

Algorithm `DelObj`

Input: G_+, M

Output: $G_+, flag$

$flag$:= 0, i := 1;

while $i \leq \|G_+\|$ **do**

  if $G_+(i)'' \nsubseteq G_+$ **then** `// Is` $(G_+(i), G_+(i)')$ `not a test?`

    $G_+ := G_+ \backslash G_+(i)$;

    $flag$:= 1;

  $i := i + +$;

return;

Fig. 3. Algorithm `DelObj` of `GenAllGMRTs`

Algorithm `FormSubTask`

Input: G_+, M

Output: G_{SUB}, M_{SUB}

i := 1;

$G_{SUB} := M'_{na} \cap G_+$;

while $i \leq \|G_{SUB}\|$ **do**

  $M_{SUB} := M_{SUB} \cup ((G_{SUB}(i))' \cap M)$;

  $i := i + +$;

return;

Fig. 4. Algorithm `FormSubTask` of `GenAllGMRTs`

minimal set of essential objects in s.

Finding quasi-maximal (minimal) subsets of objects and values is the key procedures behind searching for initial content of STGOOD and determining the number of subtasks to be solved.

A. An Approach for Searching for Initial Content of STGOOD

In the beginning of inferring GMRTs, the set STGOOD is empty. We need the following procedure to obtain an initial content of it. This procedure extracts a quasi-maximal subset $s_* \subseteq G_+$ which is the extent of a test for G_+ (maybe not good).

We begin with the first index i_1 of s_*, then we take the next index i_2 of s_* and evaluate the function to_be_test$(\{i_1, i_2\}, \text{val}(\{i_1, i_2\}))$. If the value of the function is "true", then we take the next index i_3 of s_* and evaluate the function to_be_test$(\{i_1, i_2, i_3\}, \text{val}(\{i_1, i_2, i_3\}))$. If the value of the function to_be_test$(\{i_1, i_2\}, \text{val}(\{i_1, i_2\}))$ is "false", then the index i_2 of s_* is skipped and the function to_be_test$(\{i_1, i_3\}, \text{val}(\{i_1, i_3\}))$ is evaluated. We continue this process until we achieve the last index of s_*.

The complexity of this procedure is evaluated as the product of $||s_*||$ by the complexity of the function to_be_test(). To obtain the initial content of STGOOD, we use the set SPLUS $= \{\text{splus}(m) | m \in M\}$ and apply the procedure described above to each element of SPLUS.

To illustrate this procedure, we use the sets D_+ and D_- represented in Tab.II and III (our illustrative example). In these tables, $M = \{m_1, \ldots, m_{26}\}$ and subsets of values $\{m_8, m_9\}$, $\{m_{14}, m_{15}\}$ are denoted by m_* and m_+, respectively.

TABLE II
THE SET D_+ OF POSITIVE OBJECT DESCRIPTIONS

G	D_+
1	m_1 m_2 m_5 m_6 m_{21} m_{23} m_{24} m_{26}
2	m_4 m_7 m_8 m_9 m_{12} m_{14} m_{15} m_{22} m_{23} m_{24} m_{26}
3	m_3 m_4 m_7 m_{12} m_{13} m_{14} m_{15} m_{18} m_{19} m_{24} m_{26}
4	m_1 m_4 m_5 m_6 m_7 m_{12} m_{14} m_{15} m_{16} m_{20} m_{21} m_{24} m_{26}
5	m_2 m_6 m_{23} m_{24}
6	m_7 m_{20} m_{21} m_{26}
7	m_3 m_4 m_5 m_6 m_{12} m_{14} m_{15} m_{20} m_{22} m_{24} m_{26}
8	m_3 m_6 m_7 m_8 m_9 m_{13} m_{14} m_{15} m_{19} m_{20} m_{21} m_{22}
9	m_{16} m_{18} m_{19} m_{20} m_{21} m_{22} m_{26}
10	m_2 m_3 m_4 m_5 m_6 m_8 m_9 m_{13} m_{18} m_{20} m_{21} m_{26}
11	m_1 m_2 m_3 m_7 m_{19} m_{20} m_{21} m_{22} m_{26}
12	m_2 m_3 m_{16} m_{20} m_{21} m_{23} m_{24} m_{26}
13	m_1 m_4 m_{18} m_{19} m_{23} m_{26}
14	m_{23} m_{24} m_{26}

TABLE III
THE SET D_- OF NEGATIVE OBJECT DESCRIPTIONS

G	D_-
15	$m_3m_8m_{16}m_{23}m_{24}$
16	$m_7m_8m_9m_{16}m_{18}$
17	$m_1m_{21}m_{22}m_{24}m_{26}$
18	$m_1m_7m_8m_9m_{13}m_{16}$
19	$m_2m_6m_7m_9m_{21}m_{23}$
20	$m_{19}m_{20}m_{21}m_{22}m_{24}$
21	$m_1m_{20}m_{21}m_{22}m_{23}m_{24}$
22	$m_1m_3m_6m_7m_9m_{16}$
23	$m_2m_6m_8m_9m_{14}m_{15}m_{16}$
24	$m_1m_4m_5m_6m_7m_8m_{16}$
25	$m_7m_{13}m_{19}m_{20}m_{22}m_{26}$
26	$m_1m_2m_3m_5m_6m_7m_{16}$
27	$m_1m_2m_3m_5m_6m_{13}m_{18}$
28	$m_1m_3m_7m_{13}m_{19}m_{21}$
29	$m_1m_4m_5m_6m_7m_8m_{13}m_{16}$
30	$m_1m_2m_3m_6m_{12}m_{14}m_{15}m_{16}$
31	$m_1m_2m_5m_6m_{14}m_{15}m_{16}m_{26}$
32	$m_1m_2m_3m_7m_9m_{13}m_{18}$
33	$m_1m_5m_6m_8m_9m_{19}m_{20}m_{22}$
34	$m_2m_8m_9m_{18}m_{20}m_{21}m_{22}m_{23}m_{26}$
35	$m_1m_2m_4m_5m_6m_7m_9m_{13}m_{16}$
36	$m_1m_2m_6m_7m_8m_{13}m_{16}m_{18}$
37	$m_1m_2m_3m_4m_5m_6m_7m_{12}m_{14}m_{15}m_{16}$
38	$m_1m_2m_3m_4m_5m_6m_9m_{12}m_{13}m_{16}$
39	$m_1m_2m_3m_4m_5m_6m_{14}m_{15}m_{19}m_{20}m_{23}m_{26}$
40	$m_2m_3m_4m_5m_6m_7m_{12}m_{13}m_{14}m_{15}m_{16}$
41	$m_2m_3m_4m_5m_6m_7m_9m_{12}m_{13}m_{14}m_{15}m_{19}$
42	$m_1m_2m_3m_4m_5m_6m_{12}m_{16}m_{18}m_{19}m_{20}m_{21}m_{26}$
43	$m_4m_5m_6m_7m_8m_9m_{12}m_{13}m_{14}m_{15}m_{16}$
44	$m_3m_4m_5m_6m_8m_9m_{12}m_{13}m_{14}m_{15}m_{18}m_{19}$
45	$m_1m_2m_3m_4m_5m_6m_7m_8m_9m_{12}m_{13}m_{14}m_{15}$
46	$m_1m_3m_4m_5m_6m_7m_{12}m_{13}m_{14}m_{15}m_{16}m_{23}m_{24}$
47	$m_1m_2m_3m_4m_5m_6m_8m_9m_{12}m_{14}m_{16}m_{18}m_{22}$
48	$m_2m_8m_9m_{12}m_{14}m_{15}m_{16}$

The set SPLUS_0 for positive class of examples is in Tab.IV. The initial content of STGOOD_0 is {(2,10), (3, 10), (3, 8), (4, 12), (1, 4, 7), (1, 5,12), (2, 7, 8), (3, 7, 12), (1, 2, 12, 14), (2, 3, 4, 7), (4, 6, 8, 11)}.

Applying operation $\mathrm{generalization\_of}(s) = s'' = \mathrm{obj}(\mathrm{val}(s))$ to $\forall s \in \mathrm{STGOOD}_0$, we obtain $\mathrm{STGOOD}_1 =$ {(2,10), (3, 10), (3, 8), (4, 7, 12), (1, 4, 7), (1, 5,12), (2, 7, 8), (3, 7, 12), (1, 2, 12, 14), (2, 3, 4, 7), (4, 6, 8, 11)}.

By Th. 1, one can delete value m_{12} from consideration, see $\mathrm{splus}(m_{12})$ in Tab.IV. The initial content of STGOOD allows to decrease the number of using the procedure $\mathrm{to\_be\_test}()$ and the number of putting extents of tests into STGOOD. Apart from this, it helps to find essential objects in the sub-contexts (projections).

TABLE IV
THE SET SPLUS_0

$\mathrm{splus}(m), m \in M$
$\mathrm{splus}(m_*) \to \{2, 8, 10\}$
$\mathrm{splus}(m_{13}) \to \{3, 8, 10\}$
$\mathrm{splus}(m_{16}) \to \{4, 9, 12\}$
$\mathrm{splus}(m_1) \to \{1, 4, 11, 13\}$
$\mathrm{splus}(m_5) \to \{1, 4, 7, 10\}$
$\mathrm{splus}(m_{12}) \to \{2, 3, 4, 7\}$
$\mathrm{splus}(m_{18}) \to \{3, 9, 10, 13\}$
$\mathrm{splus}(m_2) \to \{1, 5, 10, 11, 12\}$
$\mathrm{splus}(m_+) \to \{2, 3, 4, 7, 8\}$
$\mathrm{splus}(m_{19}) \to \{3, 8, 9, 11, 13\}$
$\mathrm{splus}(m_*) \to \{2, 8, 10\}$
$\mathrm{splus}(m_{13}) \to \{3, 8, 10\}$
$\mathrm{splus}(m_{16}) \to \{4, 9, 12\}$
$\mathrm{splus}(m_1) \to \{1, 4, 11, 13\}$
$\mathrm{splus}(m_5) \to \{1, 4, 7, 10\}$
$\mathrm{splus}(m_{12}) \to \{2, 3, 4, 7\}$
$\mathrm{splus}(m_{18}) \to \{3, 9, 10, 13\}$
$\mathrm{splus}(m_2) \to \{1, 5, 10, 11, 12\}$
$\mathrm{splus}(m_+) \to \{2, 3, 4, 7, 8\}$
$\mathrm{splus}(m_{19}) \to \{3, 8, 9, 11, 13\}$
$\mathrm{splus}(m_{22}) \to \{2, 7, 8, 9, 11\}$
$\mathrm{splus}(m_{23}) \to \{1, 2, 5, 12, 13, 14\}$
$\mathrm{splus}(m_3) \to \{3, 7, 8, 10, 11, 12\}$
$\mathrm{splus}(m_4) \to \{2, 3, 4, 7, 10, 13\}$
$\mathrm{splus}(m_6) \to \{1, 4, 5, 7, 8, 10\}$
$\mathrm{splus}(m_7) \to \{2, 3, 4, 6, 8, 11\}$
$\mathrm{splus}(m_{24}) \to \{1, 2, 3, 4, 5, 7, 12, 14\}$
$\mathrm{splus}(m_{20}) \to \{4, 6, 7, 8, 9, 10, 11, 12\}$
$\mathrm{splus}(m_{21}) \to \{1, 4, 6, 8, 9, 10, 11, 12\}$
$\mathrm{splus}(m_{26}) \to \{1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 13, 14\}$

B. The number of subtasks to be solved

This number is determined by the number of essential values in the set M. The quasi-minimal subset of essential values in M can be found by a procedure analogous to the procedure applicable to search for the initial content of STGOOD.

We begin with the first value m_1 of M, then we take the next value m_2 of M and evaluate the function $\mathrm{to\_be\_test}(\mathrm{obj}(\{m_1, m_2\}), \{m_1, m_2\})$. If the value of the function is false, then we take the next value m_3 of M and evaluate the function $\mathrm{to\_be\_test}(\mathrm{obj}(\{m_1, m_2, m_3\}), \{m_1, m_2, m_3\})$. If the value of the function $\mathrm{to\_be\_test}(\mathrm{obj}(\{m_1, m_2\}), \{m_1, m_2\})$ is true, then value m_2 of M is skipped and the function $\mathrm{to\_be\_test}(\mathrm{obj}(\{m_1, m_3\}), \{m_1, m_3\})$ is evaluated. We continue this process until we achieve the last value of M.

As a result of the procedure, we have quasi-maximal subset $\mathrm{sbmax}(M)$, such that $(\mathrm{obj}(\mathrm{sbmax}(M)), \mathrm{sbmax}(M))$ is not a test for positive examples. Then subset $\mathrm{LEV} = M \setminus \mathrm{sbmax}(M)$ is quasi-minimal subset of essential values in M.

The complexity of this procedure is evaluated as the product of $||M||$ by the complexity of the function $\mathrm{to\_be\_test}()$. For our example (Tab.II,III), we have the following set (list of essential values) LEV: $\{m_{16}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{26}\}$.

Proposition 1: Each essential value is included at least in one positive object description.

Proof: Assume that for an object description $t_i, i \in G_+$, we have $t_i \cap \mathrm{LEV} = \emptyset$. Then $t_i \subseteq M \setminus \mathrm{LEV}$. But $M \setminus \mathrm{LEV}$ is included at least in one of negative object descriptions and, consequently, t_i also possesses of this property. But it contradicts to the fact that t_i is a description of positive object. ■

Proposition 2: Assume that $X \subseteq M$. If $X \cap \mathrm{LEV} = \emptyset$, then $\mathrm{to\_be\_test}((\mathrm{obj}(X), X) = \mathrm{false}$. This proposition is the consequence of Proposition 1.

For finding all GMRTs containing in a given main classification context, it is sufficient to solve this problem only for subcontexts associated with essential values.

Note that the description of $t_{14} = \{m_{23}, m_{24}, m_{26}\}$ is closed because of $\mathrm{obj}(\{m_{23}, m_{24}, m_{26}\} = \{1, 2, 12, 14\}$ and $\mathrm{val}(\{1, 2, 12, 14\} = \{m_{23}, m_{24}, m_{26}\}$. We also know that $s = \{1, 2, 12, 14\}$ is closed too (we obtained this result during generalization of elements of STGOOD. So $(\mathrm{obj}(\{m_{23}, m_{24}, m_{26}\}))$, $\{m_{23}, m_{24}, m_{26}\})$ is a maximally redundant test for positive objects and we can, consequently, delete t_{14} from consideration. As a result of deleting m_{12} and t_{14}, we have the modified set SPLUS (Tab.V).

TABLE V
THE SET SPLUS_1

$\mathrm{splus}(m), m \in M$
$\mathrm{splus}(m_*) \to \{2, 8, 10\}$
$\mathrm{splus}(m_{13}) \to \{3, 8, 10\}$
$\mathrm{splus}(m_{16}) \to \{4, 9, 12\}$
$\mathrm{splus}(m_1) \to \{1, 4, 11, 13\}$
$\mathrm{splus}(m_5) \to \{1, 4, 7, 10\}$
$\mathrm{splus}(m_{18}) \to \{3, 9, 10, 13\}$
$\mathrm{splus}(m_2) \to \{1, 5, 10, 11, 12\}$
$\mathrm{splus}(m_+) \to \{2, 3, 4, 7, 8\}$
$\mathrm{splus}(m_{19}) \to \{3, 8, 9, 11, 13\}$
$\mathrm{splus}(m_{22}) \to \{2, 7, 8, 9, 11\}$
$\mathrm{splus}(m_{23}) \to \{1, 2, 5, 12, 13\}$
$\mathrm{splus}(m_3) \to \{3, 7, 8, 10, 11, 12\}$
$\mathrm{splus}(m_4) \to \{2, 3, 4, 7, 10, 13\}$
$\mathrm{splus}(m_6) \to \{1, 4, 5, 7, 8, 10\}$
$\mathrm{splus}(m_7) \to \{2, 3, 4, 6, 8, 11\}$
$\mathrm{splus}(m_{24}) \to \{1, 2, 3, 4, 5, 7, 12\}$
$\mathrm{splus}(m_{20}) \to \{4, 6, 7, 8, 9, 10, 11, 12\}$
$\mathrm{splus}(m_{21}) \to \{1, 4, 6, 8, 9, 10, 11, 12\}$
$\mathrm{splus}(m_{26}) \to \{1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 13\}$

VI. Selecting and Ordering Subcontexts in inferring GMRTs

The main question is how we should approach the problem of selecting and ordering subtasks (subcontexts). Consider Tab.VI with auxiliary information.The columns of this table correspond to the essential values, the lines of it correspond to the objects of the main context in the beginning of finding GMRTs.

It is clear that if we shall obtain all the intents of GMRTs entering into descriptions of objects 1, 2, 3, 5, 7, 9, 10, 12 and 13, then the main task will be over because the remaining object descriptions (objects 4, 6, 8, 11) give, in their intersection, the intent of already known test (see, please, the initial content of STGOOD). Thus we have to consider only the subcontexts of essential values associated with object descriptions 1, 2, 3, 5, 7, 9, 10, 12, 13. The number of such subcontexts is 39. But this estimation is not realistic.

TABLE VI
Auxiliary information

No	m_{16}	m_{18}	m_{19}	m_{20}	m_{21}	m_{22}	m_{23}	m_{24}	m_{26}	$\sum m_{ij}$
1					×		×	×	×	4
2						×	×	×	×	4
3		×	×					×	×	4
5							×	×		2
7				×		×		×	×	4
9	×	×	×	×	×	×			×	7
10		×		×	×				×	4
12	×			×	×		×	×	×	4
13		×	×				×		×	4
4	×			×	×			×	×	
6				×	×				×	
8			×	×	×	×			×	
11			×	×	×	×			×	
$\sum d_i$	2	4	3	4	4	3	5	6	8	39

We begin with ordering the set of indices of objects by increasing the number of their entering into elements of STGOOD_1, see Tab.VII.

TABLE VII
Ordering the set of object indices in STGOOD_1

Index of object	**9**	**13**	**5**	**10**	**1**
The number of entering in elements of STGOOD_1	0	0	1	2	3
Index of object	**2**	**3**	**12**	**7**	
The number of entering in elements of STGOOD_1	4	4	4	5	

Now we select the subcontexts (subtasks), based on $\mathrm{proj}(t \times m)$, where t is the object description whose index enters into smallest number of elements of STGOOD and m is an essential value in t, entering in the smallest number of object descriptions in $\mathrm{proj}(t)$ (the objects $4, 6, 8, 11$ are not considered).

After solving each subtask, we have to correct the sets SPLUS, STGOOD, and auxiliary information. So, the first sub-task is $t_9 \times m_{16}$. Solving this sub-task, we have not any new test, but we can delete m_{16} from t_9 and then we solve the sub-task $t_9 \times m_{19}$. As a result, we introduce $s = \{9, 11\}$ in STGOOD and delete t_9 from consideration because of m_{16}, m_{19} are the only essential values in this object description (because each object description has several individual minimal subsets of essential values). .

Then we solve subtasks $t_{13} \times m_{19}$ and $t_{13} \times m_{18}$. The result is introducing $s = \{13\}$ in STGOOD and deleting t_{13} because m_{18} is the only essential value in this object description. After deleting t_9, t_{13}, we can modify SPLUS and delete from it $\mathrm{splus}(m_{16}) = \{4, 12\}$ and $\mathrm{splus}(m_{18}) = \{3, 10\}$. This means that we delete from consideration values m_{16}, m_{18}. Tabs VIII and IX contain the modified SPLUS and Auxiliary information, respectively.

TABLE VIII
The set SPLUS_2

$\mathrm{splus}(m), m \in M$
$\mathrm{splus}(m_*) \to \{2, 8, 10\}$
$\mathrm{splus}(m_{13}) \to \{3, 8, 10\}$
$\mathrm{splus}(m_1) \to \{1, 4, 11\}$
$\mathrm{splus}(m_5) \to \{1, 4, 7, 10\}$
$\mathrm{splus}(m_2) \to \{1, 5, 10, 11, 12\}$
$\mathrm{splus}(m_+) \to \{2, 3, 4, 7, 8\}$
$\mathrm{splus}(m_{19}) \to \{3, 8, 11\}$
$\mathrm{splus}(m_{22}) \to \{2, 7, 8, 11\}$
$\mathrm{splus}(m_{23}) \to \{1, 2, 5, 12\}$
$\mathrm{splus}(m_3) \to \{3, 7, 8, 10, 11, 12\}$
$\mathrm{splus}(m_4) \to \{2, 3, 4, 7, 10\}$
$\mathrm{splus}(m_6) \to \{1, 4, 5, 7, 8, 10\}$
$\mathrm{splus}(m_7) \to \{2, 3, 4, 6, 8, 11\}$
$\mathrm{splus}(m_{24}) \to \{1, 2, 3, 4, 5, 7, 12\}$
$\mathrm{splus}(m_{20}) \to \{4, 6, 7, 8, 10, 11, 12\}$
$\mathrm{splus}(m_{21}) \to \{1, 4, 6, 8, 10, 11, 12\}$
$\mathrm{splus}(m_{26}) \to \{1, 2, 3, 4, 6, 7, 10, 11, 12\}$

TABLE IX
Auxiliary information (2)

No	m_{19}	m_{20}	m_{21}	m_{22}	m_{23}	m_{24}	m_{26}	$\sum m_{ij}$
1			×		×	×	×	4
2				×	×	×	×	4
3	×					×	×	4
5					×	×		2
7		×		×		×	×	4
10		×	×				×	4
12		×	×		×	×	×	4
4		×	×			×	×	
6		×	×				×	
8	×	×	×	×			×	
11	×	×	×	×			×	
$\sum d_i$	1	3	3	2	4	6	6	25

Tab. X illustrates all modeling process of inferring all GMRTs, some part of which is omitted.

Tab.XI shows the final sets STGOOD and TGOOD. There are various ways of using the proposed decomposition and

TABLE X
MODELING PROCESSES OF INFERRING GMRTS BY SEQUENCE OF SUBTASKS

N	Subcontext	Extent of New Test	Deleted values	Deleted objects
1	$t_9 \times m_{16}$			
2	$t_9 \times m_{19}$	$(9, 11)$		t_9
3	$t_{13} \times m_{18}$			
4	$t_{13} \times m_{19}$	(13)	m_{16}, m_{18}	t_{13}
5	$t_5 \times m_{23}$		m_{23}	
6	$t_5 \times m_{24}$			t_5
7	$t_{10} \times m_{20}$	$(8, 10)$		
8	$t_{10} \times m_{21}$			
9	$t_{10} \times m_{26}$		m_*, m_{13}, m_4, m_5	t_{10}
10	$t_1 \times m_{21}$			
11	$t_1 \times m_{24}$		m_1, m_2	t_1
12	$t_2 \times m_{22}$	$(7, 8, 11)$	m_{22}	
13	$t_2 \times m_{22}$			
14	$t_2 \times m_{24}$			t_2
15	$t_3 \times m_{19}$	$(3, 11)$	m_{19}	
16	$t_3 \times m_{24}$		m_{24}	t_{12}, t_7
17	$t_3 \times m_{26}$			t_3

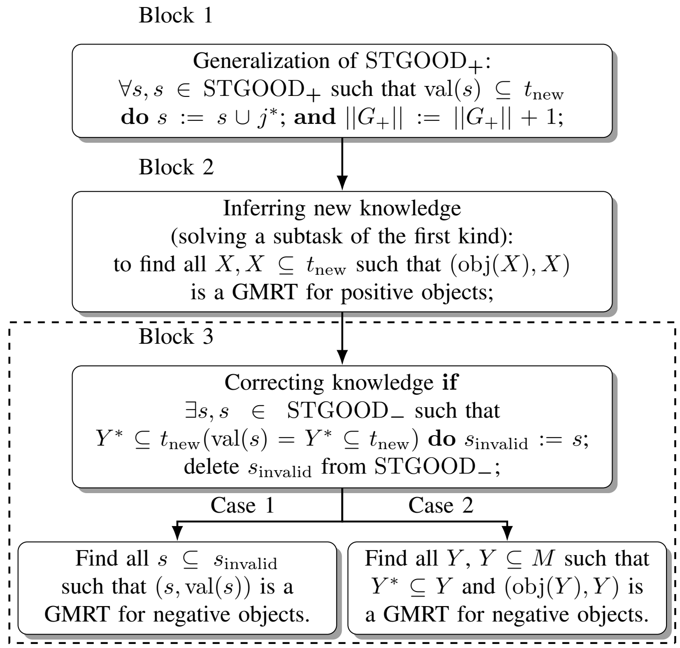

Fig. 5. Block-scheme of Incremental inferring GMRTs

extracted initial information about GMRTs [6], but we confine ourselves to the most important ideas which allow developing incremental processes of inferring GMRTs contained in a given classification context.

TABLE XI
THE SETS STGOOD AND TGOOD

N	STGOOD	TGOOD
1	13	$m_1 m_4 m_{18} m_{19} m_{23} m_{26}$
2	2,10	$m_4 m_* m_{26}$
3	3,10	$m_3 m_4 m_{13} m_{18} m_{26}$
4	8,10	$m_3 m_6 m_* m_{13} m_{20} m_{21}$
5	9,11	$m_{19} m_{20} m_{21} m_{22} m_{26}$
6	3,11	$m_3 m_7 m_{19} m_{26}$
7	3,8	$m_3 m_7 m_{13} m_+ m_{19}$
8	1,4,7	$m_5 m_6 m_{24} m_{26}$
9	2,7,8	$m_+ m_{22}$
10	1,5,12	$m_2 m_{23} m_{24}$
11	4,7,12	$m_{20} m_{24} m_{26}$
12	3,7,12	$m_3 m_{24} m_{26}$
13	7,8,11	$m_3 m_{20} m_{22}$
14	2,3,4,7	$m_4 m_{12} m_+ m_{24} m_{26}$
15	4,6,8,11	$m_7 m_{20} m_{21}$
16	1,2,12,14	$m_{23} m_{24} m_{26}$

VII. APPROACH TO INCREMENTAL INFERRING GMRTS

Incremental supervised learning is necessary when a new portion of observations becomes available over time. Suppose that each new object comes with the indication of its class membership. The following actions are necessary with arrival of a new object: 1) checking whether it is possible to perform generalization of some existing rules (tests) for the class to which a new object belongs (a class of positive objects, for certainty); 2) inferring all GMRTs induced by the new object description; 3) checking the validity of rules (tests) for negative objects, and, if it is necessary, modifying the tests that are invalid (test for negative objects is invalid if its intent is included in a new (positive) object description). Thus the following mental acts are performed:

- Pattern recognition and generalization of knowledge (increasing the power of already existing inductive knowledge);
- Increasing knowledge (inferring new knowledge);
- Correcting knowledge (diagnostic reasoning).

The first act modifies already existing tests (rules). The second act is reduced to subtask of the first kind. The third act can be implemented by the following ways. In the first way, we delete invalid tests (rules) and, by the use of subtask of the first kind, we must find new GMRTs generated by negative objects descriptions that have been covered by invalid tests. In the second way, this act can be reduced to subtask of the second kind.

An idea of an incremental approach to inferring GMRTs is described in Fig. 5, where STGOOD$_+$ and STGOOD$_-$ are sets of extents of GMRTs for G_+ and G_-, respectively. Case 1 and Case 2 are the subtasks of the first and second kind, respectively.

VIII. CONCLUSION

In the paper, the decomposition of inferring good classification tests into subtasks of the first and second kinds is presented. The decomposition allows, in principle, to transform the process of inferring good tests into a “step by step” reasoning process. A block-scheme of this incremental process is considered. The rules of forming and reducing sub-contexts are given, in this paper. Some possibilities of constructing algorithms for GMRTs inferring with the use of both attributive and object subcontexts are considered depending on the nature of GMRTs’ features.

References

[1] X. A. Naidenova and J. G. Polegaeva, “An Algorithm of Finding the Best Diagnostic Tests,” in *The 4-th All Union Conference ”Application of Mathematical Logic Methods”*, G. Mintz and E. Lorents, Eds., 1986, pp. 87 – 92, (in Russian).

[2] H. Sug, “Discovery of multidimensional association rules focusing on instances in specific class,” *International Journal of Mathematics and Computers in Simulation*, vol. 5, pp. 250–256, 2011.

[3] J. Illes and I. Vajk, “Performance evaluation of apriori algorithm on a hadoop cluster,” in *Latest Trends in Circuits, Systems, Signal Processing and Automatic Control : Proc. of the 4th. Int. Conf. on Circuits, Systems, Control, Signals (CSCS’14), Proc. of the 2nd Int. Conf. on Acoustics, Speech and Audio Processing (ASAP ’14)*, F. Neri and C. Guarnaccia, Eds. WSEAS, 2014, pp. 114–121.

[4] M. B. Branco and N. Franco, “Study of algorithms for decomposition of a numerical semigroup,” *International Journal of Mathematics and Computers in Simulation*, vol. 1, pp. 106–110, 2007.

[5] B. Ganter and R. Wille, *Formal concept analysis: mathematical foundations*. Springer, Berlin, 1999.

[6] X. Naidenova and V. Parkhomenko, “Methods to choosing subcontexts in good maximally redundant tests inferring,” in *Recent advances in mathematical methods in applied sciences: Proc. of the 2014 Int. Conf. on Mathematical Models and Methods in Applied Sciences (MMAS ’14), Proc. of the 2014 Int. Conf. on Economics and Applied Statistics (EAS ’14)*, Y. Senichenkov and et al., Eds. WSEAS, 2014, pp. 230–237.

[7] X. Naidenova and A. Ermakov, “The decomposition of good diagnostic test inferring algorithms,” in *”Computer-Aided Design of Discrete Devices” (CAD DD2001), Proceedings of the 4-th Inter. Conf.*, J. Alty, L. Mikulich, and A. Zakrevskij, Eds., Minsk, 2001, vol. 3, pp. 61 – 68.

[8] B. Ganter and S. Kuznetsov, “Pattern structures and their projections,” in *Conceptual Structures: Broadening the Base*, ser. Lecture Notes in Computer Science, H. Delugach and G. Stumme, Eds. Springer Berlin Heidelberg, 2001, vol. 2120, pp. 129–142.

[9] O. Ore, “Galois connections,” *Trans. Amer. Math. Soc*, vol. 55, pp. 494–513, 1944.

[10] X. Naidenova, “The data-knowledge transformation,” *Text Procesing and Cognitive Technologies*, vol. 3, pp. 130–151, 1999.

[11] S. Ferré and O. Ridoux, “The use of associative concepts in the incremental building of a logical context,” in *ICCS*, ser. Lecture Notes in Computer Science, U. Priss, D. Corbett, and G. Angelova, Eds., vol. 2393. Springer, 2002, pp. 299–313.

[12] X. A. Naidenova, M. V. Plaksin, and V. L. Shagalov, “Inductive Inferring All Good Classification Tests,” in *”Knowledge-Dialog-Solution”, Proceedings of International Conference*, J. Valkman, Ed., vol. 1. Kiev, Ukraine: Kiev Institute of Applied Informatics, 1995, pp. 79 – 84.

[13] X. A. Naidenova, “DIAGARA: An Incremental Algorithm for Inferring Implicative Rules from Examples,” *Inf. Theories and Application*, vol. 12 - 2, pp. 171 – 196, 2005.

[14] E. Sperner, “Ein satz uber untermengen einer endlichen menge,” *Mathematische Zeitschrift*, vol. 27, no. 1, pp. 544–548, 1928.