Attributive and Object Subcontexts in Inferring Good Maximally Redundant Tests

Xenia Naidenova[1] and Vladimir Parkhomenko[2]

[1] Military Medical Academy, Saint-Petersburg, Russia
ksennaid@gmail.com
[2] St. Petersburg State Polytechnical University, Saint-Petersburg, Russia
parhomenko.v@gmail.com

Abstract. Inferring Good Maximally Redundant Classification Tests (GMRTs) as Formal Concepts is considered. Two kinds of classification subcontexts are defined: attributive and object ones. The rules of forming and reducing subcontexts based on the notion of essential attributes and objects are given. They lead to the possibility of the inferring control. In particular, an improved Algorithm for Searching all GMRTs on the basis of attributive subtask is proposed. The hybrid attributive and object approaches are presented. Some computational aspects of algorithms are analyzed.

Keywords: good classification test, Galois lattice, essential attributes and objects, implications, subcontexts

1 Introduction

Good Test Analysis (GTA) deals with the formation of the best descriptions of a given object class (class of positive objects) against the objects which do not belong to this class (class of negative objects) on the basis of lattice theory. We assume that objects are described in terms of values of a given set U of attributes, see an example in Tab.1. The key notion of GTA is the notion of classification. To give a target classification of objects, we use an additional attribute KL $\notin U$. A target attribute partitions a given set of objects into disjoint classes the number of which is equal to the number of values of this attribute. In Tab.1, we have two classes: the objects in whose descriptions the target value k appears and all the other objects.

Denote by M the set of attribute values such that $M = \{\cup \text{dom(attr)}, \text{attr} \in U\}$, where dom(attr) is the set of all values of attr, i.e. a plain scaling in terms of [3]. Let $G = G_+ \cup G_-$ be the set of objects, where G_+ and G_- are the sets of positive and negative objects respectively. Let $P(B), B \subseteq M$, be the set of all the objects in whose descriptions B appears. $P(B)$ is called the interpretation of B in the power set 2^G. If $P(B)$ contains only G_+ objects and the number of these objects is more than 2, then B is called a description of some positive objects or a diagnostic (classification) **test** for G_+ [1]. The words diagnostic (classification) can be omitted in the paper.

© Karell Bertet, Sebastian Rudolph (Eds.): CLA 2014, pp. 181–193,
ISBN 978–80–8152–159–1, Institute of Computer Science, Pavol Jozef Šafárik University in Košice, 2014.

Table 1. Motivating Example of classification

No	Height	Color of Hair	Color of Eyes	KL
1	Low	Blond	Blue	$k(+)$
2	Low	Brown	Blue	$k(-)$
3	Tall	Brown	Hazel	$k(-)$
4	Tall	Blond	Hazel	$k(-)$
5	Tall	Brown	Blue	$k(-)$
6	Low	Blond	Hazel	$k(-)$
7	Tall	Red	Blue	$k(+)$
8	Tall	Blond	Blue	$k(+)$

Let us recall the definition of a good test or good description for a subset of G_+ (via partitions of objects). A subset $B \subseteq M$ of attribute values is a **good test** for a subset of positive objects if it is a test and no such subset $C \subseteq M$ exists, so that $P(B) \subset P(C) \subseteq G_+$ [7].

Sec.2 is devoted to defining a concept of good diagnostic (classification) test as a formal concept. Sec.3 gives the decomposition of good tests inferring based on two kinds of subcontexts of the initial classification context. Sec.4 is devoted to an analysis of algorithms based on using subcontexts including the evaluation of the number of sub-problems to be solved, the depth of recursion, the structure of sub-problems and their ordering, and some others.

2 Good Maximally Redundant Tests as Formal Concepts

Assume that $G = \overline{1, N}$ is the set of objects indices (objects, for short) and $M = \{m_1, m_2, \ldots, m_j, \ldots m_m\}$ is the set of attributes values (values, for short). Each object is described by a set of values from M. The object descriptions are represented by rows of a table whose columns are associated with the attributes taking their values in M.

Let $A \subseteq G$, $B \subseteq M$. 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: for $A \subseteq G$ and $B \subseteq M$, $A' = \text{val}(A) = \{\text{intersection of all } B_i | \; B_i \subseteq M, i \in A\}$ and $B' = \text{obj}(B) = \{i | \; i \in G, B \subseteq B_i\}$. Of course, we have $\text{obj}(B) = \{\text{intersection of all obj}(m) | \; \text{obj}(m) \subseteq G, m \in B\}$.

There are two closure operators [9]: $\text{generalization\_of}(B) = B'' = \text{val}(\text{obj}(B))$ and $\text{generalization\_of}(A) = A'' = \text{obj}(\text{val}(A))$. A set A is closed if $A = \text{obj}(\text{val}(A))$. A set B is closed if $B = \text{val}(\text{obj}(B))$. For $g \in G$ and $m \in M$, $\{g\}'$ is denoted by g' and called object intent, and $\{m\}'$ is denoted by m' and called value extent. Let us recall the main definitions of GTA [7].

A **Diagnostic Test** (DT) for the positive examples G_+ is a pair (A, B) such that $B \subseteq M$, $A = B' \neq \emptyset$, $A \subseteq G_+$, $B \not\subseteq g' \; \forall g \in G_-$. A diagnostic test (A, B)

for G_+ is **maximally redundant** if $obj(B \cup m) \subset A$ for all $m \notin B$ and $m \in M$. 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_+.

In the paper, we deal with Good Maximally Redundant Tests (GMRTs). If a good test (A, B) for G_+ is maximally redundant, then any extension $B_* = B \cup m$, $m \notin B$, $m \in M$ implies that $(\text{obj}(B_*), B_*)$ is not a good test for G_+. Any object description d of $g \in G$ in a given classification context is a maximally redundant set of values because $\forall m \notin d$, $m \in M, \text{obj}(d \cup m)$ is equal to $\emptyset$. GMRT can be regarded as a special type of hypothesis [4]

In Tab.1, $((1, 8)$, Blond Blue) is a GMRT for $k(+)$, $((4, 6)$, Blond Hazel) is a DT for $k(-)$ but not a good one, and $((3, 4, 6)$, Hazel) is a GMRT for $k(-)$.

3 The Decomposition of Inferring GMRTs into Subtasks

There are two possible kinds of subtasks of inferring GMRTs for a set G_+ [8]:

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

For solving these subtasks we need only form subcontexts of a given classification context. The first subtask is useful to find all GMRTs whose intents are contained in the description d of an object g. This subtask is considered in [2] for fast incremental concept formation, where the definition of subcontexts is given.

We introduce the **projection of a positive object description** d on the set D_+, i.e. descriptions of all positive objects. The $\text{proj}(d)$ is $Z = \{z | \ z = d \cap d_* \neq \emptyset, d_* \in D_+$ and $(\text{obj}(z), z)$ is a test for $G_+\}$.

We also introduce a concept of **value projection** $\text{proj}(m)$ of a given value m on a given set D_+. The value projection is $\text{proj}(m) = \{d | \ m$ appears in d, $d \in D_+\}$.

Algorithm Algorithm for Searching all GMRTs on the basis of attributive subtask (ASTRA), based on value projections, was advanced in [6]. Algoritm DIAGaRa, based on object projections, was proposed in [5]. In what follows, we are interested in using both kinds of subcontexts for inferring all GMRTs for a positive (or negative) class of objects. The following theorem gives the foundation of reducing subcontexts [6].

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

Consider some example of reducing subcontext (see Tab.1). Let $\text{splus}(m)$ be $\text{obj}(m) \cap G_+$ or $\text{obj}(m) \cap G_-$ and SPLUS be $\{\text{splus}(m) | \ m \in M\}$. In Tab.1, we have $\text{SPLUS} = \text{obj}(m) \cap G_- = \{\{3, 4, 6\}, \{2, 3, 5\}, \{3, 4, 5\}, \{2, 5\}, \{4, 6\}, \{2, 6\}\}$ for values “Hazel, Brown, Tall, Blue, Blond, and Low” respectively.

We have val(obj(Hazel)) = Hazel, hence ((3, 4, 6), Hazel) is a DT for G_-. Then value "Blond" can be deleted from consideration, because splus(Blond) $\subset$ splus(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 SPLUS = $\{\{2, 3, 5\}, \{3, 5\}, \{2, 5\}, \{2\}\}$. Now we have val(obj(Brown)) = Brown and ((2, 3, 5), Brown) is a test for G_-. All values are deleted and all GMRTs for G_- have been obtained.

The initial information for finding all the GMRTs contained in a positive object description is the projection of it on the current set D_+. 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 whose descriptions produce the projection. In the projection, $\text{splus}(m) = \text{obj}(m) \cap s_*$, $m \in t_*$.

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

1. Check whether $(s_*, \text{val}(s_*)$ 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 go to the next step.
2. The value m can be deleted from the projection if $\text{splus}(m) \subseteq s$ for some $s \in$ STGOOD.
3. For each value m in the projection, check whether $(\text{splus}(m), \text{val}(\text{splus}(m))$ is a test and if so, then value m is deleted from the projection and 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 $(\text{obj}(t), t)$ 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, and Step 4 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 the choice of 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 algorithm of forming STGOOD is based on topological sorting of partially ordered sets. The set TGOOD of all the GMRTs is obtained as follows: TGOOD = $\{\text{tg} \mid \text{tg} = (s, \text{val}(s)),\ s \in \text{STGOOD}\}$.

4 Selecting and Ordering Subcontexts and Inferring GMRTs

Algorithms for inferring GMRTs are constructed by the rules of selecting and ordering subcontexts of the main classification context. Before entering into the details, let us recall some extra definitions. Let t be a set of values such that $(\text{obj}(t), t)$ is a test for G_+. We say that **the value** $m \in M, m \in t$ **is essential** in t if $(\text{obj}(t \setminus m), (t \setminus m))$ is not a test for a given set of objects. Generally, we are interested in finding the maximal subset $\text{sbmax}(t) \subset t$ such that $(\text{obj}(t), t)$ is a test but $(\text{obj}(\text{sbmax}(t)), \text{sbmax}(t))$ is not a test for a given set of positive objects. Then $\text{sbmin}(t) = t \setminus \text{sbmax}(t)$ is a minimal set of essential values in t. Let $s \subseteq G_+$, assume also that $(s, \text{val}(s))$ is not a test.

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

An Approach for Searching for Initial Content of STGOOD. In the beginning of inferring GMRTs, the set STGOOD is empty. Next we describe the 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 $\text{to\_be\_test}(\{i_1, i_2\}, \text{val}(\{i_1, i_2\}))$. If the value of the function is true, then we take the next index i_3 of s_* and evaluate the function $\text{to\_be\_test}(\{i_1, i_2, i_3\}, \text{val}(\{i_1, i_2, i_3\}))$. If the value of the function is false, then the index i_2 of s_* is skipped and the function $\text{to\_be\_test}(\{i_1, i_3\}, \text{val}(\{i_1, i_3\})))$ is evaluated. We continue this process until we achieve the last index of s_*.

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

The idea of using subcontexts in inferring GMRTs, described in Sec.3, can be presented in a pseudo-code form, see Fig.1. It presents a modification of ASTRA. DIAGARA and a hybrid approach can be easily formalized by the same way. The example below describes two general hybrid methods.

The initial part of GenAllGMRTs() is well discussed above. The abbreviation LEV stands for the List (set) of Essential Values. The function $\text{DelObj}(M, G_+)$ returns modified G and $flag$. The variable $flag$ is necessary for switching attributive subtasks. The novelty of ASTRA-2 is mainly based on using LEV. There is the new function ChoiceOfSubtask(). It returns $na := \text{LEV}_j$ with the maximal $2^{splus(\text{LEV}_j)}$. MainContext, defined $\text{FormSubTask}(na, M, G_+)$, consists of object descriptions. There is the auxiliary function $\text{kt}(m) = true$ if $(m' \in G_- = false)$ and $false$ otherwise.

To illustrate this procedure, we use the sets D_+ and D_- represented in Tab.2 and 3 (our illustrative example). In these tables, $M = \{m_1, \ldots, m_{26}\}$. The set SPLUS_0 for positive class of examples is in Tab.4. The initial content of

```
Algorithm GenAllGMRTs()
    Input: G, M
    Output: STGOOD
    begin
        Forming STGOOD ;
        Forming and Ordering LEV ;
        flag:=1;
    end
    while true do
        while flag=1 do
            M, flag DelVal(M, G+);
            if flag=1 then
                return;
            end
            G+, flag
            DelObj(M, G+);
        end
        if M' ⊆ G− or
        G+ ⊆ STGOOD then
            return STGOOD;
        end
        M_SUB:=∅;
        G_SUB:=∅;
        ChoiceOfSubtask;
        M_SUB, G_SUB
        FormSubTask(na, M, G+);
        GenAllGMRTs();
        M:=M\M_na;
        G+, flag DelObj(M, G+);
    end
```

(a) GenAllGMRTs

```
Algorithm DelVal()
    i := 1;
    flag := 0;
    while i ≤ 2^M do
        if M'_i ⊆ G+ then
            M := M\M_i;
            flag := 1;
        end
        else if kt(M'_i ∩ G+) then
            j :=1 ;
            while j ≤ 2^STGOOD do
                if STGOOD_j ⊆
                M'_i ∩ G+ then
                    STGOOD :=
                    STGOOD\
                    STGOOD_j
                end
            end
            STGOOD :=
            STGOOD ∪ M'_i ∩ G+;
            M := M\M_i;
            flag := 1;
        return;
    end
```

(b) DelVal

```
Algorithm DelObj()
    i := 1;
    flag := 0;
    while i ≤ 2^(G+) do
        if  G+(i) ⊆ M\LEV then
            G+ := G+\G+(i);
            flag := 1;
        end
    end
    return;
```

(c) DelObj

```
Algorithm FormSubTask()
    i := 1;
    G_SUB := M'_na ∩ G+;
    while i ≤ 2^(G_SUB) do
        M_SUB := M_SUB ∪
        (MainContext(G_SUB(i)∩M));
    end
    return;
```

(d) FormSubTask

Fig. 1. Algorithms of ASTRA-2

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)}.

Table 2. 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}

In these tables we denote subsets of values $\{m_8, m_9\}$, $\{m_{14}, m_{15}\}$ by m_a and m_b, respectively. Applying operation generalization_of$(s) = s'' = \text{obj}(\text{val}(s))$ to $\forall s \in$ STGOOD, we obtain 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, we can delete value m_{12} from consideration, see splus(m_{12}) in Tab.4. The initial content of STGOOD allows to decrease the number of using the procedure to_be_test() and the number of putting extents of tests into STGOOD.

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 to_be_test(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 to_be_test(obj($\{m_1, m_2, m_3\}$), $\{m_1, m_2, m_3\}$). If the value of the function is true, then value m_2 of M is skipped and the function to_be_test(obj($\{m1, m3\}$), $\{m1, m3\}$) is evaluated. We continue this process until we achieve the last value of M. The complexity of this procedure is evaluated as the production of $||M||$ by the complexity of the function to_be_test(). In Tab.2,3 we have the following LEV: $\{m_{16}, m_{18}, m_{19}, m_{20}, m_{21}, m_{22}, m_{23}, m_{24}, m_{26}\}$.

Table 3. The set D_- of negative object descriptions

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

Table 4. The set SPLUS_0

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

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 $t_i \cap LEV = \emptyset$. Then $t_i \subseteq M \backslash \text{LEV}$. But $M \backslash \text{LEV}$ is included at least in one of the negative object descriptions and, consequently, t_i also possesses this property. But it contradicts to the fact that t_i is a description of a positive object. □

Proposition 2. *Assume that* $X \subseteq M$. *If* $X \cap \text{LEV} = \emptyset$, *then* $\text{to\_be\_test}(X) =$ *false.*

Proposition 2 is the consequence of Proposition 1.

Note that the description of $t_{14} = \{m_{23}, m_{24}, m_{26}\}$ is closed because of $\text{obj}\{m_{23}, m_{24}, m_{26}\} = \{1, 2, 12, 14\}$ and $\text{val}\{1, 2, 12, 14\} = \{m_{23}, m_{24}, m_{26}\}$. We also know that $s = \{1, 2, 12, 14\}$ is closed too (we obtained this result during generalization of elements of STGOOD. So $(\text{obj}(\{m_{23}, m_{24}, m_{26}\}), \{m_{23}, m_{24}, m_{26}\})$ 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.5).

Table 5. The set SPLUS_1

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

The main question is how we should approach the problem of selecting and ordering subtasks (subcontexts). Consider Tab.6 with auxiliary information. It is clear that if we shall have all the intents of GMRTs entering into descriptions of objects 1, 2, 3, 5, 7, 9, 10, 12, 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 an 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 6. Auxiliary information

index of object	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 index of objects by the number of their entering in tests in $STGOOD_1$, see Tab.7.

Table 7. Ordering index of objects in $STGOOD_1$

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

Then we continue with object descriptions t_9 and t_{13}. Now we should select the subcontexts (subtasks), based on $\text{proj}(t \times m)$, where t is object description containing the smallest number of essential values and m is an essential value in t, entering in the smallest number of object descriptions. 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.

In the example (**method 1**), we have the following subtasks (Tab. 8).

Tab.10 shows the sets STGOOD and TGOOD. All subtasks did not require a recursion. A simpler method of ordering contexts is based on the basic recursive procedure for solving any kind of subtask described in the previous section. At

Table 8. The sequence of subtasks (method 1)

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_a, 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

each level of recursion, we can select the value entering into the greatest number of object descriptions; the object descriptions not containing this value generate the contexts to find GMRTs whose intents are included in them. For our example, value m_{26} does not cover two object descriptions: t_5 and t_8. The initial context is associated with m_{26}. The sequence of subtasks in the basic recursive procedure is in Tab.9 (**method 2**). We assume, in this example, that the GMRT intent of which is equal to t_{14} has been already obtained.

We consider only two possible ways of GMRTs construction based on decomposing the main classification context into subcontexts and ordering them by the use of essential values and objects. It is possible to use the two sets $\text{QT} = \{\{i,j\} \subseteq G_+ | \ (\{i,j\}, \text{val}(\{i,j\})$ is a test for $G_+\}$ and $\text{QAT} = \{\{i,j\} \subseteq G_+ | (\{i,j\}, \text{val}(\{i,j\})$ is not a test for $G_+\}$ for forming subcontexts and their ordering in the form of a tree structure.

5 Conclusion

In this paper, the decomposition of inferring good classification tests into subtasks of the first and second kinds is presented. This decomposition allows, in principle, to transform the process of inferring good tests into a step by step reasoning process.

The rules of forming and reducing subcontexts are given, in this paper. Various possibilities of constructing algorithms for inferring GMRTs with the use of both subcontexts are considered depending on the nature of GMRTs features.

Table 9. The sequence of subtasks (method 2)

N	Context, associated with	Extents of tests obtained	Values deleted from context	Object descriptions deleted from context
1	m_{26}	$(2,10),(3,10),$ $(2,3,4,7),(1,4,7)$	$m_a, m_{13}, m_b,$ m_5, m_6	t_{10}
2	m_{26}, m_{24}	$(3,7,12),$ $(4,7,12)$	$m_3, m_{20}, m_{23}, m_1,$ $m_2, m_4, m_7, m_{16},$ m_{18}, m_{19}, m_{22}	
	Subtask is over; return to the previous context and delete m_{24}			
3	m_{26}, not m_{24}, m_{23}	(13)	$m_3, m_7, m_{16}, m_{18},$ m_{19}, m_{20}, m_{22}	
	Subtask is over; return to the previous context, delete m_{23}			
4	m_{26}, not m_{24}, not m_{23}		$m_2, m_3, m_4, m_{16},$ $m_{18} m_{19}, m_{21}$	
5	m_{26}, m_{22}, not m_{24}, not m_{23}	(9,11), (7,11)		t_2, t_7
	Subtask is over; return to the previous context and delete m_{22}			
6	m_{26}, not m_{24}, not m_{23}, not m_{22}	(3,11), (4,6,11)	$m_2, m_3, m_4, m_{16},$ m_{18}, m_{19}	t_7, t_9, t_2, t_3
	Subtask is over; we have obtained all GMRTs whose intents contain m_{26}			
7	Context t_5	(1,5,12)		t_5
	Subtask is over; we have found all GMRTs whose intents are contained in t_5 .			
8	Context $t_8 \times m_{22}$	(7,8,11), (2,7,8)	$m_3, m_{20}, m_b, m_6,$ $m_a, m_{13}, m_{19}, m_{21}$	
	Subtask is over; return to the previous context and delete m_{22}			
9	Context t_8 without m_{22}	(8,10)	m_a	t_2, t_7
10	Context $t_8 \times m_{21}$ without m_{22}	(4,6,8,11)	m_7, m_{13}, m_{19}	t_6, t_{10}, t_{11}
	Subtask is over; return to the previous context and delete m_{21}, m_{20}			
11	Context t_8 without m_{22}, m_{21}, m_{20}	(3, 8)		t_4, t_6, t_{10}, t_{11}
	Subtask is over; we have found all GMRTs whose intents are contained in t_8.			

Table 10. The sets STGOOD and TGOOD

N	STGOOD	TGOOD	N	STGOOD	TGOOD
1	13	$m_1m_4m_{18}m_{19}m_{23}m_{26}$	9	2,7,8	m_bm_{22}
2	2,10	$m_4m_am_{26}$	10	1,5,12	$m_2m_{23}m_{24}$
3	3,10	$m_3m_4m_{13}m_{18}m_{26}$	11	4,7,12	$m_{20}m_{24}m_{26}$
4	8,10	$m_3m_6m_am_{13}m_{20}m_{21}$	12	3,7,12	$m_3m_{24}m_{26}$
5	9,11	$m_{19}m_{20}m_{21}m_{22}m_{26}$	13	7,8,11	$m_3m_{20}m_{22}$
6	3,11	$m_3m_7m_{19}m_{26}$	14	2,3,4,7	$m_4m_{12}m_bm_{24}m_{26}$
7	3,8	$m_3m_7m_{13}m_bm_{19}$	15	4,6,8,11	$m_7m_{20}m_{21}$
8	1,4,7	$m_5m_6m_{24}m_{26}$	16	1,2,12,14	$m_{23}m_{24}m_{26}$

References

1. Chegis, I., Yablonskii, S.: Logical methods of electric circuit control. Trudy Mian SSSR 51, 270–360 (1958), (in Russian)
2. Ferré, S., Ridoux, O.: The use of associative concepts in the incremental building of a logical context. In: Priss, U., Corbett, D., Angelova, G. (eds.) ICCS. Lecture Notes in Computer Science, vol. 2393, pp. 299–313. Springer (2002)
3. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin (1999)
4. Ganter, B., Kuznetsov, S.O.: Formalizing hypotheses with concepts. In: Proceedings of the Linguistic on Conceptual Structures: Logical Linguistic, and Computational Issues. pp. 342–356. Springer-Verlag (2000)
5. Naidenova, X.A.: DIAGARA: An Incremental Algorithm for Inferring Implicative Rules from Examples. Inf. Theories and Application 12 - 2, 171 – 196 (2005)
6. Naidenova, X.A., Plaksin, M.V., Shagalov, V.L.: Inductive Inferring All Good Classification Tests. In: Valkman, J. (ed.) "Knowledge-Dialog-Solution", Proceedings of International Conference. vol. 1, pp. 79 – 84. Kiev Institute of Applied Informatics, Kiev, Ukraine (1995)
7. Naidenova, X.A., Polegaeva, J.G.: An Algorithm of Finding the Best Diagnostic Tests. In: Mintz, G., Lorents, E. (eds.) The 4-th All Union Conference "Application of Mathematical Logic Methods". pp. 87 – 92 (1986), (in Russian)
8. Naidenova, X., Ermakov, A.: The decomposition of good diagnostic test inferring algorithms. In: Alty, J., Mikulich, L., Zakrevskij, A. (eds.) "Computer-Aided Design of Discrete Devices" (CAD DD2001), Proceedings of the 4-th Inter. Conf., vol. 3, pp. 61 – 68. Minsk (2001)
9. Ore, O.: Galois connections. Trans. Amer. Math. Soc 55, 494–513 (1944)