Computer Science Journal of Moldova, vol.**17**, no.2(50), 2009

Determination of the normalization level of database schemas through equivalence classes of attributes

Cotelea Vitalie

Abstract

In this paper, based on equivalence classes of attributes there are formulated necessary and sufficient conditions that constraint a database schema to be in the second, third or Boyce-Codd normal forms. These conditions offer a polynomial complexity for the testing algorithms of the normalizations level.

Keywords: Relational database schema, functional dependencies, equivalence classes of attributes, normal forms, polynomial algorithms.

1 Introduction

The anomalies that appear during database maintaining are known as insertion, update and deletion anomalies. These are directly related to the dependencies between attributes. A rigorous characterization of the quality grade of a database schema can be made through the exclusion of mentioned anomalies, with consideration of attributes dependencies, which offers the possibility to define some formal techniques for design of desirable relation schemes.

The process of design of some relation scheme structure with intend to eliminate the anomalies, is called *normalization*. Normalization consists in following a set of defined rules on data arrangement with the scope to reduce the complexity of scheme structures and its transformation into smaller and stable structures which will facilitate data maintenance and manipulation. There exist several normalization levels that are called *normal forms*.

The normal forms based on functional dependencies are *first normal form* (1NF), *second normal form* (2NF), *third normal form* (3NF) and *Boyce-Codd normal form* (BCNF). These forms have increasingly restrictive requirements: every relation in BCNF is also in 3NF, every relation in 3NF is also in 2NF and every relation in 2NF is in 1NF. A relation is in 1NF if every attribute contains only atomic values. 2NF is mainly of historical interest. 3NF and BCNF are important from a database design standpoint [1].

For example, the design of a 3NF database schema, through the synthesizing method, can be performed in a polynomial time [2]. Unfortunately, the problem of determination of the normalization level is known to be NP-complete [3, 4], because normalization testing requires finding the candidate keys and nonprime attributes. Firstly, the definitions of normal schemes (second, third or BCNF) contain the notion of key. But it is known that a relation can have an exponential number of keys under the number of all attributes of its scheme [5]. Secondly, the definitions of normal forms use the notions of prime and nonprime attributes, which are also related to key.

The problem of prime and nonprime attributes finding has been solved in a polynomial time [6]. In this paper necessary and sufficient conditions for a scheme to be in 2NF, 3NF or BCNF are defined. These conditions are described in terms of redundant and nonredundant equivalence classes of attributes and the computation of these classes can be performed in polynomial time [6]. Therefore, the determination of normalization level of a scheme is also polynomial. Thus a database designer may work in terms of attributes sets and data dependencies, and not in terms of keys. This approach can be a part of the database analysis and design toolset, i.e. for the automation of database design and testing.

In Section 2, most of the definitions needed in this paper are presented. In Section 3, several properties for equivalence classes of attributes, proved in [6], are given.

Besides this, the correlation is proven between nonredundant classes of attributes and the right and left sides of functional dependency that is inferred from a given set of functional dependencies (Theorem 3). In Sections 4, 5 and 6 there are presented necessary and sufficient conditions (Theorems 4-6), in terms of equivalence classes of attributes, for a relation scheme to be in 2NF, 3NF or BCNF, respectively. The final section is about algorithmic aspects, where it is shown that the determination of the normalization level of database schemas can be performed in polynomial time.

2 Preliminary notions

In this and in the next section, that will be as concise as possible, some definitions and statements used in this paper are presented.

Let $Sch(R, F)$ be a relation scheme, where F is a set of functional dependencies defined on a set R of attributes. The set of all functional dependencies implied by a given set F of functional dependencies is called the *closure* of F and is denoted as F^+, that is $F^+ = \{V \to W|F| = V \to W\}$ [7].

If F is a set of functional dependencies over R and X is a subset of R, then the closure of the set X with respect to F, written as X^+, is the set of attributes A such that $X \to A$ can be inferred using the Armstrong Axioms, that is $X^+ = \{A|X \to A \in F^+\}$ [7].

Armstrong's Axioms are *sound* in that they generate only functional dependencies in F^+ when applied to a set F. They are *complete* in that repeated application of these rules will generate all functional dependencies in the closure F^+ [1].

Let X and Y be two nonempty finite subsets of R. The set X is a determinant for Y with respect to F if $X' \to Y$ is not in F^+ for every proper subset X' of X.

If X is a determinant for R with respect to F, then X is a key for relation scheme $Sch(R, F)$. Note that some relation scheme may have more than one key.

An attribute A is *prime* in $Sch(R, F)$ if A is contained in some key of $Sch(R, F)$. Otherwise A is nonprime in $Sch(R, F)$.

In what follows, it will be assumed that the set F of functional dependencies is reduced [7].

Given a relation scheme $Sch(R, F)$, the set F can be represented by a graph, called contribution graph [6] for F and denoted by $G = (S, E)$, where:

- for every attribute A in R, there is a vertex labeled by A in S;
- for every functional dependence $X \to Y$ in F and for every attribute A in X and every B in Y there is an edge $a = (A, B)$ in E that is directed from vertex A to vertex B.

Let $G = (S, E)$ be divided into strongly connected components. The relation of strong connectivity is an equivalence relation over the set S. So, there is a partition of set of vertices S into pairwise disjoint subsets, that is, $S = \bigcup_{i=1}^{n} S_i$.

Let $S_1, ..., S_n$ be the strongly connected components of a graph $G = (S, E)$. Then the condensed graph [8] of G, $G^* = (S^*, E^*)$ is defined as follows:

$S^* = \{S_1, ..., S_n\}$ and

$E^* = \{(S_i, S_j) | i \neq j,\ (A, B)\ in\ E,\ A \in S_i\ and\ B \in S_j\}$.

Evidently the condensed graph G^* is free of directed circuits. Over the set S^* of vertices of graph G^* a strict partial order is defined. Strict partial orders are useful because they correspond more directly to directed acyclic graphs. Vertex S_i precedes vertex S_j, if S_j is accessible from S_i.

From the ordered sequence of sets $S_1, ..., S_n$ a sequence of ordered nonredundant sets can be built $T_1, ..., T_n$, where $T_1 = S_1$ and $T_j = S_j - (\bigcup_{i=1}^{j-1} T_i)_F^+$ for $j = \overline{2, n}$. All empty sets are excluded from the sequence and a sequence of nonempty sets $T_1, ..., T_m$ is obtained, keeping the precedence of prior sets.

Lemma 1. [6]. If $X \to Y \in F^+$ and X is a determinant of Y under F, then for every attribute $A \in (X - Y)$ there is an attribute $B \in Y$ so that in the contribution graph G there exists a path from vertex A to vertex B and for every attribute $B \in (Y - X)$ there exists in X an attribute A, from which the vertex B can be reached.

3 Some properties of equivalence classes of attributes

In this section a brief overview of several properties of equivalence classes of attributes is given. And their proofs are presented in [6].

Theorem 1. ([**6**], Theorem 2). Set X is a determinant of set $S_1 \bigcup ... \bigcup S_n$ under F, if and only if X is determinant of set $T_1 \bigcup ... \bigcup T_m$ under F.

Lemma 2. ([**6**], Lemma 3). If X is a determinant under F of set $T_1 \bigcup ... \bigcup T_m$, then Z, where $Z = X \bigcap (T_1 \bigcup ... \bigcup T_j)$ and $j = \overline{1, m}$, is a determinant for $T_1 \bigcup ... \bigcup T_j$ under F.

Theorem 2. ([**6**], Theorem 4). If set of attributes X is a determinant of set $T_1 \bigcup ... \bigcup T_m$, then $X \bigcap T_i \neq \emptyset$, where $i = \overline{1, m}$.

Corollary 1. ([**6**], Corollary 3). If an attribute A in $S_1 \bigcup ... \bigcup S_n$ is prime in scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$, then $A \in \bigcup_{i=1}^{m} T_i$.

Corollary 2. ([**6**], Corollary 4). If an attribute A in $S_1 \bigcup ... \bigcup S_n$ is nonprime in scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$, then $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$.

Theorem 3. Let $X \to Y \in F^+$, where X is a determinant for Y under F and $X, Y \subseteq T_1 \bigcup ... \bigcup T_m$. For a T_j, where $j = \overline{1, m}$, the following takes place: if $Y \bigcap T_j \neq \emptyset$, then $X \bigcap T_j \neq \emptyset$.

Proof. The soundness of this statement is proven by contradiction: let $Y \bigcap T_j \neq \emptyset$, but $X \bigcap T_j = \emptyset$. Evidently that $X \subseteq T_1 \bigcup ... \bigcup T_{j-1} \bigcup T_{j+1} \bigcup ... \bigcup T_m$ and $X \to (Y \bigcap T_j) \in F^+$. Let X', where $X' \subseteq X$, is a determinant for $Y \bigcap T_j$ under F. According to Lemma 1, on the contribution graph of set F of dependencies, from every vertex labeled with an attribute in X' there exists a path to a vertex labeled with an attribute in $Y \bigcap T_j$. Thereby, $X' \subseteq T_1 \bigcup ... \bigcup T_{j-1}$. But in this case, T_j is redundant. A contradiction has been reached.

Using above structures and statements it will be shown that the problem of determination of the normalization level has polynomial complexity. In the following sections, in terms of equivalence classes of attributes, sufficient and necessary conditions for a relation scheme to be in a normal form are presented.

4 Second normal form

Thus, the relation scheme in the 2NF can be defined:

Definition 1. [9]. Scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ is in the 2NF under a set of functional dependencies F, if it is in 1NF and each nonprime attribute in $\bigcup_{i=1}^{n} S_i$ doesn't partially depend on every key for Sch. Database schema is in the 2NF, if each constituent relation scheme is in the 2NF.

Definition 2. [9]. Let $X \to A \in F$ be a nontrivial functional dependency (namely $A \notin X$). An attribute A is called *partially dependent* on X, if there exists a proper subset X' of set X, such that $X' \to A \in F^+$. If such a proper subset doesn't exist, then A is called that *completely depends* on X.

Proposition 1. If set of attributes X is a determinant for attribute A under set of attributes F, then A completely depends on X.

The next theorem gives a characterization of the 2NF in terms of equivalence classes of attributes.

Theorem 4. Relation scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ is in the 2NF, if and only if it is in the 1NF and for every T_j, $j = \overline{1, m}$, $(\bigcup_{i=1}^{m} T_i - T_j)^+ = \bigcup_{i=1}^{m} T_i - T_j$ takes place.

Proof. Necessity. Let scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ be in the 2NF. Then every nonprime attribute A, that is a member of set $\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i$ completely depends on every determinant X of set $S_1 \bigcup ... \bigcup S_n$. According to Theorem 1, X is a determinant of set $T_1 \bigcup ... \bigcup T_m$. In addition, $X \subseteq T_1 \bigcup ... \bigcup T_m$. Assuming to the contrary, that Sch is in the 2NF, but there is an attribute $A \in (\bigcup_{i=1}^{m} T_i - T_j)^+$ such that $A \notin (\bigcup_{i=1}^{m} T_i - T_j)$. There are two cases: either $A \in T_j$, or $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$.

Let $A \in T_j$. From the construction of contribution graph, follows that $(T_1 \bigcup ... \bigcup T_j) \to A \in F^+$. Because $A \in (\bigcup_{i=1}^{m} T_i - T_j)^+$, namely $A \in (\bigcup_{i=1}^{j-1} T_i - T_j)^+$, then $(T_1 \bigcup ... \bigcup T_{j-1}) \to A \in F^+$. But this contradicts the fact that set T_j is nonredundant.

Let $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$. If X is a determinant for $T_1 \bigcup ... \bigcup T_m$ under F, taking into account Lemma 2, $X \bigcap (T_1 \bigcup ... \bigcup T_{j-1})$ is a determinant for $T_1 \bigcup ... \bigcup T_{j-1}$. So that $(T_1 \bigcup ... \bigcup T_{j-1}) \rightarrow A \in F^+$, then $(X \bigcap (T_1 \bigcup ... \bigcup T_{j-1})) \rightarrow A \in F^+$. In other words, the nonprime attribute A partially depends on determinant X. That is A partially depends on key X, fact that contradicts the assumption that scheme Sch is in the 2NF.

Sufficiency. Let scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ be in the 1NF and for every T_j, $j = \overline{1, m}$, the following equality takes place: $(\bigcup_{i=1}^{m} T_i - T_j)^+ = \bigcup_{i=1}^{m} T_i - T_j$. It will be proven that scheme Sch is in the 2NF. Two cases are possible: either $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i) = \emptyset$, or $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i) \neq \emptyset$.

If $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i) = \emptyset$, then scheme doesn't contain nonprime attributes and, therefore, scheme is in the 2NF and it is even in the third.

If $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i) \neq \emptyset$, that is in the case when set of nonprime attributes is not empty, results that every nonprime attribute A completely depends on $T_1 \bigcup ... \bigcup T_m$, furthermore it completely depends on determinant X under F of set $T_1 \bigcup ... \bigcup T_m$. So, the scheme is in the 2NF.

Corollary 3. Scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ is in the 2NF, if and only if for every $i = \overline{1, m}$ $T_i = S_i$ holds.

Proof. The soundness of this statement follows from the fact that $(\bigcup_{i=1}^{m} T_i - T_j)^+ = \bigcup_{i=1}^{m} T_i - T_j$ takes place when $T_i = S_i$ holds for every $i = \overline{1, m}$ and vice versa.

5 Third normal form

In this section a characterization of the 3NF is given through the equivalence classes.

Definition 3. [9]. Scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ is in 3NF under a set of functional dependencies F, if it is in the 1NF and every nonprime attribute doesn't transitively depend on a key of scheme Sch. Database schema is in the 3NF, if every constituent relation scheme is in the 3NF.

Definition 4. [10]. Let scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$, $V, W \subseteq \bigcup_{i=1}^{n} S_i$ and $A \in \bigcup_{i=1}^{n} S_i$. It is considered that the attribute A *transitively depends* on V through W, if the following conditions are all satisfied:

1. $V \to W \in F^+$;

2. $W \to V \notin F^+$ (namely V doesn't functionally depend on W);

3. $W \to A \in F^+$;

4. $A \notin VW$.

Theorem 5. Relation scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ is in the 3NF, if and only if $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ is a determinant for $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ under F.

Proof. Necessity. Let scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ be in the 3NF. Then scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ is also in the 2NF and each attribute A, where $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$, fully functionally depends on key X of scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$, namely it is fully functionally dependent on determinant X of set $T_1 \bigcup ... \bigcup T_m$ under F. In addition, no attribute A, where $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$, transitively depends on X. That is, there doesn't exist any dependency $W \to A \in F^+$, such that $W \subseteq (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ and $A \notin XW$. Therefore, dependency $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i) \to (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ is reduced on the left side, fact that confirms that $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ is a determinant for $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ under F.

Sufficiency. Assume $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ is a determinant for $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ under F. Let X be a determinant of set $T_1 \bigcup ... \bigcup T_m$ under F. According to Theorem 1, X is a determinant of set $S_1 \bigcup ... \bigcup S_n$. Hence, $X \to (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i) \in F^+$ holds. Because $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ is a determinant for $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ under F, then there doesn't exist any dependency $W \to A \in F^+$, such that $W \subseteq (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$, $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ and $A \notin XW$. That is, all nonprime attributes A don't depend transitively on determinant X.

6 Boyce-Codd normal form

The concept of BCNF is refined from the notion of 3NF. In the determination of a database schema being in BCNF, a given set F of functional dependencies is used.

Definition 5. [11]. Relation scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ is in BCNF under set F of functional dependencies, if it is in the 1NF and for every nontrivial dependency $V \to A \in F^+$ $V \to \bigcup_{i=1}^{n} S_i \in F^+$ takes place, that is, the left side of each functional dependency functionally determines all attributes of scheme.

Theorem 6. Scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ is in the normal form Boyce-Codd, if and only if it is in the 3NF and for every T_j, $j = \overline{1,m}$, the set of attributes $(\bigcup_{i=1}^{m} T_i - T_j)$ is a determinant for $(\bigcup_{i=1}^{m} T_i - T_j)$ under F.

Proof. Necessity. Let scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ be in BCNF. Then for every nontrivial functional dependency $V \to A \in F^+$, that is, the case when $A \notin V$, $V \to \bigcup_{i=1}^{n} S_i \in F^+$ holds. Based on the reflexivity rule, $(\bigcup_{i=1}^{m} T_i - T_j) \to (\bigcup_{i=1}^{m} T_i - T_j) \in F^+$. If it's supposed that $(\bigcup_{i=1}^{m} T_i - T_j)$ is not a determinant for $(\bigcup_{i=1}^{m} T_i - T_j)$ under F, that is, if there exists a set of attributes $V \subset (\bigcup_{i=1}^{m} T_i - T_j)$, so that $V \to (\bigcup_{i=1}^{m} T_i - T_j) \in F^+$, then the last dependency is not trivial. By the definition of BCNF $V \to \bigcup_{i=1}^{n} S_i \in F^+$ holds. But this functional dependency contradicts the fact that every determinant $X$of set $\bigcup_{i=1}^{n} S_i$ and consequently a set $\bigcup_{i=1}^{m} T_i$ contains, according to Theorem 2, attributes in T_j $j = \overline{1,m}$ too, namely $X \bigcap T_j \neq \emptyset$ for $j = \overline{1,m}$.

Sufficiency. Let scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ be in the 3NF and for every T_j, $j = \overline{1,m}$, the set of attributes $(\bigcup_{i=1}^{m} T_i - T_j)$ is a determinant for $(\bigcup_{i=1}^{m} T_i - T_j)$ under F. It will be proven that scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ is in BCNF.

Let scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ not be in BCNF. In this case, there is a nontrivial functional dependency $V \to A \in F^+$, so that $V \to \bigcup_{i=1}^{n} S_i \notin F^+$ holds. Then it can be stated that $V \to \bigcup_{i=1}^{m} T_i \notin F^+$. Without constraining the generality, let V be a determinant for A under F. From the construction of the contribution graph and from the fact that dependency $V \rightarrow A$ is reduced, three cases can be examined (other cases don't exist):

1. $V \subseteq (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ and $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$, that is, left and right sides are formed just from nonprime attributes,

2. $V \subseteq \bigcup_{i=1}^{m} T_i$ and $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ - the left side is formed from prime attributes, and the right side consists of a nonprime attribute.

3. $V \subseteq \bigcup_{i=1}^{m} T_i$ and $A \in \bigcup_{i=1}^{m} T_i$, that is, left and right sides are formed just from prime attributes,

Suppose that $V \subseteq (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ and $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$, then nonprime attribute A would transitively depend through V on every determinant of set $\bigcup_{i=1}^{n} S_i$ and then scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ will not be in the 3NF, which contradicts the hypothesis.

If it is considered that $V \subseteq \bigcup_{i=1}^{m} T_i$ and $A \in (\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$, then nonprime attribute A would partially depend on a determinant of set $\bigcup_{i=1}^{n} S_i$ therefore scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ will not be in the 2NF, that is, neither in the third, fact that contradicts the hypothesis.

If it is considered that $V \subseteq \bigcup_{i=1}^{m} T_i$ and $A \in \bigcup_{i=1}^{m} T_i$, then the set $(\bigcup_{i=1}^{m} T_i - T_j)$ of attributes is not a determinant for $(\bigcup_{i=1}^{m} T_i - T_j)$ under F. Indeed, let $m > 1$ and $A \in T_k$. Then by Theorem 3 and the construction way of drawing the contribution graph, it can exist two cases either $V \subseteq T_k$, or $V \not\subset T_k$, but $V \subseteq (T_l \bigcup T_{l+1} \bigcup ... \bigcup T_k)$, where $V \bigcap T_i \neq \emptyset$, $i = \overline{l,k}$. Evidently $m > k - l + 1$, but in this case there exists a T_j, where $V \bigcap T_j = \emptyset$, so that $(\bigcup_{i=1}^{m} T_i - T_j)$ is not a determinant for $(\bigcup_{i=1}^{m} T_i - T_j)$ under F, because $((\bigcup_{i=1}^{m} T_i - \{A\}) - T_j) \rightarrow (\bigcup_{i=1}^{m} T_i - T_j) \in F^{+}$.

7 Algorithms' complexities

Based on the above characterization, the polynomiality of the normal form testing problem can be proved. A few comments about the complexity of the algorithms for finding the normal form of scheme are made below.

Both construction of equivalence classes of scheme's attributes and redundancy elimination from these classes have a complexity $O(|R| \cdot ||F||)$ [6].

It is not hard to calculate the complexity of algorithms that determine whether a scheme is in the second, third or Boyce-Codd normal form. That is, $|R| \cdot ||F||$ for each of these algorithms. This is explained through the fact that the complexity of calculation of the classes of nonredundant attributes exceeds the complexity of calculation of the verification conditions that determine if a scheme is in one of the enumerated forms.

Thus, if nonredundant classes $\bigcup_{i=1}^{m} T_i$ are built, then calculation of the condition for the scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ to be in the 2NF (that is, if for every T_j, $j = \overline{1,m}$, $(\bigcup_{i=1}^{m} T_i - T_j)^+ = \bigcup_{i=1}^{m} T_i - T_j)$ requires a time $O(|NonRedEquivClasses| \cdot ||F||)$. Therefore the time is $O(|R| \cdot ||F||)$.

Computation of the condition for the scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ to be in the 3NF (that is, if $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i)$ is a determinant for $(\bigcup_{i=1}^{n} S_i - \bigcup_{i=1}^{m} T_i))$ requires a time $O(||F||)$.

Similarly, verification of the condition for the scheme $Sch = (\bigcup_{i=1}^{n} S_i, F)$ to be in BCNF (that is, if for every T_j, $j = \overline{1,m}$, set of attributes $(\bigcup_{i=1}^{m} T_i - T_j)$ is a determinant for $(\bigcup_{i=1}^{m} T_i - T_j)$ under F) requires a time $O(|R| \cdot ||F||)$.

References

[1] Ramakrishnan, Raghu and Gehrke, Johannes. *Database Management Systems.* Second Edition, McGraw-Hill Higher Education, 2000, 900 pp.

[2] Bernstein, Philip A. *Synthesizing Third placeNormal Form Relations from Funcional Dependencies.* ACM Trans. Database Syst., V.1, N 4, 1976, p.277–298.

[3] Beeri, C.; Bernstein, P.A. *Computational Problems Related to the design of placeNormal Form Relations Schemes.* ACM Trans. Database Syst., V.4, N 1, March 1979, p.30–59.

[4] Jou, J.H., Fisher, P.C. *The complexity of recognizing 3FN relation schemes.* Inform. Process. Letters 14, 1982, p.187–190.

[5] Yu C.T., Johnson D.T. *On the complexity of finding the set of candidate keys for a given set of functional dependencies.* Information Processing Letters, V.5, N.4, 1978, p.100–101.

[6] Cotelea, Vitalie. *An approach for testing the primeness of attributes in relational schemas.* Computer Science Journal of Moldova, Chisinau, Vol.17, Nr.1(49), 2009, p. 89–99.

[7] Maier, D. *The theory of relational database.* Computer Science Press, 1983, 637 p.

[8] Even, Shimon. *Graph Algorithms.* Computer Science press, 1979, 250 p.

[9] Codd, E.F. *Further Normalization of the data base relational model.* Data Base Systems, R. Rustin (ed), Prentice Hall, 1972, p. 33–64,.

[10] Chao-Chih Yang. *Relational Databases.* Englewood Cliffs. NJ, Prentice-Hall, 1986, 260 p.

[11] Codd, E.F. *Recent Investigation in Relation Data Base Systems.* IFIP Congress, 1974, p. 1017–1021.

Vitalie Cotelea

Received May 27, 2009

Vitalie Cotelea
Academy of Economic Studies of Moldova
Phone: (+373 22) 40 28 87
E-mail: *vitalie.cotelea@gmail.com*

Computer Science Journal of Moldova, vol.17, no.2(50), 2009

Modelling Inflections in Romanian Language by P Systems with String Replication

Artiom Alhazov, Elena Boian,
Svetlana Cojocaru, Yurii Rogozhin

Abstract

The aim of this article is the formalization of inflection process for the Romanian language using the model of P systems with cooperative string replication rules, which will make it possible to automatically build the morphological lexicons as a base for different linguistic applications.

1 Introduction

Natural language processing has a wide range of applications, the spectrum of which varies from a simple spell-check up to automatic translation, text and speech understanding, etc. The development of appropriate technology is extremely difficult due to the specific feature of multidisciplinarity of the problem. This problem involves several fields such as linguistics, psycholinguistics, computational linguistics, philosophy, computer science, artificial intelligence, etc.

As in many other fields, solving of a complex problem is reduced to finding solutions for a set of simpler problems. In our case among the items of this set we find again many traditional compartments of the language grammar. The subject of our interest is the morphology, and more specifically, its inflectional aspect.

The inflectional morphology studies the rules defining how the inflections of the words of a natural language are formed, i.e., the aspect of form variation (of the inflection, which is the action of words modification by gender, number, mood, time, person) for various expressing grammatical categories.

In terms of natural language typology the morphological classification can be *analytical* and *synthetic*. Of course, this classification is a relative one, having, however, some irrefutable poles: Chinese, Vietnamese, as typical representatives of the analytical group, and Slavic and Romance languages serving as examples of synthetic ones. The English language, with a low degree of morpheme use, is often among the analytical ones, sometimes is regarded as synthetic, indicating however that it is "less synthetic" comparatively with other languages from the same group. It is evident that it is the inflectional morphology of synthetic languages that presents special interest, being a problem more complex comparatively with analytical class.

The object of our studies is the Romanian language, which belongs to the category of synthetic flective languages. The last notion stresses the possibility to form new words by declension and conjugation. Moreover, the Romanian language is considered a highly inflectional language, because the number of word-forms is big enough.

The inflection simplicity in English makes that the majority of researchers in the field of computational linguistics neglect the inflection morphology. For efficient processing of other natural languages, including Romanian, it is necessary to develop suitable computational models of morphology of each language. In the case of Romanian language, some inflectional models are known [25], [19],[7].

In [25] it is certified an advanced number of morpho-syntactic specifications for Romanian language, namely 34 for nouns, 44 for verbs, 24 for adjectives, 15 for pronouns, etc. The aim of our paper is to describe the process of inflection (i.e. the process of obtaining both the derivative words and their morphological attributes) by P systems [17]. This paper is a final version of [1].

2 Description of the inflection process

To develop a formalism for the inflection process description we invoke a number of definitions and notions which allow us to understand the essence of this process. Inflection is a part of morphology - the science which "includes the rules considering the word forms and the formal modifications of the words" [24]. From the morphological point of view the words are classified corresponding to the part of speech, and their structure is described in terms of inflection, derivation and composition. Inflection is the systematic variation of the word form which allows to obtain different semantic and syntactic functions [10]. The words combine in themselves two components: a *constant* and a *variable* [12]].

The root of primary lexical units is called the *constant.* For the derivative ones the term *lexical theme* is used. Since in our study this distinction does not play any role, for both cases we use a single term "*root*".

The *variable* is the bearer of grammatical meanings, it consists of one or more morphemes being called also *flective.* This term will be used in exposure below. In accordance with [24] we identify three ways of achieving the inflections:

analytical: the flective is a free morpheme (separated from root) and the root remains invariable (e.g., adverb, *bine – mai bine* (engl. well - better));

synthetic: the flective is a conjunctive morpheme (group of morphemes), related to the root (e.g., for noun, pronoun; *studentă – studente – studentei; care-căreia-căruia-cărora* (engl. *student – students – student's, who-whose-whom*), etc.).

synthetic and analytical: the flective consists of free and conjunctive morphemes (e.g., adjective, verb, *frumos – frumoasă – mai frumoasă; cântasem – am cântat* (engl. beautiful – beautiful – more beautiful, singing – I sang), etc.).

In the following we will deal with the synthetic method, the analytical one is effectuated relatively easy through a set of simply formulated rules. Following the model from [10] we present in Figure 1 the classification of Romanian language parts of speech in terms of the inflection process.

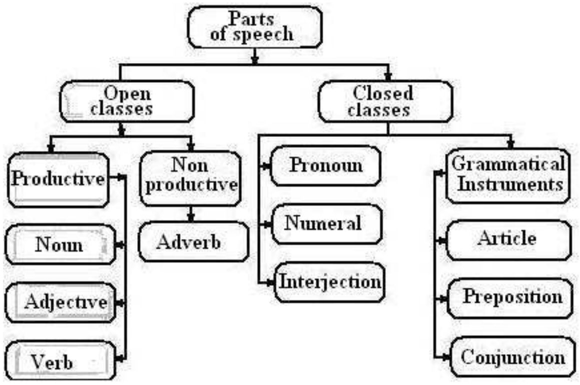

Figure 1. The classification of the Romanian language parts of speech (in terms of the inflection process.)

The class of opened productive parts of speech is the most interesting in terms of inflection, and it will be the primary object of our investigations.

Indeed, opened classes, containing tens of thousands of elements, are characterized by a productive process of inflection, derivation and composition, while the closed ones include a reduced number of items (practically excluding the possibility of the new ones apparition), because the morphological processes of word formation are poorly productive [12]. Moreover, in the case of opened classes the problem is complicated not only because we cannot enumerate the elements, existing at the moment, but also because a successful formalism should be able to “serve” the future neologisms that could occur in language development process. In the following we will operate with the paradigms of inflection, by which we imply the systematic arrangement of all inflection forms of a word [13].

For our purposes we will work not with the whole words, but with their variable parts. Hereinafter by paradigm we mean a list of flectives.

For each flective we can put into correspondence a set of morphological attributes.

Example. Let us examine the morphological attributes for masculine nouns of Romanian language [25].

N	noun (part of speech),
m	masculine gender,
s	singular number,
p	plural number,
d	direct (nominative – accusative cases),
o	oblique (genitive – dative cases),
v	vocative case,
y	yes – definiteness,
n	no – definiteness.

(Given that the Romanian forms for nominative and accusative cases coincide, as well as for the genitive and dative ones, we reduced the paradigm merging both word forms, and respective attributes.)

Thus, the list of flectives $F = \{-, -, -,$ ul, ului, ule, i, i, i, ii, ilor, ilor$\}$, where "$-$" denotes the empty word, can be regarded as a morphologically annotated one.

$$\begin{aligned} F_{morf} \; = \{ \;\; & (-, Nmsdn), (-, Nmson), (-, Nmsvn), \\ & (\text{ul}, Nmsdy), (\text{ului}, Nmsoy), (\text{ule}, Nmsvy), \\ & (\text{i}, Nmpdn), (\text{i}, Nmpon), (\text{i}, Nmpvn), \\ & (\text{ii}, Nmpdy), (\text{ilor}, Nmpoy), (\text{ilor}, Nmpvy)\}. \end{aligned}$$

Let us mention the use of paradigmatic model for the Romanian language [8, 9, 20, 21, 22].

We will refer also to the works [18] and [11], which treat the subject of generation of the flectioned forms for the Romanian language. The authors do not provide the inflection algorithms, but offer some useful suggestions for generation of flectioned forms. In paper [18] it is proposed a method of encoding vowel and consonant alternations in the root, taken by the authors from researches of acad. G. Moisil, namely: each alternation is presented in the root by a distinct code. In paper [11] it is found a (incomplete) set of rules, which indicates the way of concatenation of flective for nouns and adjectives without concerning the problem of the alternations in the root. Therefore, having the aim to achieve the synthetic model of inflection, we must develop a formalism, which should include two processes:

- making the alternation in the root, and
- concatenation of a flective.

The starting point of our approach was the dictionary [13], in which the flective words of Romanian language are classified according to the way of inflections formation. There were set 100 groups of inflection for masculine nouns, 273 – for verbs, etc. A dictionary of about 30,000 words with the specification of the number of the group was constructed. The classification was made taking into account all linguistic aspects, e.g. accents. In our case we will focus only on the way of writing a word, which in equal measure simplifies and complicates the problem. However this classification is extremely useful suggesting us the idea of defining a special class of grammars to formalize the inflection process [2, 3, 4, 5].

In general case, from a whole variety of inflection groups, we can identify two classes:

– without alternations, and

– with alternations.

In the first case the inflection is made in the following manner. Let $\Im$ be a set formed from lists of flectives, $F = \{f_1, f_2, \cdots, f_n\}$, $w = w'\alpha$ is a word-lemma, where $|\alpha| \geq 0$. In the simplest case the inflected words will be those of the form $w'f_i$, $f_i \in F$, $(i = 1, \cdots, n)$.

General case: Let $w = w_1 a_1 w_2 a_2 \cdots w_m \alpha$. The inflected words will be of the form:

$$\begin{array}{llllll}
w^{(1)} = w_1 & a_1 & w_2 & a_2 & \cdots & w_m f_{i_1}, \\
w^{(2)} = w_1 & u_1^{(2)} & w_2 & u_2^{(2)} & \cdots & w_m f_{i_2}, \\
\ \ \cdots & & & & & \\
w^{(s)} = w_1 & u_1^{(s)} & w_2 & u_2^{(s)} & \cdots & w_m f_{i_s},
\end{array}$$

where w_i, $a_i \in V^+$, $u_i^{(j)} \in V^*$, $f_{i_1} \in F^{(1)}, \ldots, f_{i_s} \in F^{(s)}$, and $F^{(1)} \cup \ldots \cup F^{(s)}$ forms a complete paradigm.

Note: the analysis of inflection rules allowed us to ascertain that for the Romanian language $m \leq 4$, $s \leq 3$.

Example 1. Inflection of masculine nouns without alternations.

Let $F = \{-, -, -, ul, ului, ule, i, i, i, ii, ilor, ilor\}$ – a list of flectives, where '-' denotes the empty word. Let $w =$'stejar' (engl. *oak*), $|\alpha| = 0$, $|F| = 12$. The set of inflected words supplied by morphological attributes will be:

$$\{\ \begin{array}{lll}
(\text{stejar}, Nmsdn), & (\text{stejar}, Nmson), & (\text{stejar}, Nmsvn), \\
(\text{stejarul}, Nmsdy), & (\text{stejarului}, Nmsoy), & (\text{stejarule}, Nmsvy), \\
(\text{stejari}, Nmsdn), & (\text{stejari}, Nmpon), & (\text{stejari}, Nmpvn), \\
(\text{stejarii}, Nmpdy), & (\text{stejarilor}, Nmpoy), & (\text{stejarilor}, Nmpvy)\ \}
\end{array}$$

Taking advantage of paradigmatic ordering of the elements from the list of flectives, in what follows we will omit the explicit writing of morphological attributes implying their conformity to respective flectives.

Example 2. Inflection of masculine nouns with alternations.

Let $w =$*tânăr* (engl. *young*), $|\alpha| = 0$. The vowel alternations *â*→ *i* and *ă*→ *e* will be used. The obtained roots $w =$*'tânăr'* and $w' =$*'tiner'* are respectively annexed by the endings: $F_1 = \{-, -, \text{ul}, \text{ului}, \text{ule}\}$ and $F_2 = \{e, \text{i}, \text{i}, \text{i}, \text{ii}, \text{ilor}, \text{ilor}\}$, $|F_1| + |F_2| = 12$.

$$\{\ \begin{array}{lll}
(\text{tânăr}, Nmsdn), & (\text{tânăr}, Nmson), & (\text{tânărule}, Nmsvy), \\
(\text{tânărul}, Nmsdy), & (\text{tânărului}, Nmsoy), & (\text{tinere}, Nmsvn), \\
(\text{tineri}, Nmsdn), & (\text{tineri}, Nmpon), & (\text{tineri}, Nmpvn), \\
(\text{tinerii}, Nmpdy), & (\text{tinerilor}, Nmpoy), & (\text{tinerilor}, Nmpvy)\ \}
\end{array}$$

Note: In most cases (for 80 groups of inflexion from [13]), when declining the masculine noun, 12 words are obtained. Exceptions are the following nouns:

– irregular, for example, those which can not have the plural definite form (instance, the word *gnu*);

– those which are singularia tantum (nouns which appear only in the singular form), *ianuarie* etc.;

– those which are pluralia tantum (nouns that appear only in the plural and do not have a singular form), for example, *ochelari, pantaloni* etc.

In general, the 100 groups of inflection of masculine nouns in relation to the number of words produced at inflection, present the following table:

Forms of the lemma	Number of forms	Number of groups
all forms	12	80
singularia tantum	6	13
pluralia tantum	6	4
irregular	6-8	3

Modern dictionaries contain hundreds of thousands of words–lemma. Their forms of inflexion (the amount of which exceeds millions) are needed for developing various applications based on natural language: from the spell-checker up to the systems understanding the speech. Obviously, to solve the problem of creating a dictionary with a morphologically representative coverage, as well as to build various applications based on it, effective mechanisms are needed, especially those that allow parallel processing. One of the possible ways to perform parallel computation is based on biological models.

Let us mention a series of works that used the biological calculation approaches for solution of linguistic problems. In [15] there are presented some attempts to construct linguistic membrane systems and some applications related to analysis of conversational acts, bio-inspired for dealing with semantics. In [16] two parsing methods using P automata are presented. The first method uses P automata with active membranes for parsing natural language sentences into dependency trees. The second method uses a variant of P automata with evolution and communication rules for parsing Marcus contextual Languages [14].

Our paper tries to expand the area of potential applications of P systems to linguistics problems, introducing a formalism to capture inflections with their morphological attributes.

To formalize the inflection process for the Romanian language the model of cooperative membrane P systems with replication will be used [17].

3 P systems with string replication and input

Let us recall the basics of P systems with string objects and input. The membrane structure μ is defined as a rooted tree with nodes labeled $1, \cdots, p$. The objects of the system are strings (or words) over a finite alphabet O. A sub-alphabet $\Sigma \subseteq O$ is specified, as well as the input region i_0, $1 \leq i_0 \leq p$. In this paper we need to use cooperative rewriting rules (i.e. string rewriting rules, not limited by context-free ones) with string replication and target indications.

A rule $a \to u_1$, where $a \in O^+$ and $u_1 \in O^*$, can transform any string of the form w_1aw_2 into $w_1u_1w_2$. Application of a rule $a \to u_1||u_2|| \cdots ||u_k$ transforms any string of the form w_1aw_2 into the multiset of strings $w_1u_1w_2$, $w_1u_2w_2$, $\cdots$, $w_1u_kw_2$. If in the right side of the rule (u_i, t) is written instead of some u_i, $1 \leq i \leq k$, $t \in \{out\} \cup \{in_j \mid 1 \leq j \leq p\}$, then the corresponding string would be sent to the region specified by t.

Hence, such a P system is formally defined as follows:

$$\begin{array}{lcl} \Pi & = & (O, \Sigma, \mu, M_1, \cdots, M_p, R_1, \cdots, R_p, i_0), \text{ where} \\ M_i & & \text{is the multiset of strings initially present in region } i, 1 \leq i \leq p, \\ R_i & & \text{is the set of rules of region } i,\ 1 \leq i \leq p, \\ & & \text{and } O, \Sigma, \mu, i_0 \text{ are described above.} \end{array}$$

The initial configuration contains the input string(s) over Σ in region i_0 and strings M_i in regions i. Rules of the system are applied in parallel to all strings in the system. The computation consists in non-deterministic application of the rules in a region to a string in that region. The computation halts when no rules are applicable. The result of the computation is the set of all words sent out of the outermost region (called skin).

4 Describing the inflection process by P systems

Let us define the P system performing the inflection process. Let L be the set of words which form opened productive classes. We will start by assuming that the words in L are divided into groups of inflection, i.e. for each $w \in L$ the number of inflection group is known [13]. The inflection group is characterized by the set $G = \{\alpha, R_G, F_G\}$, where $|\alpha| \geq 0$ is the length of ending which is reduced in the process of inflection, F_G is the set of the lists of flectives, the assembly of which forms complete paradigm, R_G is the set of the rules, which indicate vowel/consonant alternation of type $a \to u$, $a \in V^+$, $u \in V^*$, and also the conformity of the roots obtained by the lists of flectives from F_G. To each group of inflexion a membrane system Π_G will be put into correspondence.

As it was mentioned earlier, we will investigate two cases:
– without alternations, and
– with vowel/consonant alternation.

The first model is very simple. For any group $G = (\alpha, \emptyset, \{f_{1_G}, f_{2_G}, \cdots, f_{n_G}\})$ of inflection without alternation,

$$\begin{aligned}
\Pi_G &= (O, \Sigma, [\ \ ]_1, \emptyset, R_1, 1), \text{ where}\\
O &= \Sigma = V \cup \{\#\},\\
V &= \{\text{a}, \cdots, \text{z}\} \text{ is the alphabet of the Romanian language, and}\\
R_1 &= \{\alpha\# \to (f_{1_G}, out)||(f_{2_G}, out)|| \cdots ||(f_{n_G}, out)\}
\end{aligned}$$

If this system receives as an input the words $w'\alpha\#$, where $w'\alpha$ corresponds to the inflection group G, then it sends all its inflected words out of the system in one step. Clearly, Π_G is non-cooperative if $\alpha = \lambda$, but non-cooperativeness is too restrictive in general, since then the system would not be able to distinguish the termination to be reduced from any other occurrence of α.

The general model will require either a more complicated structure, or a more sophisticated approach. Let G be an arbitrary inflection group, with $m-1$ alternations $a_1 = a_1^{(1)}a_2^{(1)}\cdots a_{n_1}^{(1)}, \cdots, a_m = a_1^{(m)}a_2^{(m)}\cdots a_{n_m}^{(m)}$. Let the set of flectives consist of s subsets, and for subset $F_{k_G} = \{f_1^{(k)}, \cdots, f_{p_1}^{(k)}\}$, $1 \le k \le s$, the following alternations occur: $a_1 \to u_1^{(k)}$, $\cdots$, $a_m \to u_m^{(k)}$ (the alternations are fictive for $k=1$), and $\bigcup_{k=1}^{s} F_{k_G}$ corresponds to a complete paradigm. For instance, Example 2 corresponds to $s=2$ sublists (singular and plural), and $m-1=2$ alternations.

The associated P system should perform the computation

$$w\# = \prod_{j=1}^{m-1}(w_j a_j)\, w_m \alpha\# \Rightarrow^*$$

$$\Rightarrow^* \left\{ \prod_{j=1}^{m-1}\left(w_j u_j^{(k)}\right) w_m f_{i_k} \mid 1 \le k \le s,\ f_{i_k} \in F^{(k)} \right\},$$

where $u_j^{(1)} = a_j$, $1 \le j \le m$.

The first method assumes the alternating subwords a_j are present in the input word in just one occurrence, or marked. Moreover, we assume that carrying out previous alternations does not introduce more occurrences of the next alternations.

For modeling such process of inflection for the group G we define the following P system with $1+(s-1)m$ membranes

$$\begin{array}{rcl}
\Pi'_G & = & (O,\Sigma,\mu,\emptyset,\cdots,\emptyset,R_1,\cdots,R_{1+(s-1)m},1),\ \text{where}\\
\Sigma & = & V\cup\{\#\},\\
O & = & \Sigma\cup E,\\
\mu & = & [\ [\ \ ]_2[\ \ ]_3\cdots[\ \ ]_{1+(s-1)m}\ ]_1,\\
E & = & \{\#_k\mid 2\le k\le s\}\cup\{A_{k,j}\mid 1\le k\le s,\ 1\le j\le m\},\\
V & = & \{\text{a},\cdots,\text{z}\}\ \text{is the alphabet of the Romanian language,}
\end{array}$$

(V can be extended by marked letters if needed), and the rules are given below.

$$\begin{array}{rcl}
R_1 & = & \{\alpha\#\to A_{1,m}||(\#_2,in_2)||\cdots||(\#_s,in_s)\}\\
 & \cup & \{A_{k,j}\to(\lambda,in_{k+(s-1)j})\mid 2\le k\le s,1\le j\le m-1\}\\
 & \cup & \{A_{k,m}\to(f_1^{(k)},out)||\cdots||(f_{pm}^{(k)},out)\mid 1\le k\le s\},\\
R_{k+(s-1)(j-1)} & = & \{a_j\to(u_j^{(k)}A_{k,j},out)\},\ 2\le k\le s,\ 1\le j\le m-1,\\
R_{k+(s-1)(m-1)} & = & \{\#_k\to(A_{k,m},out)\},\ 2\le k\le s.
\end{array}$$

The work of P system Π'_G is the following. First, s copies of the string are made, and the first one stays in the skin, while others enter regions $2,\cdots,s$. Each copy in region k is responsible to handle the k-th subset of inflections. The first one simply performs a replicative substitution in the end, and sends the results out, in the same way as Π_G works. Consider a copy of the input in region k, $2\le k\le s$. When j-th alternation is carried out, the string returns to the skin, and symbol $A_{k,j}$ is additionally produced. This symbol will be used to send the string in the corresponding region to carry out alternation $j+1$. Finally, if $j=m$, then the system performs a replicative substitution in the end, and sends the results out.

Assuming $s\ge 2$, the system halts in $2m+1$ steps, making an efficient use of scattered rewriting with parallel processing of different inflection subsets. For instance, the inflection group from Example 2 would transform into a P systems with 4 membranes, halting in 7 steps. Notice that this system is non-cooperative if $\alpha=\lambda$ and $|a_j|=1$,

$1 \leq j \leq m$. It is also worth noticing that it is possible to reduce the time to $m+1$ steps by using tissue P systems with parallel channels.

The second method avoids the limiting assumptions of the first methods. More exactly, it performs the first alternation at its leftmost occurrence, the second alternation at its leftmost occurrence which is to the right of the first one, etc. Formally, such a P system discovers the representation of the input string as $\prod_{j=1}^{m-1}(w_j a_j)\, w_m \alpha$, where a_j has no other occurrences inside $w_j a_j$ except as a suffix.

A theoretical note: overlapping occurrences or occurrences with context can be handled by rules with a longer left-hand side. A different order of occurrences of the alternations can be handled by renumbering the alternations. Should the specification of a group require, e.g., second-leftmost occurrence for $a \rightarrow u$, this can be handled by inserting a fictive substitution $a \rightarrow a$ before $a \rightarrow u$, etc. Therefore, this is the most general method.

We construct the following P system, which takes the input in the form

$$\#_l w \#_r = \#_l \prod_{j=1}^{m-1}(w_j a_j)\, w_m \alpha \#_r.$$

$$\begin{array}{rcl}
\Pi''_G & = & (O, \Sigma, [\ \ ]_1, \emptyset, R_1, 1), \text{ where} \\
\Sigma & = & V \cup \{\#_l, \#_r\}, \\
O & = & \Sigma \cup E, \\
E & = & \{A_{k,j} \mid 1 \leq k \leq s,\ 0 \leq j \leq m\}, \\
V & = & \{a, \cdots, z\} \text{ is the alphabet of the Romanian language,}
\end{array}$$

and the rules are given below.

$$\begin{array}{rcll}
R_1 & = & \{\#_l \rightarrow A_{1,0} || \cdots || A_{s,0}\} & (1) \\
 & \cup & \{A_{k,j-1}\gamma \rightarrow \gamma A_{k,j-1} \mid \gamma \in V \setminus \{a_1^{(j)}\}, & \\
 & & 1 \leq k \leq s,\ 1 \leq j \leq m\} & (2) \\
 & \cup & \{A_{k,j-1} a_1^{(j)} v\gamma \rightarrow a_1^{(j)} A_{k,j-1} v\gamma \mid a_1^{(j)} v \in Pref(a_j), &
\end{array}$$

$$|v| < |a_j| - 1, \gamma \in V \setminus \{a_1^{(|v|+2)}\},\ 1 \le k \le s, 1 \le j \le m\} \tag{3}$$

$$\cup \quad \{A_{k,j-1}a_j \to u_j^{(k)} A_{k,j} \mid 1 \le k \le s,\ 1 \le j \le m\} \tag{4}$$

$$\cup \quad \{\alpha A_{k,m}\#_r \to (f_1^{(k)}, out)||\cdots||(f_{p_m}^{(k)}, out) \mid 1 \le k \le s\}. \tag{5}$$

The rules are presented as a union of 5 sets. The rule in the first set replicates the input for carrying out different inflection subsets. The symbol $A_{k,j}$ is a marker that will move through the string. Its index k corresponds to the inflection subset, while index j tells how many alternations have been carried out so far.

The rules in the second set allow the marker to skip a letter if it does not match the first letter needed for the current alternation. The rules in the third set allow the marker to skip one letter if some prefix of the needed subword is found, followed by a mismatch. The rules in the fourth set carry out an alternation, and the last set of rules perform the replicative substitution of the flectives.

This system halts in at most $|w| + 2$ steps.

5 Determining the inflection group

The rules of the systems described above define, in fact, the way of inflection at algorithmic level:

– deleting the given number of symbols at the end of the word (α),

– obtaining the roots by making substitutions (vowel and consonant alternations),

– attachment of the respective endings to each root.

But this method can be applied only for the case when the number of the inflexion group is known. Otherwise there appears the problem of inflexion model establishing, knowing the graphical representation of the word. Is it possible to solve algorithmically this problem? The answer is negative. The first obstacle is the determination of part of speech: there are several examples of homonyms which mean different parts of speech. (Example: *abate* – masculine noun (*abbat*) and verb (*to divert*). In English this phenomenon is very common, and most nouns are the verbs too.) Let us restrict the formulation of the problem: is it possible to establish the model of inflection (in the conditions indicated above) knowing the part of speech? The answer is negative in this case too. For confirmation we can bring a list of examples, which show us that without invoking phonetic information or the etymological one we cannot determine the model of inflection. Let us illustrate this assertion by analyzing female noun *masă*. Following the meaning of furniture object we will form plural *mese*, using the model with vowel alternation $a \rightarrow e$. But if you are following the meaning "compact crowd of people" [23], the plural *mase* will be produced without alternation. The origin of this phenomenon is etymological: in the first case the origin of the word is from Latin *mensa*, and in the second – from the French word *masse* [23]. But the problem can be tackled in another way: we can set certain criteria that allow us as a result of analysis of the word structure to conclude, if it is possible to determine the inflection model or not. If so, we determine precisely which is the respective model.

In [6] the algorithm had been proposed, which, analyzing the dictionary of classification into morphological groups with entries of type (w, σ), where w is a word in natural language, and σ – number (label) of inflection group, constructs two groups of sets $A = \{A_1, A_2, \ldots, A_k\}$ and $P = \{P_1, P_2, \ldots P_s\}$, $\cap_{i=1}^{k} A_i = \emptyset$, $\cap_{i=1}^{s} P_i = \emptyset$. $A_i \cap P_j = \emptyset$.

These sets consisted of subwords α_i of the words $w = w'\alpha_j$, where $1 \leq |\alpha_j| \leq |w|$. In [6] it is shown that for certain categories of words it is possible to construct such sets A_i, that from the fact that $\alpha_j \in A_i$ it results unequivocally that the word w belongs to the single inflection group σ, and these words being named "absolutely regular". With the help of the same algorithm there are constructed also such sets P_i, that from the fact that $\alpha_j \in P_i$ it results that $w = w'\alpha_j$ can belong to several inflection groups $\sigma_1, \ldots, \sigma_m$, and the respective words being named "partially regular".

So, in the case of an arbitrary word w, using the algorithm mentioned above, the inflection group is established at first, and then with the help of membrane system described above, the inflection is carried out obtaining word forms (with respective morphological attributes).

6 Conclusions

The membrane system to describe the inflexional process when the inflexional morphological model is known is investigated in this article.

In the case when the model is not known in advance, it can be determined by using the algorithm from [6]. The membrane systems presented in this paper can be also adapted for other natural languages with high level of inflection, such as Italian, French, Spanish etc., having structured morphological dictionaries, similar to the Romanian one.

Future work: we plan to also consider the problem of representation of the algorithm determining the inflection group by membrane systems.

Acknowledgments The authors acknowledge the support of the Science and Technology Center in Ukraine, project 4032 "Power and efficiency of natural computing: neural-like P (membrane) systems". The first author also acknowledges the support of the Japan Society for the Promotion of Science, and the Grant-in-Aid, project 20·08364. The fourth author gratefully acknowledges the support of the European Commission, project MolCIP, MIF1-CT-2006-021666.

References

[1] A. Alhazov, E. Boian, S. Cojocaru, Yu. Rogozhin. *Modelling Inflections in Romanian Language by P Systems with String Replication.* Preproc. of the Tenth Workshop on Membrane Computing (WMC10), Curtea de Argeş, 2009, 116–128.

[2] E. Boian, S. Cojocaru. *The Inflexion Regularities for the Romanian Language.* Computer Science Journal of Moldova, 4, 1, 1996, 40–58.

[3] E. Boian, S. Cojocaru, L. Malahova. *Tools for Linguistic Applications* (Instruments pour Applications Linguistiques). in: La terminologie en Roumanie et en Republique de Moldova, Hors serie, N4, 2000, 42–44 (in French).

[4] E. Boian, A. Danilchenco, L. Topal. *The Automation of Speech Parts Inflexion Process.* Computer Science Journal of Moldova, 1(2), 1993, 14–26.

[5] S. Cojocaru. *Romanian Lexicon: Tools, Implementation, Utilization.* in: Language and Technology. (Lexicon român: instrumentar, implementare, utilizare. In: Limbaj şi tehnologie), Academia Română, Bucureşti, 1996, 37–40 (in Romanian).

[6] S.Cojocaru. *The Ascertainment of the Inflexion Models for Romanian.* Computer Science Journal of Moldova, 14, 1(40), 2006, 103–112.

[7] S. Cojocaru, M. Evstiunin, V. Ufnarovski. *Detecting and Correcting Spelling Errors for Romanian Language.* Computer Science Journal of Moldova, 1(1), 1993, 3–22.

[8] C. Coşman. Paradigmatic Morphology of Romanian language. *Environment of development – actualization. (Morfologia paradigmatică a limbii române. Mediu de dezvoltare-actualizare.* Teză de licenţă), Facultatea de Informatică, Universitatea "A.I.Cuza", Iaşi, 2002. (http://consilr.info.uaic.ro) (in Romanian).

[9] D. Cristea, C. Forăscu. *Linguistic Resources and Technologies for Romanian Language.* Computer Science Journal of Moldova, 14, 1(40), 2006, 34–73.

[10] R. Hausser. *Foundations of Computational Linguistics. Human-Computer Communication in Natural Language.* 2nd edition, revised and extended. Springer, 2001.

[11] T. Hristea, C. Moroianu. *Generation of Flexional Forms of Nouns and Adjective for Romanian Language (Generarea formelor flexionare substantivale şi adjectivale în limba română).* in: Building Awareness in Language Technology. F.Hristea, M.Popescu (eds.), Editura Universităţii din Bucureşti, 2003, 443–460 (in Romanian).

[12] D. Irimia. *The Grammar of Romanian Language (Gramatica limbii române).* Ed.II-a. Polirom, Bucureşti, 2004 (in Romanian).

[13] A. Lombard, C.Gâdei. *Morphological Romanian Dictionary (Dictionnaire morphologique de la langue roumaine).* Bucureşti, Editura Academiei, 1981 (in French).

[14] S. Marcus, Gh. Păun , C. Martín-Vide, *Contextual grammars as generative models of natural languages,* Computational Linguistics, v.24 n.2, June, 1998, 245–274.

[15] G. Bel Enguix, M. D. Jimenez Lopez. *Linguistic Membrane Systems and Applications.* in: Applications of Membrane Computing. G. Ciobanu, M. J.Pérez-Jiménez, Gh.Păun, (Eds.) 2006, 347–388.

[16] R.Gramatovici, G. Bel Enguix, *Parsing with P automata.* in: Applications of Membrane Computing. G. Ciobanu, M. J.Pérez-Jiménez, Gh.Păun, (Eds.) 2006, 389–436.

[17] Gh. Păun. *Membrane Computing: an Introduction.* Springer, 2002.

[18] L. Peev, L. Bibolar, E. Jodal. *A Formalization Model of Romanian Morphology.* in: Language and Technology (Un model de formalizare a morfologiei limbii române. în: *Limbaj şi Tehnologie.*) Editura Academiei Române, Bucureşti, 1996, 67–72 (in Romanian).

[19] L. Peev, F. Şerban. *Methods of Romanian Text Linguistic for Terminological Extraction.* In Tools and Resources. (Metode de analiză lingvistică a textelor în limba română pentru extragerea terminologică. Instrumente şi resurse.) - in http://dtil.unilat.org/seminar_bucuresti_2008/actes/peev_serban.htm (in Romanian)

[20] D. Tufiş. *Paradigmatic Morphology Learning.* Computers and Artificial intelligence 9(3), 1990, 273–290.

[21] D. Tufiş, A. M. Barbu, V. Pătraşcu, G. Rotariu, C. Popescu. *Corpora and Corpus-Based Morpho-Lexical Processing.* In: D.Tufiŝ, P.Andersen (eds.). Recent Advances in Romanian Language Technology, Editura Academie Române, Bucureşti, 1997, 115-128.

[22] D. Tufiş, L. Diaconu, C. Diaconu, A. M. Barbu. *Morphology of Romanian Language, a Reversible and Reusable Resource.* In: Language and Technology (Morfologia limbii române, o resursă lingvistică reversibilă şi reutilizabilă. În: *Limbaj şi Tehnologie*). Editura Academiei Române, Bucureşti, 1996, 59–65 (in Romanian).

[23] *The explanatory Romanian Dictionary (Dicţionarul explicativ al limbii române.)* Academia Română, Institutul de Lingvistică “Iorgu Iordan”, Editura Univers Enciclopedic, 1998 (in Romanian).

[24] *The Grammar of Romanian language (Gramatica limbii române)*, vol.I, Editura Academiei Republicii Populare Române, Bucureşti, 1963 (in Romanian).

[25] nl.ijs.si/ME/V3/msd/html/

[26] http://www.thefreedictionary.com/paradigm/

Artiom Alhazov[1,2], Elena Boian[1],
Svetlana Cojocaru[1], Yurii Rogozhin[1,3]

Received October 2, 2009

[1] Institute of Mathematics and Computer Science
Academy of Sciences of Moldova
Academiei 5, Chişinău MD-2028 Moldova
E–mail: {artiom,lena,sveta,rogozhin}@math.md

[2] IEC, Department of Information Engineering, Graduate School of Engineering
Hiroshima University, Higashi-Hiroshima 739-8527 Japan

[3] Research Group on Mathematical Linguistics, Rovira i Virgili University
Av. Catalunya, 35, Tarragona 43002 Spain

Correcting Inconsistency in Linear Inequalities by Minimal Change in the Right Hand Side Vector

Saeed Ketabchi Maziar Salahi

Abstract

Correcting an inconsistent set of linear inequalities by minimal changes in problem data is a well studied problem and up to now several algorithms have been developed to do this task. In this paper, we consider doing the minimal correction using the l_2 norm by changing just the right hand vector. A new formulation of the problem is introduced and its relation with the normal solution of the alternative system of the original system is given. Then a generalized Newton algorithm is designed to solve the new formulation. Extensive computational results using this algorithm and conjugate gradient method is reported to demonstrate the advantages and disadvantages of the two algorithms.

Keywords: Linear Inequalities, Convex Optimization, Conjugate Gradient Method, Generalized Newton Method, Barrier Method.

1 Introduction

In this paper we consider the following set of linear inequalities that are inconsistent:

$$Ax \leq b, \tag{1}$$

where $A \in R^{m\times n}$ and $b \in R^m$. In other words, there is no $x \in R^n$ for which (1) is feasible. The inconsistency in system (1) might be due to the various reasons, such as lack of interaction between different groups who are defining the constraints, wrong or inaccurate estimates, error in data, over optimistic goals, and many others. Correcting system (1) to a feasible system by minimal changes in its data have been known for long time and up to now several algorithms have been developed to do it [1, 2]. A very simple approach to form a feasible system from (1) is to consider changes just in the right hand side vector b, which is usually called the resources vector, using l_1 norm i.e.,

$$\begin{aligned} &\min \sum_{i=1}^{m} |r_i| \\ &Ax \leq b + r. \end{aligned} \tag{2}$$

As we know, this problem is easily convertable to an linear programming (LP) problem which is efficiently solvable by either the *Simplex or Interior Point Methods* [3, 5]. It is also worth to note that one may consider the infinity norm in the objective function which results to:

$$\begin{aligned} &\min \ \|r\|_\infty \\ &Ax \leq b + r. \end{aligned} \tag{3}$$

This still is equivalent to an LP problem. In the next section we discuss the minimal correction using the l_2 norm. An equivalent formulation of the problem is given and two efficient algorithms are designed to solve the new formulation.

2 2-Norm Corrections

The minimal correction using the l_2 norm by changing the right hand side vector is:

$$\begin{aligned} &\min_{x,r} \frac{1}{2} \|r\|^2 \\ &Ax \leq b + r. \end{aligned} \tag{4}$$

In the following theorem we show how we compute optimal x and r values.

Theorem 2.1. *Let x^* and r^* be the optimal solution of (4). Then $r^* = (Ax^* - b)_+$, where $a_+ = \max(a, 0)$ and x^* is an optimal solution of*

$$\min_x \frac{1}{2} \left\| (Ax - b)_+ \right\|^2 . \tag{5}$$

Proof. Let us write (4) as:

$$\begin{aligned} \min_x \min_r \frac{1}{2} \|r\|^2 \\ Ax \le b + r. \end{aligned} \tag{6}$$

Now for a given $x \in R^n$, let us first consider the inner minimization problem i.e.,

$$\begin{aligned} \min_r \frac{1}{2} \|r\|^2 \\ Ax \le b + r. \end{aligned} \tag{7}$$

It is obvious that problem (7) is a convex minimization problem, therefore the KKT conditions are necessary and sufficient for optimality and are given by:

$$\begin{aligned} & r - \lambda = 0, \\ & Ax \le b + r, \\ & \lambda^T (Ax - b - r) = 0, \\ & \lambda \ge 0, \end{aligned}$$

where the vector λ denotes the lagrange multipliers. From the first equation one has $r = \lambda$. Now if $\lambda_i \neq 0$ for some i, then from the third equation $(Ax - b)_i = r_i = \lambda_i$. However, when $\lambda_i = 0$, from the first equation one has $r_i = 0$. All these together imply that $r = (Ax - b)_+$. Therefore, we can write problem (4) as

$$\min_x \frac{1}{2} \left\| (Ax - b)_+ \right\|^2 .$$

This completes the proof. □

It is worth mentioning that (5) is the dual of the following optimization problem:

$$\begin{aligned} \max \quad -b^T u - \frac{1}{2}\|u\|^2 \\ A^T u = 0, \\ u \geq 0. \end{aligned} \tag{8}$$

In the following corollary we give an optimal solution of (8) using an optimal solution of (5).

Corollary 2.2. *Let x^* be an optimal solution of problem (5). Then $u^* = (Ax^* - b)_+$ is an optimal solution of (8).*

Proof. Let x^* be an optimal solution of (5). It is obvious that for $u^* = (Ax^* - b)_+$, $A^T u^* = 0$, which is the optimality condition for (5) and $u \geq 0$. Now we further show that the objective values of (5) and (8) are equal i.e.,

$$-b^T(Ax^* - b)_+ = \|(Ax^* - b)_+\|^2 .$$

Since $A^T(Ax^* - b)_+ = 0$, therefore, $-b^T(Ax^* - b)_+ = (Ax^* - b)^T(Ax^* - b)_+ = (Ax^* - b)_+^T(Ax^* - b)_+ = \|(Ax^* - b)_+\|^2 .$ □

In the sequel we further show that using an optimal solution of (5) we can construct an optimal solution for

$$\begin{aligned} \min \quad \frac{1}{2}\|u\|^2 \\ A^T u = 0, \\ b^T u = -\rho, \\ u \geq 0, \end{aligned} \tag{9}$$

where ρ is an arbitrary strictly positive parameter.

Remark 2.3. *It is worth to note that the constraints of (9) are the alternative system of (1).*

Corollary 2.4. *Let x^* be an optimal solution of problem (5), then $u^* = -\frac{\rho(Ax^*-b)_+}{\|(Ax^*-b)_+\|^2}$ is the normal solution of*

$$A^T u = 0, \quad b^T u = -\rho, \; u \geq 0$$

namely a solution of (9).

Proof. The Lagrangian dual of (9) is

$$\max_{\lambda,\mu} \; -\frac{1}{2}\left\|(A\lambda - b\mu)_+\right\|^2 + \mu\rho, \tag{10}$$

where $u = (A\lambda - b\mu)_+$. At optimality the objective values of (9) and (10) should be equal. This implies that

$$\|(A\lambda^* - b\mu^*)_+\|^2 = \rho\mu^*.$$

From this we further can deduce that $\mu > 0$, then

$$\mu^* \left\| \left(A \left(\frac{\lambda^*}{\mu^*} \right) - b \right)_+ \right\| = \rho.$$

By further defining $x = \frac{\lambda^*}{\mu^*}$ we have $\mu^* = \frac{\rho}{\|(Ax-b)_+\|}$. Now let x^* be the optimal solution of (5) and also let $\mu^* = \frac{\rho}{\|(Ax^*-b)_+\|}$. This implies that $\lambda^* = \mu^* x^*$. Now for this choice of variables the two objective values are equal. Thus we have the optimal solutions of both problems. □

To solve (5) we use conjugate gradient algorithm and the so called generalized Newton algorithm that is discussed in the sequel. As it is obvious, the objective function of (5) is a convex function, but it just has the first derivative not the second one [6]. However, the generalized hessian is defined for this function that follows:

$$\nabla f(x) = A^T(Ax - b)_+$$

and

$$\nabla^2 f(x) = A^T D A,$$

where D is an $n \times n$ diagonal matrix for which $D(i,i) = 1$ when $(Ax - b)_i > 0$, $D(i,i) = 0$, when $(Ax-b)_i < 0$, and in $[0,1]$ when $(Ax-b)_i = 0$.

Obviously the generalized Hessian is a set and for simplicity in this article we consider a specific element of this set, namely $D(i,i) = 0$ when $(Ax - b)_i = 0$. Now the generalized Newton algorithm can be outlined as follows:

Generalized Newton Algorithm

- **Inputs:** An accuracy parameter $\epsilon > 0$, a regularization parameter[1], $\delta = 10^{-4}$ and a starting point $x_0 \in R^n$.
- i=0;
- **While** $\|\nabla f(x_i)\|_\infty \geq \epsilon$.
- $x_{i+1} = x_i - (\nabla^2 f(x_i) + \delta I)^{-1} \nabla f(x_i)$.
- i=i+1.
- **End.**

Remark 2.5. *It is worth to note that one may use line search techniques such as Armijo or Wolf in the structure of the algorithm. Moreover the finite global convergence of generalized Newton algorithm with Armijo line search is proved in [6].*

3 Linear Inequalities with Nonnegativity Constraints

In this section we consider the set of linear inequalities (1) by adding extra nonnegativity constraint to them i.e.,

$$\begin{aligned} Ax &\leq b \\ x &\geq 0. \end{aligned} \tag{11}$$

It is obvious that one can consider (11) as a special case of (1), but due to its special structure it is reasonable to do the correction of this sort

[1] It guarantees the nonsingularity of hessian matrix.

of inconsistent set of linear inequalities specifically. Here we consider the case where the correction is done by just correcting the right hand side of the first set of inequalities not the $x \geq 0$ i.e.,

$$\begin{aligned} \min \quad & \frac{1}{2}\|r\|^2 \\ & Ax \leq b + r \\ & x \geq 0. \end{aligned} \tag{12}$$

In the following theorem we show how one can compute optimal x and r values.

Theorem 3.1. *Let x^* and r^* be optimal solutions of (12). Then $r^* = (Ax^* - b)_+$, where x^* is an optimal solution of*

$$\min_{x \geq 0} \quad \frac{1}{2}\|(Ax - b)_+\|^2 . \tag{13}$$

Proof. Similar to the proof of Theorem 2.1. □

As it is obvious the only difference between (13) and (5) is the nonnegativity constraint and it makes the problem a constraint optimization problem. To solve (13) we use the logarithmic barrier [7] approach by bringing the $x \geq 0$ to the objective functions as:

$$\min_{x} \quad \frac{1}{2}\|(Ax - b)_+\|^2 - \mu \sum_{i=1}^{n} \log(x_i), \tag{14}$$

where μ is the barrier parameter. Then we apply the generalized Newton method by starting from a strictly positive vector x and $\mu_0 = 1$. The logarithmic term does not allow the components of variable x to get negative and the value of μ approaches to zero during the iterations of the algorithm, for example $\mu_{k+1} = 0.8\mu_k$. Another approach which one might consider to solve (13) is the penalty function method as:

$$\min_{x} \quad \frac{1}{2}\|(Ax - b)_+\|^2 + \frac{1}{2}M\|(-x)_+\|^2 , \tag{15}$$

where M is a very big number, for example 10^{10}. This does not allow to have big $\|(-x)_+\|^2$. It is worth mentioning that vector x might have very small negative values in the optimal solution which can be rounded to zero.

4 Computational Results

In this section we present numerical results for the generalized Newton and conjugate gradient algorithms on various randomly generated problems. Test problems are generated using the following MATLAB code:

MATLAB random insolvable linear inequalities generator

```
% Generates random inconsistent system $Ax <= b$;
% Input:m,n,d(density); Output:$A \in R^{m\times n}, b \in R^m$;
pl=inline('(abs(x)+x)/2');%pl(us) function;
m=input('enter m= '); n=input('enter n= '); d=input('enter
d= ');
m1=max(m-round(0.5*m),m-n);
A1=sprand(m1,n,d);A1=1*(A1-0.5*spones(A1));
x=spdiags(rand(n,1),0,n,n)*1*(rand(n,1)-rand(n,1));
x=spdiags(ones(n,1)-sign(x),0,n,n)*10*(rand(n,1)-rand(n,
1));
m2=m-m1;u=randperm(m2);A2=A1(u,:);
b1=A1*x+spdiags((rand(m1,1)),0,m1,m1)*1*ones(m1,1);
b2=b1(u)+spdiags((rand(m2,1)),0,m2,m2)*10*ones(m2,1);
A=100*[A1;-A2]; b=[b1;-b2];
```

In Tables 1 and 2 we present comparison between the gradient based algorithm (**GR**) and our new generalized Newton algorithm (**GNewton**) with Armijo linesearch for various randomly generated problems with different densities. Our numerical experiments show that the generalized Newton method finds an optimal solution much faster than the gradient based algorithm for majority of problems and for all problems the optimal objective values are much smaller than the gradient algorithm. It is worth mentioning that we run both algorithms for at most 500 seconds with the tolerance equal to 10^{-5} for gradient algorithm and 10^{-8} for the generalized Newton method.

In Tables 3 and 4 we report numerical experiments on various randomly generated problems with different densities for inconsistent linear inequalities that involve nonnegativity of variables. To solve these

Table 1. **Comparison of Generalized Newton method and Gradient algorithm with Armijo linesearch**

m, n, d		$\|(Ax^* - b)_+\|$	$\|x^*\|$	$\|\nabla f(x^*)\|_\infty$	time(sec)
$2 \times 10^6, 10^3, 0.001$	GNewton	1.9416×10^2	1.7966	1.6584×10^{-10}	33
	GR	1.9450×10^2	1.7975	6.7686×10^2	500
$5 \times 10^6, 2000, 0.001$	GNewton	2.7879×10^2	2.5694	1.7558×10^{-11}	134
	GR	1.2976×10^4	9.5597	5.1477×10^5	500
$2000, 1500, 0.01$	GNewton	1.2657×10^2	2.2207×10^2	6.442×10^{-7}	10
	GR	1.4245×10^2	22.236	3.1181×10^2	4
$3000, 2000, 0.01$	GNewton	1.613×10^2	2.2550×10^2	2.1859×10^{-7}	23
	GR	1.8045×10^2	22.855	7.9328×10^2	6
$5000, 2500, 0.01$	GNewton	2.0322×10^2	5.0537×10	2.1461×10^{-9}	75
	GR	2.6368×10^2	13.356	6.7538×10^2	8
$7000, 3000, 0.01$	GNewton	2.4224×10^2	4.3524	1.4739×10^{-8}	42
	GR	2.7263×10^2	6.8396	5.5115×10^2	10
$10000, 5000, 0.01$	GNewton	2.8706×10^2	1.1382×10^2	2.4625×10^{-9}	571
	GR	3.4522×10^2	14.383	4.7877×10^2	19
$10^5, 500, 0.01$	GNewton	1.3636×10^2	1.2889	1.4504×10^{-10}	3
	GR	1.3636×10^2	1.2905	2.2150×10^2	71
$10^6, 1000, 0.01$	GNewton	1.9512×10^2	1.7957	1.3308×10^{-11}	64
	GR	4.9737×10^2	1.8019	4.7845×10^4	501

Table 2. **Comparison of Generalized Newton method and Gradient algorithm with Armijo line-search**

m, n, d		$\|(Ax^* - b)_+\|$	$\|x^*\|$	$\|\nabla f(x^*)\|_\infty$	time(sec)
$1000, 500, 0.1$	GNewton	9.1251×10	1.3206×10	3.0594×10^{-9}	1
	GR	1.1227×10^2	5.6497	3.1439×10^2	5
$2000, 1500, 0.1$	GNewton	1.3063×10^2	2.2709×10^2	1.8781×10^{-5}	7
	GR	1.3138×10^2	22.788	2.7351×10^2	13
$3000, 2000, 0.1$	GNewton	1.5668×10^2	2.1775×10^2	2.044×10^{-5}	14
	GR	1.5736×10^2	21.884	2.4234×10^2	17
$4000, 3000, 0.1$	GNewton	1.8075×10^2	3.2590×10^2	7.8024×10^{-8}	163
	GR	1.8134×10^2	32.697	2.1083×10^2	29
$6000, 4000, 0.1$	GNewton	2.2436×10^2	3.1374×10^2	1.5982×10^{-5}	91
	GR	2.4760×10^2	31.531	9.1938×10^3	55
$5000, 1000, 1$	GNewton	1.6045×10^2	1.8174	8.3030×10^{-9}	25
	GR	1.8707×10^2	1.8241	1.7451×10^4	72
$5000, 3000, 1$	GNewton	1.7489×10^2	1.8673	1.0023×10^{-9}	49
	GR	1.9926×10^2	1.8644	1.8771×10^4	149
$10000, 1000, 1$	GNewton	2.0643×10^2	2.2497×10^2	9.8919×10^{-6}	132
	GR	1.4414×10^3	23.696	3.0855×10^4	216

Table 3. **Comparison between Generalized Newton method and Barrier approach**

m, n, d		$\|(Ax^* - b)_+\|$	$\|x^*\|$	$\|\nabla f(x^*)\|_\infty$	time(sec)
$2 \times 10^6, 10^3, 0.001$	GNewton	1.0186×10^3	1.2894	1.5717×10^5	20
	Barrier	3.1282×10^3	4.1182	1.5370×10^5	503
$5 \times 10^6, 2000, 0.001$	GNewton	2.3461×10^3	1.7933	3.7410×10^5	97
	Barrier	1.0125×10^4	7.7755	3.6690×10^5	511
$2000, 1500, 0.01$	GNewton	1.6669×10^2	4.6664	1.4246×10^3	10
	Barrier	1.6669×10^2	4.6664	1.12670×10^{-2}	503
$3000, 2000, 0.01$	GNewton	2.5783×10^2	3.7585	3.3761×10^3	19
	Barrier	2.8079×10^2	4.1085	3.9359×10^3	513
$5000, 2500, 0.01$	GNewton	3.5457×10^2	3.1902	4.4665×10^3	28
	Barrier	1.6086×10^3	11.942	2.3028×10^4	520
$7000, 3000, 0.01$	GNewton	4.5226×10^2	3.1648	8.3185×10^3	43
	Barrier	2.4054×10^3	14.101	43.4104×10^4	514
$10000, 5000, 0.01$	GNewton	6.1538×10^2	4.3014	8.8359×10^3	211
	Barrier	1.9856×10^3	10.725	2.9873×10^4	519
$10^5, 500, 0.01$	GNewton	5.8234×10^2	0.83951	7.1131×10^4	3
	Barrier	1.07228×10^3	1.9616	6.8137×10^4	105
$10^6, 1000, 0.01$	GNewton	2.4497×10^3	1.2288	1.3796×10^5	51
	Barrier	9.7547×10^3	5.1913	7.3701×10^5	509

Table 4. **Comparison between Generalized Newton method and Barrier approach**

m, n, d		$\|(Ax^* - b)_+\|$	$\|x^*\|$	$\|\nabla f(x^*)\|_\infty$	time(sec)
$1000, 500, 0.1$	GNewton	1.7786×10^2	1.4306	7.4710×10^3	1.2
	Barrier	1.9887×10^2	1.5843	7.5234×10^3	66
$2000, 1500, 0.1$	GNewton	3.2366×10^2	3.2806	1.0015×10^4	21
	Barrier	3.4735×10^2	3.3738	1.1146×10^4	500
$3000, 2000, 0.1$	GNewton	4.2772×10^2	3.0036	1.9189×10^4	45
	Barrier	7.0578×10^2	3.7875	2.5369×10^4	500
$4000, 3000, 0.1$	GNewton	5.2605×10^2	4.9115	1.8869×10^4	150
	Barrier	1.3361×10^3	7.4745	4.8884×10^4	512
$6000, 4000, 0.1$	GNewton	8.5661×10^2	431.15	3.4293×10^4	282
	Barrier	2.9507×10^4	87.763	1.0627×10^6	540
$5000, 1000, 1$	GNewton	1.3920×10^3	1.4183	2.4942×10^5	26
	Barrier	4.6499×10^3	4.3363	5.1778×10^5	505
$5000, 3000, 1$	GNewton	1.6982×10^3	3.4684	1.7544×10^5	330
	Barrier	5.7628×10^3	6.8423	631×10^5	561
$10000, 1000, 1$	GNewton	1.9394×10^3	1.4330	4.7339×10^5	50
	Barrier	7.0689×10^3	4.8063	8.5838×10^5	508

problems we have employed the generalized Newton (**GNewton**) and barrier methods (**Barrier**). In the optimal solution obtained by the generalized Newton method we might have very small components of x that are negative. In this case we rounded them to zero and this is the reason for having norm infinity of the gradient vector far away than zero, however at optimality it is usually around $O(10^{-8})$. On the other hand this shows the sensitivity of these problems to very small changes in the optimal solution. As we observe from these tables, the generalized Newton method beats the barrier approach both in time and quality of solution for all of the problems.

5 Conclusion

In this paper we have furthermore investigated how to correct an inconsistent set of linear inequalities by minimal changes in its data. A new formulation of the original model is given and its relation to normal solution of alternative system for original system is discussed. Then we have presented a generalized Newton based algorithm to solve the new formulation. We also discussed inconsistent set of inequalities that involve nonnegativity of variables. To solve this specific case we have utilized the generalized Newton method and barrier approach. At last, our computational experiments on several randomly generated problems show the superior performance of the generalized Newton algorithm to the classical gradient based algorithm and barrier approach.

References

[1] P. Amaral, M.W. Terosset, and P. Barahona. *Correcting an inconsistent set of linear inequalities by nonlinear programming.* Technical Report pp. 1–27, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005, 2000.

[2] P. Amaral and P. Barahona. *A framework for optimal correction of inconsistent linear constraint.* Constraints, vol. 10, no. 1 (2005), pp. 67–86.

[3] E.D. Andersen and K.D. Andersen. *The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm.* In: H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, Kluwer Academic Publishers, (2000), pp. 197–232.

[4] M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali. *Linear Programming and Network Flows.* John Wiely and Sons, 1990.

[5] *CPLEX: ILOG Optimization.* `http://www.ilog.com.`

[6] O. Mangasarian. *A Newton method for linear programming.* Journal of Optimization Theory and Applications, vol. 121, no. 1 (2004), pp. 1–18.

[7] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer Science, 1999.

Saeed Ketabchi, Maziar Salahi,

Received November 3, 2009

Saeed Ketabchi,
Department of Mathematics,
Faculty of Sciences,
University of Guilan,
Rasht, Namjoo Street, Guilan, P.O.Box 1914
sketabchi@guilan.ac.ir

Maziar Salahi
Department of Mathematics,
Faculty of Sciences,
University of Guilan,
Rasht, Namjoo Street, Guilan, P.O.Box 1914
salahim@guilan.ac.ir

Computer Science Journal of Moldova, vol.**17**, no.2(50), 2009

Quasigroups in cryptology

V.A. Shcherbacov

Abstract

We give a review of some known published applications of quasigroups in cryptology.

Keywords: cryptology, quasigroup, (r,s,t)-quasigroup, stream-cipher, secret-sharing system, zero knowledge protocol, authentication of a message, NLPN sequence, Hamming distance.

Contents

1 Introduction

Now the theory of quasigroups applications in cryptology goes through the period of rapid enough growth. Therefore any review of results in the given area of researches quite quickly becomes outdated. Here we give a re-written and supplemented form of more early versions [111, 112] of such kind of reviews. See also [55, 123].

Almost all results obtained in the domain of quasigroups application in cryptology and coding theory till the end of eighties years of the XX-th century are described in [25, 26, 28]. In the present survey the main attention is devoted to the later articles in this direction.

It is possible to find basic facts on quasigroup theory in [6, 8, 7, 102, 83, 111]. Information on basic fact in cryptology can be found in many books, see, for example, [3, 13, 95, 96].

Cryptology is a science that consists of two parts: cryptography and cryptanalysis. Cryptography is a science on methods of transformation (ciphering) of information with the purpose of this information protection from an unlawful user. Cryptanalysis is a science on methods and ways of breaking down the ciphers [37].

In some sense cryptography is a "defense", i.e. this is a science on construction of new ciphers, but cryptanalysis is an "attack", i.e. this is a science and some kind of "art", a set of methods on breaking the ciphers. This situation is similar to situation with intelligence and contr-intelligence.

These two objects (cryptography and cryptanalysis) are very close and there does not exist a good cryptographer that does not know methods of cryptanalysis.

It is clear, that cryptology depends on level of development of society, of science and level of technology development.

We recall, a cipher is a way (a method, an algorithm) of information transformation with the purpose of its defense. A key is some hidden part (usually, a little one) or parameter of a cipher.

Steganography is a set of means and methods of hiding the fact of sending (or passing) the information, for example, a communication or a letter. Now there exist methods of hiddenness of the fact of information sending by usual post, by e-mail and so on.

In this survey as Coding Theory (Code Theory) will be meant a science on defense of information from accidental errors caused by transformation and sending (passing) this information.

When sending the important and confidential information, as it seems to us, there exists a sense to use methods of Code Theory, Cryptology, and Steganography all together [80].

In cryptology one often uses the following Kerkhoff's (1835 - 1903) rule: an opponent (an unlawful user) knows all ciphering procedure (sometimes a part of plaintext or ciphertext) with exception of key.

Many authors of books, devoted to cryptology divide this science (sometimes not paying attention to this fact) in two parts: before article of Diffie and Hellman [30] (so-called cryptology with non-public (symmetric) key) and after this work (a cryptology with public or non-symmetric key). Practically namely Diffie and Hellman article opened new era in cryptology. Moreover, it is possible to apply these new approaches in practice.

Especially fast development of the second part of cryptology is connected with very fast development of Personal Computers and Nets of Personal Computers, other electronic technical devices in the end of XX-th century. Many new mathematical, cryptographical problems appeared in this direction and some of them are not solved. Solving of these problems have big importance for practice.

Almost all known construction of error detecting and error correcting codes, cryptographic algorithms and enciphering systems have made use of associative algebraic structures such as groups and fields, see, for example, [84, 21].

There exists a possibility to use such non-associative structures as quasigroups and neo-fields in almost all branches of coding theory, and especially in cryptology.

Often the codes and ciphers based on non-associative systems show better possibilities than known codes and ciphers based on associative systems [28, 78].

Notice that in the last years the quantum code theory and quantum cryptology [114, 47, 124, 14] have been developed intensively. Quantum cryptology also use theoretical achievements of "usual" cryptology [12].

Efficacy of applications of quasigroups in cryptology is based on the fact that quasigroups are "generalized permutations" of some kind and the number of quasigroups of order n is larger than $n! \cdot (n-1)! \cdot \ldots \cdot 2! \cdot 1!$ [25].

It is worth noting that several of the early professional cryptographers, in particular, A.A. Albert, A. Drisko, M.M. Glukhov, J.B. Rosser, E. Schönhardt, C.I. Mendelson, R. Schaufler were connected with the development of Quasigroup Theory. The main known "applicants" of quasigroups in cryptology were (and are) J. Denes and A.D. Keedwell [22, 25, 26, 28, 23].

Of course, one of the most effective cipher methods is to use unknown, non-standard or very rare language. Probably the best enciphering method was (and is) to have a good agent.

2 Quasigroups in "classical" cryptology

There exist two main elementary methods when ciphering the information.

(i). Symbols in a plaintext (or in its piece (its bit)) are permuted by some law. The first known cipher of such kind is cipher "Scital" (Sparta, 2500 years ago).

(ii). All symbols in a fixed alphabet are changed by a law on other letters of this alphabet. One of the first ciphers of such kind was Cezar's cipher ($x \to x + 3$ for any letter of Latin alphabet, for example $a \to d, b \to e$ and so on).

In many contemporary ciphers (DES, Russian GOST, Blowfish [95, 31]) the methods (i) and (ii) are used with some modifications.

Trithemius cipher makes use of 26×26 square array containing 26 letters of alphabet (assuming that the language is English) arranged in a Latin square. Different rows of this square array are used for enciphering various letters of the plaintext in a manner prescribed by the keyword or key-phrase [3, 65]. Since a Latin square is the multiplication table of a quasigroup, this may be regarded as the earliest use of a non-associative algebraic structure in cryptology. There exists a possibility to develop this direction using quasigroup approach, in particular, using orthogonal systems of binary or n-ary quasigroups.

R. Schaufler in his Ph.D. dissertation discussed the minimum amount of plaintext and corresponding ciphertext which would be required to break the Vigenere cipher (a modification of Trithemius cipher) [106]. That is, he considered the minimum member of entries of particular Latin square which would determine the square completely.

Recently this problem has re-arisen as the problem of determining of so-called critical sets in Latin squares, see [67, 32, 33, 36, 35, 69]. See, also, articles, devoted to Latin trades, for example, [5].

More recent enciphering systems which may be regarded as extension of Vigenere's idea are mechanical machines such as Jefferson's wheel and the M-209 Converter (used by U.S.Army until the early 1950's) and the electronically produced stream ciphers of the present day [77, 95].

During the second World War R.Shauffler while working for the German Cryptography service, developed a method of error detection based on the use of generalized identities (as they were later called by V.D. Belousov) in which the check digits are calculated by means of an associative system of quasigroups (see also [19]). He pointed out that the resulting message would be more difficult to decode by unauthorized receiver than in the case when a single associative operation is used for calculation [107].

Therefore it is possible to assume that information on systems of quasigroups with generalized identities (see, for example, works of Yu. Movsisyan [97] may be applied in cryptography of the present day.

Definition 2.1. *A bijective mapping $\varphi : g \rightarrowtail \varphi(g)$ of a finite group $(G, \cdot)$ onto itself is called an orthomorphism if the mapping $\theta : g \rightarrowtail \theta(g)$ where $\theta(g) = g^{-1}\varphi(g)$ is again a bijective mapping of G onto itself. The orthomorphism is said to be in canonical form if $\varphi(1) = 1$ where 1 is the identity element of $(G, \cdot)$.*

A direct application of orthomorphisms to cryptography is described in [92, 91].

3 Quasigroup-based stream ciphers

"Stream ciphers are an important class of encryption algorithms. They encrypt individual characters (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time.

By contrast, block ciphers tend to simultaneously encrypt groups of characters of a plaintext message using a fixed encryption transformation. Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry.

They are also more appropriate, and in some cases mandatory (e.g., in some telecommunications applications), when buffering is limited or when characters must be individually processed as they are received. Because they have limited or no error propagation, stream ciphers may also be advantageous in situations where transmission errors are highly probable" [90].

Often for ciphering a block (a letter) B_i of a plaintext the previous ciphered block C_{i-1} is used. Notice that Horst Feistel was one of the first who proposed such method of encryption (Feistel net) [51].

In [77] (see also [78, 79]) C. Koscielny has shown how quasigroups/neofields-based stream ciphers may be produced which are both more efficient and more secure than those based on groups/fields.

In [100, 87] it is proposed to use quasigroups for secure encoding.

A quasigroup $(Q, \cdot)$ and its (23)-parastrophe $(Q, \backslash)$ satisfy the following identities $x\backslash(x \cdot y) = y,\ x \cdot (x\backslash y) = y$. The authors propose to use this property of the quasigroups to construct a stream cipher.

Algorithm 3.1. *Let A be a non-empty alphabet, k be a natural number, $u_i, v_i \in A$, $i \in \{1, ..., k\}$. Define a quasigroup $(A, \cdot)$. It is clear that the quasigroup $(A, \backslash)$ is defined in a unique way. Take a fixed element l ($l \in A$), which is called a leader.*

Let $u_1u_2...u_k$ be a k-tuple of letters from A. The authors propose the following ciphering procedure $v_1 = l \cdot u_1, v_i = v_{i-1} \cdot u_i$, $i = 2, ..., k$. Therefore we obtain the following cipher-text $v_1v_2 \dots v_k$.

The enciphering algorithm is constructed in the following way: $u_1 = l\backslash v_1, u_i = v_{i-1}\backslash v_i, i = 2, ..., k$.

The authors claim that this cipher is resistant to the brute force attack (exhaustive search) and to the statistical attack (in many languages some letters meet more frequently, than other ones).

Example 3.1. Let alphabet A consists from the letters a, b, c. Take the quasigroup $(A, \cdot)$:

$\cdot$	a	b	c
a	b	c	a
b	c	a	b
c	a	b	c

Then $(A, \backslash)$ has the following Cayley table

$\backslash$	a	b	c
a	c	a	b
b	b	c	a
c	a	b	c

Let $l = a$ and open text is $u = b\,b\,c\,a\,a\,c\,b\,a$. Then the cipher text is $v = c\,b\,b\,c\,a\,a\,c\,a$. Applying the decoding function on v we get $b\,b\,c\,a\,a\,c\,b\,a = u$.

Probably the cipher which is described here (Algorithm 3.1) and its generalizations are now the most known and the most used quasigroup based stream-ciphers.

Authors [100] say that this cipher is resistant to the brute force attack and to the statistical one.

Cryptanalyses of Algorithm 3.1 was made by M. Vojvoda [122]. He showed that this cipher is not resistant relatively to chosen ciphertext attack, chosen plaintext attack and ciphertext-only attack.

We give the following 3-ary modification of Algorithm 3.1 [101]. The possibility of such modification of Algorithm 3.1 was observed in [111].

Algorithm 3.2. *Let A be a non-empty alphabet, k be a natural number, $u_i, v_i \in A$, $i \in \{1, ..., k\}$. Define a 3-ary quasigroup (A, β). It is clear that this quasigroup defines $(4! - 1)$ parastrophes including (14)-, (24)- and (34)-parastrophe.*

Take the fixed elements l_1, l_2, l_3, l_4 ($l_i \in A$), which are called leaders.

Let $u_1 u_2 ... u_k$ be a k-tuple of letters from A. The author proposes the following ciphering procedure $v_1 = \beta(u_1, l_1, l_2), v_2 = \beta(u_2, l_3, l_4), v_i = \beta(u_i, v_{i-2}, v_{i-1}), i = 3, 4, ..., k-1$. Therefore we obtain the following cipher-text $v_1 v_2 ... v_k$.

The enciphering algorithm is constructed in the following way: $u_1 = {}^{(14)}\beta(v_1, l_1, l_2), u_2 = {}^{(14)}\beta(v_2, l_3, l_4), u_i = {}^{(14)}\beta(v_i, v_{i-2}, v_{i-1}), i = 3, 4, ..., k-1$.

In [101] also variants of Algorithm 3.2 are given using (24)- and (34)-parastrophes of a ternary quasigroup.

Further development of Algorithm 3.1 is presented in [54].

Definition 3.1. *Let r be a positive integer. let $(Q, *)$ be a quasigroup and $a_j, b_j \in Q$. For each fixed $m \in Q$ define first the transformation $Q_m : Q^r \longrightarrow Q^r$ by*

$$Q_m(a_0, a_1, \dots, a_{r-1}) = (b_0, b_1, \dots, b_{r-1}) \Longleftrightarrow$$

$$b_i = \begin{cases} m * a_0; & i = 0 \\ b_{i-1} * a_i; & 1 \le i \le (r-1). \end{cases}$$

Then define $\mathcal{R}_1$ as composition of transformations of kind Q_m, for suitable choices of the indexes m, as follows

$$\mathcal{R}_1(a_0, a_1, \dots, a_{r-1}) = Q_{a_0}(Q_{a_1} \dots (Q_{a_{r-1}}(a_0, a_1, \dots, a_{r-1}))).$$

Definition 3.2. *[54] (Shapeless quasigroup) A quasigroup $(Q, *)$ of order n is said to be shapeless if it is non-commutative, non-associative, it does not have neither left nor right unit, it does not contain proper subquasigroups, and there is no $k < 2n$ for which are satisfied the identities of the kinds:*

$$\underbrace{x * (x \dots x * (x(x}_{k} * y)) = y; \; y = ((y * \underbrace{x) * \dots) * x) * x}_{k} \qquad (1)$$

Remark 3.1. Condition $k < 2n$ for identities (1) means that any left and right translation of quasigroup $(Q, *)$ should have the order $k \geq (2n + 1)$.

In [54] it is proposed to construct shapeless quasigroups using transversal approach [58]. Simple quasigroups without subquasigroups and with identity automorphism group are studied in [82, 75, 64, 110].

In the article [53] it is proposed a block cipher based on Algorithm 3.1. Let $(Q, *)$ be a quasigroup of finite order 2^d. Using the operation $*$ authors define the following vector valued Boolean function (v.v.b.f.) $a * b = c \Leftrightarrow *_{vv}(x_1, x_2, ..., x_d, y_1, y_2, ..., y_d) = (z_1, z_2, ..., z_d)$, where $x_1...x_d, y_1...y_d, z_1...z_d$ are binary representations of a, b, c respectively.

Each element z_i depends on the bits $x_1, x_2, ..., x_d, y_1, y_2, ..., y_d$ and is uniquely determined by them. So, each z_i can be seen as a $2d$-ary Boolean function $z_i = f_i(x_1, x_2, ..., x_d, y_1, y_2, ..., y_d)$, where $f_i : \{0,1\}^{2d} \to \{0,1\}$ strictly depends on, and is uniquely determined by $*$.

Authors state that for every quasigroup $(Q, *)$ of order 2^d and for each bijection $Q \to \{0, 1..., 2^d - 1\}$ there are a uniquely determined v.v.b.f. $*_{vv}$ and d uniquely determined $2d$-ary Boolean functions $f_1, f_2, ..., f_d$ such that for each $a, b, c \in Q$

$$\begin{aligned}&a * b = c \Leftrightarrow *_{vv}(x_1, ..., x_d, y_1, ..., y_d) = \\ &(f_1(x_1, ..., x_d, y_1, ..., y_d), ..., f_d(x_1, ..., x_d, y_1, ..., y_d)).\end{aligned}$$

Each k-ary Boolean function $f(x_1, ..., x_k)$ can be represented in a unique way by its algebraic normal form (ANF), i.e., as a sum of products

$$ANF(f) = \alpha_0 + \sum_{i=1}^{k} \alpha_i x_i + \sum_{1 \leq i \leq j \leq k}^{k} \alpha_{i,j} x_i x_j + \sum_{1 \leq i \leq j \leq s \leq k}^{k} \alpha_{i,j,s} x_i x_j x_s + ...,$$

where the coefficients $\alpha_0, \alpha_i, \alpha_{i,j}, ...$ are in the set $\{0,1\}$ and the addition and multiplication are in the field $GF(2)$.

The ANFs of the functions f_i give information about the complexity of the quasigroup (Q, .) via the degrees of the Boolean functions f_i. The degrees of the polynomials $ANF(f_i)$ rise with the order of the quasigroup. In general, for a randomly generated quasigroup of order 2^d, $d \geq 4$, the degrees are higher than 2.

Definition 3.3. *A quasigroup $(Q,*)$ of order 2^d is called Multivariate Quadratic Quasigroup (MQQ) of type $Quad_{d-k}Lin_k$ if exactly $d-k$ of the polynomials f_i are of degree 2 (i.e., are quadratic) and k of them are of degree 1 (i.e., are linear), where $0 \leq k < d$ [53].*

Authors prove the following

Theorem 3.1. *Let $A1 = [f_{ij}]$ and $A2 = [g_{ij}]$ be two $d \times d$ matrices of linear Boolean expressions, and let $b_1 = [u_i]$ and $b_2 = [v_i]$ be two $d \times 1$ vectors of linear or quadratic Boolean expressions. Let the functions f_{ij} and u_i depend only on variables $x_1, ..., x_d$, and let the functions g_{ij} and v_i depend only on variables $x_{d+1}, ..., x_{2d}$. If $Det(A_1) = Det(A_2) = 1$ in $GF(2)$ and if*

$$A_1 \cdot (x_{d+1}, ..., x_{2d})^T + b_1 \equiv A_2 \cdot (x_1, ..., x_d)^T + b_2$$

then the vector valued operation $*_{vv}(x_1, ..., x_{2d}) = A_1 \cdot (x_{d+1}, ..., x_{2d})^T + b_1$ defines a quasigroup $(Q,*)$ of order 2^d that is MQQ [53].

The authors researched the existence of MQQ of order 8, 16 and 32.

Problem 3.1. Finding MQQs of orders 2^d, $d \geq 6$ the authors consider as an open research problem.

Authors show that the proposed cipher is resistant relatively to the chosen plain-text attack, attacks with differential cryptanalysis, XL attack, Grobner basis attacks and some other kind of attacks.

Algebraic cryptanalysis of MQQ public key cryptosystem is given in [93]: "... we present an efficient attack of the multivariate Quadratic Quasigroups (MQQ) cryptosystem. Our cryptanalysis breaks MQQ cryptosystems by solving systems of multivariate quadratic polynomial equations using a modified version of the MutantXL algorithm".

In order to make Algorithm 3.1 more complicate and quite fast we propose the following

Procedure 3.1. *Let A be a non-empty alphabet, k be a natural number, $u_i, v_i \in A$, $i \in \{1, ..., k\}$. Define a system of n n-ary orthogonal* *operations (A, f_i), $i = 1, 2, \ldots, n$. We propose the following ciphering procedure $v_i = f_i(u_1, u_2, \ldots, u_n)$, $i = 1, 2, ..., n$. Therefore we obtain the following cipher-text $v_1 v_2 ... v_n$.*

The enciphering algorithm is based on the fact that orthogonal system of n n-ary operations

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = a_1 \\ f_2(x_1, x_2, \ldots, x_n) = a_2 \\ \ldots \\ f_n(x_1, x_2, \ldots, x_n) = a_n \end{cases}$$

has a unique solution for any tuple of elements $a_1, \ldots, a_n$.

Notice that we can take as a system of orthogonal n-ary operations a set of orthogonal n-quasigroups [117, 118, 44].

Of course this choice does not make Procedure 3.1 more safe, but it gives a possibility to use Algorithm 3.2 and Procedure 3.1 together on the base of the same quasigroup system.

Probably there exists a sense to use in Algorithm 3.2 the irreducible 3-ary or 4-ary finite quasigroup [1, 2].

4 Some applications of quasigroup-based stream ciphers

In [100] (see also [87]) it is proposed to use Algorithm 3.1 for secure encoding of file system. A survey of security mechanisms in mobile communication systems is in [120].

SMS (Short Message Service) messages are sometimes used for the interchange of confidential data such as social security number, bank account number, password etc. A typing error in selecting a number when sending such a message can have severe consequences if the message is readable to any receiver.

Most mobile operators encrypt all mobile communication data, including SMS messages. But sometimes, when encrypted, the data is readable for the operator.

Among others these needs give rise for the need to develop additional encryption for SMS messages, so that only accredited parties are able to be engaged in a communication. In [60] an approach to this problem using Algorithm 3.1 is described. In [61] differential cryptanalysis of the quasigroup cipher is given. Definition of the encryption method is presented.

In [87] the authors introduce a stream cipher with almost public key, based on quasigroups for defining suitable encryption and decryption. They consider the security of this method. It is shown that the key (quasigroups) can be public and still has sufficient security. A software implementation is also given.

In [81] a public-key cryptosystem, using generalized quasigroup-based streamciphers is presented. It is shown that such a cryptosystem allows one to transmit securely both a cryptogram and a secret portion of the enciphering key using the same insecure channel. The system is illustrated by means of a simple, but nontrivial, example.

5 Neo-fields and left neo-fields

A left neo-field $(N, +, \cdot)$ of order n consists of a set N of n symbols on which two binary operations "$+$" and "$\cdot$" are defined such that $(N, +)$ is a loop, with identity element, say 0. $(N\backslash\{0\}, \cdot)$ is a group and the operation "$\cdot$" distributes from the left over "$+$". (That is, $x \cdot (y + z) = x \cdot y + x \cdot z$ for all $x, y, z \in N$.) If the right distributive law also holds, the structure is called a neofield.

A left neofield (or neofield) whose multiplication group is $(G, \cdot)$ is said to be based on that group. Clearly, every left neofield based on an abelian group is a neofield. Also, a neofield whose operation of addition satisfies the associative law is a field.

In [28, 27] some cryptological applications of neo-fields and left neo-fields are described.

6 On one-way function

A function $F : X \to Y$ is called one-way function, if the following conditions are fulfilled:

- there exists a polynomial algorithm of calculation of $F(x)$ for any $x \in X$;

- there does not exist a polynomial algorithm of inverting of the function F, i.e. there does not exist any polynomial time algorithm for solving the equation $F(x) = y$ relatively variable x.

It is proved that the problem of existence of one-way function is equivalent to well known problem of coincidence of classes P and NP.

One of better candidates to be an one-way function is so-called function of discrete logarithms [83].

A neofield $(N, +, \cdot)$ of order n consists of a set N of n symbols on which two binary operations "$+$" and "$\cdot$" are defined such that $(N, +)$ is a loop with identity element, say 0, $(N \backslash \{0\}, \cdot)$ is a group and the operation "$\cdot$" distributes from the left and right over "$+$" [28].

Let $(N, +, \cdot)$ be a finite Galois field or a cyclic ($(N \backslash \{0\}, \cdot)$ is a cyclic group) neofield. Then each non-zero element u of the additive group or loop $(N, +)$ can be represented in the form $u = a^{\nu}$, where a is a generator of the multiplication group $(N \backslash \{0\}, \cdot)$. ν is called the discrete logarithm of u with base a, or, sometimes, the exponent or index of u.

Given ν and a, it is easy to compute u in a finite field, but, if the order of the finite field is a sufficiently large prime p and also is appropriately chosen it is believed to be difficult to compute ν when u (as a residue modulo p) and a are given.

In [28] discrete logarithms are studied over a cyclic neofield whose addition is a CI-loop.

In [83] the discrete logarithm problem for the group RL_n of all row-Latin squares of order n is defined (p.103) and, on pages 138 and 139, some illustrations of applications to cryptography are given.

7 On hash function

In [46, 45] an approach for construction of hash function using quasigroups is described.

Definition 7.1. *A function $H()$ that maps an arbitrary length message M to a fixed length hash value $H(M)$ is a OneWay Hash Function (OWHF), if it satisfies the following properties:*

1. The description of $H()$ is publicly known and should not require any secret information for its operation.

2. Given M, it is easy to compute $H(M)$.

3. Given $H(M)$ in the rang of $H()$, it is hard to find a message M for given $H(M)$, and given M and $H(M)$, it is hard to find a message $M_0(\neq M)$ such that $H(M_0) = H(M)$.

Definition 7.2. *A OneWay Hash Function $H()$ is called Collision Free Hash Function (CFHF), if it is hard to find two distinct messages M and M_0 that hash to the same result $(H(M) = H(M_0))$[46, 45].*

We give construction of hashing function based on quasigroup [46].

Definition 7.3. *Let $H_Q() : Q \longrightarrow Q$ be projection defined as*

$$H_Q(q_1 q_2 \dots q_n) = ((\dots(a \star q_1) \star q_2 \star \dots) \star q_n \tag{2}$$

Then $H_Q()$ is said to be hash function over quasigroup $(Q; \star)$. The element a is a fixed element from Q.

Example 7.1. Multiplication in the quasigroup $(Q, \star)$ is defined in the following manner: $a \star b = (a - b) \pmod 4$. This quasigroup has the following multiplication table:

$\star$	0	1	2	3
0	0	3	2	1
1	1	0	3	2
2	2	1	0	3
3	3	2	1	0

Value of hash function is $H_2(0013) = (((2 \star 0) \star 0) \star 1) \star 3 = 2$.

Remark 7.1. There exists a possibility to apply n-ary quasigroup approach to study hash functions of such kind. Since, in fact, equality (2) defines an n-ary operation.

Remark 7.2. We notice, safe hash function must have at least 128-bit image, i.e. $H_Q(q_1 q_2 \dots q_n)$ must consist of at least 128-digit number [96].

In [121, 122] hash functions, proposed in [46, 45], are discussed. The author shows that for some types of quasigroups these hash functions are not secure.

From [86] we give the following summary: "In this paper we consider two quasigroup transformations $QM1\colon A^{2m} \rightarrow A^{2m}$ and $QM2\colon A^{m} \rightarrow A^{2m}$, where A is the carrier of a quasigroup. Based on these transformations we show that different kinds of hash functions can be designed with suitable security."

Further development of quasigroup based on hash function is reflected in [116].

In [105] on Algorithm 3.1 based on encrypter that has good scrambling properties is proposed.

8 CI-quasigroups and cryptology

In [28, 56] some applications of CI-quasigroups in cryptology with non-symmetric key are described.

Definition 8.1. *Suppose that there exists a permutation J of the elements of a quasigroup $(Q, \circ)$ such that, for all $x, y \in Q$*

$$J^r(x \circ y) \circ J^s x = J^t y,$$

where r, s, t are integers. Then $(Q, \circ)$ is called an (r, s, t)-inverse quasigroup ([72]).

In the special case when $r = t = 0$, $s = 1$, we have a definition of CI-quasigroup.

Example 8.1. A CI-quasigroup can be used to provide a one-time pad for key exchange (without the intervention of a key distributing centre) [28, 68].

The sender S, using a physical random number generator (see [78] on random number generator based on quasigroups), selects an arbitrary element $c^{(u)}$ of the CI-quasigroup $(Q, \circ)$ and sends both $c^{(u)}$ and enciphered key (message) $c^{(u)} \circ m$. The receiver R uses this knowledge of the algorithm for obtaining $Jc^{(u)} = c^{(u+1)}$ from $c^{(u)}$ and hence he computes $(c^{(u)} \circ m) \circ c^{(u+1)} = m$.

Example 8.2. We can propose the following application of rst-inverse quasigroups in situation similar to situation described in Example 8.1. It is possible to re-write definitive equality of rst-inverse quasigroup in the following manner $J^r(J^k u \circ m) \circ J^{s+k}u = J^t m$.

Then the schema of the previous example can be re-written in the following manner. The sender S selects an arbitrary element $J^k u$ of the rst-quasigroup $(Q, \circ)$ and sends both $J^k u$ and enciphered key (message) $J^r(J^k u \circ m)$. The receiver R uses this knowledge of the algorithm for obtaining $J^{k+s}(u)$ from $J^k(u)$ and hence he computes $J^r(J^k u \circ m) \circ J^{s+k}u = J^t m$ and after this he computes the message m. Of course this example can be modified.

Example 8.3. [28]. Take a CI-quasigroup with a long inverse cycle $(c\, c'\, c'' \dots\; c^{t-1})$ of length t. Suppose that all the users U_i $(i = 1, 2, \dots)$ are provided with apparatus (for example, a chip card) which will compute $a \circ b$ for any given $a, b \in Q$. We assume that only the key distributing centre has a knowledge of the long inverse cycle which serves as a look-up table for keys.

Each user U_i has a public key $u_i \in Q$ and a private key Ju_i, both supplied in advance by the key distributing centre. User U_s wishes to send a message m to user U_t. He uses U_t's public key u_t to compute $u_t \circ m$ and sends that to U_t. U_t computes $(u_t \circ m) \circ Ju_t = m$.

Remark 8.1. It is not very difficult to understand that opponent which knows the permutation J may decipher a message encrypted by this method.

Remark 8.2. There exists a possibility to generalize Example 8.3 using some m-inverse quasigroups [71], or (r, s, t)-inverse quasigroups [72, 73], else (α, β, γ)-inverse quasigroups [74].

9 Critical sets and secret sharing systems

Definition 9.1. *A critical set C in a Latin square L of order n is a set $C = \{(i; j; k) \mid i, j, k \in \{1, 2, \dots, n\}\}$ with the following two properties:*

(1) L is the only Latin square of order n which has symbols k in cell (i, j) for each $(i; j; k) \in C$;

(2) no proper subset of C has property (1) [83].

A critical set is called minimal if it is a critical set of smallest possible cardinality for L. In other words a critical set is a partial Latin square which is uniquely completable to a Latin square of order n.

If the scheme has k participants, a (t, k)-secret sharing scheme is a system where k pieces of information called shares or shadows of a secret key K are distributed so that each participant has a share such that

(1) the key K can be reconstructed from knowledge of any t or more shares;

(2) the key K cannot be reconstructed from knowledge of fewer than t shares.

Such systems were first studied in 1979. Simmons [115] surveyed various secret sharing schemes. Secret sharing schemes based on critical sets in Latin squares are studied in [17]. We note, critical sets of Latin squares give rise to the possibilities to construct secret-sharing systems.

Critical sets of Latin squares were studied in sufficiently big number of articles. We survey results from some of these articles. In [34] the spectrum of critical sets in Latin squares of order 2^n is studied. The paper [30] gives constructive proofs that critical sets exist for all sizes between $[n^2/4]$ and $[(n^2-n)/2]$, with the exception of size $n^2/4+1$ for even values of n.

For Latin squares of order n, the size of a smallest critical set is denoted by $\mathrm{scs}(n)$ in [15]. The main result of [15] is that $\mathrm{scs}(n) \geq n\lfloor \frac{1}{2}(\log n)^{1/3} \rfloor$ for all positive integers n.

In [63] the authors show that any critical set in a Latin square of order $n \geq 7$ must have at least $\lfloor \frac{7n-\sqrt{n}-20}{2} \rfloor$ empty cells. See, also, [62].

The paper [33] contains lists of (a) theorems on the possible sizes of critical sets in Latin squares of order less than 11, (b) publications, where these theorems are proved, (c) concrete examples of such type of critical sets. In [36] an algorithm for writing any Latin interchange as a sum of intercalates is corrected.

In [59] the author proposes a greedy algorithm to find critical sets in Latin squares. He applies this algorithm to Latin squares which are abelian 2-groups to find new critical sets in these Latin squares. The critical sets have the nice property that they all intersect some 2×2 Latin subsquare in a unique element so that it is easy to show the criticality.

In [4] the author gives an example of a critical set of size 121 in the elementary abelian 2-group of order 16.

In [94] critical sets of symmetric Latin squares are studied. Therefore the authors require all elements in their critical sets and uniquely completable partial Latin squares to lie on or above the main diagonal. For $n > 2$, a general procedure is given for writing down a uniquely completable partial symmetric $2n \times 2n$ Latin square L'_{2n} containing $n^2 - n + 2$ entries, of which $2n - 2$ are identical and lie on the main diagonal.

Paper [32] presents a solution to the interesting combinatorial problem of finding a minimal number of elements in a given Latin square of odd order n by which one may restore the initial form of this square. In particular, it is proved that in every cyclic Latin square of odd order n the minimal number of elements equals to $n(n-1)/2$.

Surveys on critical sets of Latin squares are given in [67, 69]. See, also, [70].

The concept of Latin trades is closely connected with the concept of critical set in Latin squares. Let T be a partial Latin square and L be a Latin square with $T \subseteq L$. We say that T is a Latin trade if there exists a partial Latin square T' with $T' \cap T = \emptyset$ such that $(L \setminus T) \cup T'$ is a Latin square. Information on Latin trades is in [16].

Remark 9.1. See also Introduction for other application of critical sets of Latin squares in cryptology.

"For a given triple of permutations $T = (\alpha, \beta, \gamma)$ the set of all Latin squares L such that T is its autotopy is denoted by $LS(T)$. The cardinality of $LS(T)$ is denoted by $\Delta(T)$. Specifically, the computation of $\Delta(T)$ for any triple T is at the moment an open problem having relevance in secret sharing schemes related to Latin squares" [49, 50].

10 Secret sharing systems and other algebraic systems

Some secret-sharing systems are pointed in [26]. One of such systems is the Reed-Solomon code over a Galois field $GF[q]$ with generating matrix $C(a_{ij})$ of size $k \times (q-1)$, $k \leq q-1$. The determinant formed by any k columns of G is a non-zero element of $GF[q]$. The Hamming distance d of this code is maximal ($d = q - k$) and any k from $q-1$ keys unlock the secret.

In [9] an approach to some Reed-Solomon codes as a some kind of orthogonal systems of n-ary operations is developed.

In [10] general approach to construction of secret sharing systems using some kinds of orthogonal systems of n-ary operations is given. Transformations of orthogonal systems of n-ary operations are studied in [11].

We give the summary from [52] : ”We investigate subsets of critical sets of some Youden squares in the context of secret-sharing schemes. A subset $\mathcal{C}$ of a Youden square is called a critical set if $\mathcal{C}$ can be uniquely completed to a Youden square but no proper subset of $\mathcal{C}$ has a unique completion to a Youden square.”

”That part of a Youden square Y which is inaccessible to subsets of a critical set $\mathcal{C}$ of Y, called the strongbox of $\mathcal{C}$, may be thought to contain secret information. We study the size of the secret. J. R. Seberry and A. P. Street [108] have shown how strongboxes may be used in hierarchical and compartmentalized secret-sharing schemes.”

11 Row-Latin squares based cryptosystems

A possible application in cryptology of Latin power sets is proposed in [29].

In [23] an encrypting device is described, based on row-Latin squares with maximal period equal to the Mangoldt function.

In our opinion big perspectives has an application of row-Latin squares in various branches of contemporary cryptology (”neo-cryptology”).

In [83] it is proposed to use: 1) row-Latin squares to generate an open key; 2) a conventional system for transmission of a message that is the form of a Latin square; 3) row-Latin square analogue of the RSA system; 4) procedure of digital signature based on row-Latin squares.

Example 11.1. Let

$$L = \begin{matrix} 2 & 3 & 4 & 1 \\ 4 & 1 & 3 & 2 \\ 3 & 2 & 4 & 1 \\ 4 & 3 & 1 & 2 \end{matrix}$$

Then

$$L^7 = \begin{matrix} 4 & 1 & 2 & 3 \\ 4 & 1 & 2 & 3 \\ 3 & 2 & 4 & 1 \\ 3 & 4 & 2 & 1 \end{matrix}$$

$$L^3 = \begin{matrix} 4 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{matrix}$$

Then

$$L^{21} = \begin{matrix} 2 & 3 & 4 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 \end{matrix}$$

is a common key for a user A with the key L^3 and a user B with the key L^7.

A public-key cryptosystem, using generalized quasigroup-based streamciphers, as it has been noticed earlier, is presented in [81].

12 NLPN sequences over GF[q]

Non-binary pseudo-random sequences over GF[q] of length $q^m - 1$ called PN sequences have been known for a long time [57]. PN sequences over a finite field GF[q] are unsuitable directly for cryptology because of their strong linear structure [78]. Usually PN sequences are defined over a finite field and often an irreducible polynomial for their generation is used.

In article [78] definition of PN sequence was generalized with the purpose to use these sequences in cryptology.

We notice, in some sense ciphering is making a “pseudo-random sequence” from a plaintext, and cryptanalysis is a science how to reduce a check of all possible variants (cases) by deciphering of some ciphertext.

These new sequences were called NLPN-sequences (non-linear pseudo-noise sequences). C. Koscielny proposed the following method for construction of NLPN-sequences.

Let $\overrightarrow{a}$ be a PN sequence of length $q^m - 1$ over GF[q], $q > 2$, i.e.

$$\overrightarrow{a} = a_0 a_1 \dots a_{q^m - 2}.$$

Let $\overrightarrow{a}^i$ be its cyclic i places shifted to the right. For example

$$\overrightarrow{a}^1 = a_1 \dots a_{q^m - 2} a_0.$$

Let $Q = (SQ, \cdot)$ be a quasigroup of order q defined on the set of elements of the field GF[q].

Then $\overrightarrow{b} = \overrightarrow{a} \cdot \overrightarrow{a}^i$, $\overrightarrow{c} = \overrightarrow{a}^i \cdot \overrightarrow{a}$, where $b_j = a_j \cdot a^i_j$, $c_j = a^i_j \cdot a_j$ for any suitable value of index j ($j \in \{1, 2, \dots, q^m - 1\}$) are called NLPN sequences [78].

NLPN sequences have much more randomness than PN sequences. As notice C. Koscielny the method of construction of NLPN sequences is especially convenient for fast software encryption. It is proposed to use NLPN sequences by generation of keys. See also [76].

13 Authentication of a message

By authentication of message we mean that it is made possible for a receiver of a message to verify that the message has not been modified in transit, so that it is not possible for an interceptor to substitute a false message for a legitimate one.

By identification of a message we mean that it is made possible for the receiver of a message to ascertain its origin, so that it is not possible for an intruder to masquerade as someone else.

By non-repudiation we mean that a sender should not be able later to deny falsely that he had sent a message.

In [28] some quasigroup approaches to problems of identification of a message, problem of non-repudiation of a message, production of dynamic password and to digital fingerprinting are discussed. See also [18].

In [27] authors suggested a new authentication scheme based on quasigroups (Latin squares). See also [26, 28, 20]

In [104] several cryptosystems based on quasigroups upon various combinatorial objects such as orthogonal Latin squares and frequency squares, block designs, and room squares are considered.

Definition 13.1. *Let $2 \leq t < k < v$. A generalized $S(t,k,v)$ Steiner system is a finite block design $(T, \mathcal{B})$ such that (1) $|T| = v$; (2) $\mathcal{B} = \mathcal{B}' \cup \mathcal{B}''$, where any $B' \in \mathcal{B}'$, called a maximal block, has k points and $2 \leq |B''| < k$ for any $B'' \in \mathcal{B}''$, called a small block; (3) for any $B'' \in \mathcal{B}''$ there exists a $B' \in \mathcal{B}'$ such that $B'' \subseteq B'$; (4) every subset of T with t elements not belonging to the same $B'' \in \mathcal{B}''$ is contained in exactly one maximal block.*

In [89] (see also [48]) an application of generalized $S(t,k,v)$ Steiner systems in cryptology is proposed, namely, it is introduced a new authentication scheme based on the generalized Steiner systems, and the properties of such scheme are studied in the generalized affine planes.

14 Zero knowledge protocol

In [103] Rivest introduced All-Or-Nothing (AON) encryption mode in order to devise means to make brute-force search more difficult, by appropriately pre-processing a message before encrypting it. The method is general, but it was initially discussed for block-cipher encryption, using fixed-length blocks.

It is an unkeyed transformation, mapping a sequence of input blocks $(x_1, x_2, \dots, x_s)$ to a sequence of output blocks $(y_1, y_2, \dots, y_t)$ having the following properties:

Having all blocks $(y_1, y_2, \dots, y_t)$ it is easy to compute $(x_1, x_2, \dots, x_s)$.

If any output block y_j is missing, then it is computationally infeasible to obtain any information about any input block x_J.

The main idea is to preserve a small-length key (e.g. 64-bit) for the main encryption that can be handled by special hardware with not enough processing power or memory. This gives the method a strong advantage, since we can have strong encryption for devices that have minimum performance.

Several transformation methods have been proposed in the literature for AON. In the article [88] it is proposed a special transform which is based on the use of a quasigroup (it is used in algorithm 3.1).

In [24] it is proposed to use isotopy of quasigroups in zero knowledge protocol.

Assume the users $(u_1, u_2, ..., u_k)$ form a network. The user u_i has public-key L_{u_i}, $L^{'}_{u_i}$ (denotes two isotopic Latin squares of order n) and secret-key I_{u_i} (denotes the isotopism of L_{u_i} upon $L^{'}_{u_i}$). The user u_i wants to prove identity for u_j but he doesn't want to reveal the secret-key (zero-knowledge proof).

1. u_i randomly permutes L_{u_i} to produce another Latin square H.
2. u_i sends H to u_j.
3. u_j asks u_i either to:
   a. prove that H and $L^{'}_{u_i}$ are isotopic,
   b. prove that H and L_{u_i} are isotopic.
4. u_i complies. He either
   a. proves that H and $L^{'}_{u_i}$ are isotopic,
   b. proves that H and L_{u_i} are isotopic.
5. u_i and u_j repeat steps 1. through 4. n times.

Remark 14.1. In the last procedure it is possible to use isotopy of n-ary groupoids.

15 Hamming distance between quasigroups

The following question is very important by construction of quasigroup based cryptosystems: how big is the distance between different binary or n-ary quasigroups? Information on Hamming distance between quasigroup operation is in the articles [41, 42, 39, 38, 40, 43, 119].

We recall, if α and β are two n-ary operations on a finite set Ω, then the Hamming distance of α and β is defined by

$$\mathrm{dist}(\alpha,\beta) = |\{(u_1,\dots,u_n) \in \Omega^n : \alpha(u_1,\dots,u_n) \neq \beta(u_1,\dots,u_n)\}|.$$

The author in [41] discusses Hamming distances of algebraic objects with binary operations. He also explains how the distance set of two quasigroups yields a 2-complex, and points out a connection with dissections of equilateral triangles.

For a fixed group $(G,\circ)$, $\delta(G,\circ)$ is defined to be the minimum of all such distances for $(G,\star)$ not equal to $(G,\circ)$ and $\nu(G,\circ)$ the minimum for $(G,\star)$ not isomorphic to $(G,\circ)$.

In [38] it is proved that $\delta(G,\circ)$ is $6n-18$ if n is odd, $6n-20$ if $(G,\circ)$ is dihedral of twice odd order and $6n-24$ otherwise for any group $(G,\circ)$ of order greater than 50. In [119] it is shown that $\delta(G,\circ) = 6p-18$ for $n=p$, a prime, and $p>7$.

In the article [39] there are listed a number of group orders for which the distance is less than the value suggested by the above theorems. New results obtained in this direction are in [43].

16 Generation of quasigroups for cryptographical needs

Important cryptographical problem is a generation of "big" quasigroups which it is possible to keep easily in a compact form in computer memory. It is clear that for this aims the most suitable is a way to keep a little base and some procedures of obtaining a necessary element.

Therefore we should have easily generated objects (cyclic group, abelian group, group), fast and complicate methods of their transformation (parastrophy, isotopy, isostrophy, crossed isotopy [109], homotopy, generalized isotopy), their glue and blowing (direct product, semi-direct product, wreath product [66], crossed product, generalized crossed product). For these aims various linear quasigroups (especially n-ary quasigrous) are quite suitable [7, 85, 113].

In [99] the boolean function is proposed to use by construction of n-ary and binary quasigroups.

A method of generating a practically unlimited number of quasigroups of an arbitrary (theoretically) order using the computer algebra system Maple 7 is presented in [79].

This problem is crucial to cryptography and its solution permits to implement practical quasigroup-based endomorphic cryptosystems.

In this article [79] it is proposed to use isotopy of quasigroups and direct products of quasigroups. If we start from class of finite groups, then, using these ways, it is possible to obtain only class of quasigroups that are isotopic to groups. We notice, there exists many quasigroups (especially of large order) that are not isotopic to a group. Therefore for construction of quasigroups that are not isotopic to groups probably better to use the concept of gisotopy [98, 113].

17 Conclusion remarks

In many cases in cryptography it is possible to change associative systems by non-associative ones and practically in any case this change gives in some sense better results than use of associative systems. Quasigroups in spite of their simplicity, have various applications in cryptology. Many new cryptographical algorithms can be formed on the basis of quasigroups.

Acknowledgment. The author thanks the Supreme Council for Science and Technological Development of Republic of Moldova (grant 08.820.08.08 RF) for financial support.

References

[1] M. A. Akivis and V. V. Goldberg. Solution of Belousov's problem, 2000. Arxiv:math.GR/0010175.

[2] M. A. Akivis and V. V. Goldberg. Solution of Belousov's problem. *Discuss. Math. Gen. Algebra Appl.*, 21(1):93–103, 2001.

[3] H.J. Baker and F. Piper. *Cipher Systems: the Protection of Communications*. Northwood, London, 1982.

[4] R. Bean. Critical sets in the elementary abelian 2- and 3-groups. *Util. Math.*, 68:53–61, 2005.

[5] R. Bean, D. Donovan, A. Khodkar, and A.P. Street. Steiner trades that give rise to completely decomposable latin interchanges. *Int. J. Comput. Math.*, 79(12):1273–1284, 2002.

[6] V.D. Belousov. *Foundations of the Theory of Quasigroups and Loops*. Nauka, Moscow, 1967. (in Russian).

[7] V.D. Belousov. *n-Ary Quasigroups*. Stiintsa, Kishinev, 1971. (in Russian).

[8] V.D. Belousov. *Elements of Quasigroup Theory: a special course*. Kishinev State University Printing House, Kishinev, 1981. (in Russian).

[9] G.B. Belyavskaya. Secret-sharing systems and orthogonal systems of operatins. In *Applied and Industrial Mathematics, Abstracts*, page 2, Chisinau, Moldova, 1995.

[10] G.B. Belyavskaya. Secret-sharing schemes and orthogonal systems of k-ary operations. *Quasigroups and related systems*, 17(2):111–130, 2009.

[11] G.B. Belyavskaya. Transformation of orthogonal systems of polynomial n-ary operations. In *VII-th Theoretical and Practical Seminar Combinatorial Configurations and their Applications, Kirovograd, April 17-18, 2009*, page 3. 2009.

[12] C. H. Bennet and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*, page 175, Bangalore, 1984.

[13] A. Beutelspacher. *Cryptology: An introduction to the science of encoding, concealing and hiding.* Vieweg, Wiesbaden, 2002. (in German).

[14] Cyril Branciard, Nicolas Gisin, Barbara Kraus, and Valerio Scarani. Security of two quantum cryptography protocols using the same four qubit states. *Phys. Rev. A*, 72:032301, 2005. arXiv:quant-ph/0505035v2.

[15] N.J. Cavenagh. A superlinear lower bound for the size of a critical set in a Latin square. *J. Combin. Des.*, 15(4):369–282, 2007.

[16] N.J. Cavenagh, D. Donovan, and A. Drapal. 3-homogeneous latin trades. *Discrete Math.*, 300(1-3):57–70, 2005.

[17] J. Cooper, D. Donovan, and J. Seberry. Secret sharing schemes arising from latin squares. *Bull. Inst. Combin. Appl.*, 12:33–43, 1994.

[18] D. Coppersmith. Weakness in quaternion signatures. *J. Cryptology*, 14:77–85, 2001.

[19] M. Damm. Prüfziffersysteme über Quasigruppen. Master's thesis, Philipps-Universität Marburg, 1998. (in German).

[20] E. Dawson, D. Donowan, and A. Offer. Ouasigroups, isotopisms and authentification schemes. *Australas. J. Combin.*, 13:75–88, 1996.

[21] P. Dehornoy. Braid-based cryptography. *Contemp. Math., Group theory, statistics, and cryptography*, 360:5–33, 2004.

[22] J. Dénes. Latin squares and non-binary encoding. In *Proc. conf. information theory, CNRS*, pages 215–221, Paris, 1979.

[23] J. Dénes. On latin squares and a digital encrypting communication system. *P.U.M.A., Pure Math. Appl.*, 11(4):559–563, 2000.

[24] J. Dénes and T. Dénes. Non-associative algebraic system in cryptology. Protection against "meet in the middle" attack. *Quasigroups Relat. Syst.*, 8:7–14, 2001.

[25] J. Dénes and A. D. Keedwell. *Latin Squares and their Applications.* Académiai Kiadó, Budapest, 1974.

[26] J. Dénes and A. D. Keedwell. *Latin Squares. New Development in the Theory and Applications*, volume 46 of *Annals of Discrete Mathematics.* North-Holland, 1991.

[27] J. Dénes and A. D. Keedwell. A new authentification scheme based on latin squres. *Discrete Math.*, 106/107:157–165, 1992.

[28] J. Dénes and A. D. Keedwell. Some applications of non-associative algebraic systems in cryptology. *P.U.M.A.*, 12(2):147–195, 2002.

[29] J. Dénes and P. Petroczki. A digital encrypting communication systems, 1990.

[30] W. Diffie and M.F. Hellman. New directions in cryptography. *IEEE, Transactions of Information Theory*, IT-22:644–654, 1976.

[31] V. Domashev, V. Popov, D. Pravikov, I. Prokof'ev, and A. Shcherbakov. *Programming of algorithms of defense of information.* Nolidge, Moscow, 2000. (in Russian).

[32] D. Donowan. Critical sets for families of latin squares. *Util. Math.*, 53:3–16, 1998.

[33] D. Donowan. Critical sets in latin squares of order less than 11. *J. Comb. Math. Comb. Comput.*, 29:223–240, 1999.

[34] D. Donowan, J. Fevre, and G. H. John van Rees. On the spectrum of critical sets in latin squares of order 2^n. *J. Combin. Des.*, 16(1):25–43, 2008.

[35] D. Donowan and A. Howse. Correction to a paper on critical sets. *Australas. J. Combin.*, 21:107–130, 2000.

[36] D. Donowan and E.S. Mahmoodian. Correction to a paper on critical sets. *Bull. Inst. Comb. Appl.*, 37:44, 2003.

[37] S.A. Dorichenko and V.V. Yashchenko. *25 sketches on ciphers.* Teis, Moscow, 1994. (in Russian).

[38] A. Drapal. How far apart can the group multiplication tables be? *Eur. J. Comb.*, 13(5):335–343, 1992.

[39] A. Drapal. On distances of multiplication tables of groups. *Lond. Math. Soc. Lect. Note Ser.*, 260:248–252, 1999.

[40] A. Drapal. Non-isomorphic 2-groups coincide at most in three quartes of their multiplication table. *Eur. J. Comb.*, 21:301–321, 2000.

[41] A. Drapal. Hamming distances of groups and quasi-groups. *Discrete Math.*, 235(1-3):189–197, 2001.

[42] A. Drapal. On groups that differ in one of four squares. *Eur. J. Comb.*, 23(8):899–918, 2002.

[43] A. Drapal and N. Zhukavets. On multiplication tables of groups that agree on half of the columns and half of the rows. *Glasgow Math. J.*, 45:293–308, 2003.

[44] W.A. Dudek and P.N. Syrbu. About self-orthogonal n-groups. *Bul. Acad. Stiinte Repub. Mold., Mat.*, (3):37–42, 1992. (in Russian).

[45] J. Dvorsky, E. Ochodkova, and V. Snasel. Hashovaci funkce zalozena na kvazigrupach. In *Workshop Milkulasska kryptobesidka*, Praha, 2000. (in Czech).

[46] J. Dvorsky, E. Ochodkova, and V. Snasel. Hash functions based on large quasigroups. *Velokonocni kryptologie*, pages 1–8, 2002.

[47] A. Ekert. From quantum, code-making to quantum code-breaking. In *Proceedings of the symposium on geometric issues in*

the foundations of science, Oxford, UK, June 1996 in honour of Roger Penrose in his 65th year, pages 195–214. Oxford University Press, 1998.

[48] F. Eugeni and A. Maturo. A new authentication system based on the generalized affine planes. *J. Inf. Optimization Sci.*, 13(2):183–193, 1992.

[49] R. M. Falcon. Latin squares associated to principal autotopisms of long cycles. application in cryptography. In *Proc. Transgressive Computing 2006: a conference in honor of Jean Della Dora*, pages 213–230, 2006.

[50] R. M. Falcon. Cycle structures of autotopisms of the Latin squares of order up to 11. *http://arxiv.org/*, 0709.2973:18 pages, 2007.

[51] Horst Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15–23, 1973.

[52] L. Fitina, K. G. Russell, and J. Seberry. The power and influence in some Youden squares and secret sharing. *Util. Math.*, 73:143–157, 2007.

[53] D. Gligoroski, S. Markovski, and S. J. Knapskog. A public key block cipher based on multivariate quadratic quasigroups. *http://arxiv.org/*, 0808.0247:22 pages, 2008.

[54] D. Gligoroski, S. Markovski, and L. Kocarev. Edon-R, An infnite family of cryptographic hash functions, 2006. http://csrc.nist.gov/pki/HashWorkshop/2006/Papers.

[55] M. M. Glukhov. On application of quasigroups in cryptology. *Applied discrete mathematics*, 2:28–32, 2008. (in Russian).

[56] S. Golomb, L. Welch, and J. Denes. Encryption system based on crossed inverse quasigroups, 2001. US patent, WO0191368.

[57] S.W. Golomb. *Shift Register Sequences.* Holden Day, San Francisco, 1967.

[58] Marshall Hall. *Combinatorial Theory.* Blaisdell Publishing Company, Massachusetts, 1967.

[59] C. Hamalainen. New 2-critical sets in the abelian 2-group. *J. Combin. Math. Combin. Comput.*, 61:193–219, 2007.

[60] M. Hassinen and S. Markovski. Secure SMS messaging using Quasigroup encryption and Java SMS API. In *SPLST'03*, Kuopio, Finland, June 2003.

[61] M. Hassinen and S. Markovski. Differential cryptanalysis of the quasigroup cipher. Definition of the encryption method. In *Differential cryptanalysis*, Petrozavodsk, June 2004.

[62] P. Horak, R. E. L. Aldred, and H. J. Fleischner. Completing latin squares: critical sets. i. *J. Combin. Des.*, 10(6):419–432, 2002.

[63] P. Horak and I. J. Dejter. Completing latin squares: critical sets. ii. *J. Combin. Des.*, 15(1):77–83, 2007.

[64] V.I. Izbash. Monoquasigroups without congruences and automorphisms. *Bul. Acad. Stiinte Repub. Mold., Mat.*, (4):66–76, 1992.

[65] D. Kahn. *The codebreakers: the story of secret writing.* Wiedenfield and Nicolson, London, 1967.

[66] M.I. Kargapolov and M.Yu. Merzlyakov. *Foundations of Group Theory.* Nauka, Moscow, 1977. (in Russian).

[67] A.D. Keedwell. Critical sets for latin squares, graphs and block designs: a survey. *Congressus Numeratium*, 113:231–245, 1996.

[68] A.D. Keedwell. Crossed-inverse quasigroups with long inverse cycles and applications to cryptography. *Australas. J. Combin.*, 20:241–250, 1999.

[69] A.D. Keedwell. Critical sets in latin squares and related matters: an update. *Util. Math.*, 65:97–131, 2004.

[70] A.D. Keedwell. On sudoku squares. *Bull. Inst. Combin. Appl.*, 50:52–60, 2007.

[71] A.D. Keedwell and V.A. Shcherbacov. On m-inverse loops and quasigroups with a long inverse cycle. *Australas. J. Combin.*, 26:99–119, 2002.

[72] A.D. Keedwell and V.A. Shcherbacov. Construction and properties of (r,s,t)-inverse quasigroups, I. *Discrete Math.*, 266(1-3):275–291, 2003.

[73] A.D. Keedwell and V.A. Shcherbacov. Construction and properties of (r,s,t)-inverse quasigroups, II. *Discrete Math.*, 288:61–71, 2004.

[74] A.D. Keedwell and V.A. Shcherbacov. Quasigroups with an inverse property and generalized parastrophic identities. *Quasigroups Relat. Syst.*, 13:109–124, 2005.

[75] T. Kepka. A note on simple quasigroups. *Acta Univ. Carolin. Math. Phys.*, 19(2):59–60, 1978.

[76] A. Klapper. On the existence of secure keystream generators. *J. Cryptology*, 14:1–15, 2001.

[77] C. Koscielny. A method of constructing quasigroup-based stream ciphers. *Appl. Math. and Comp. Sci.*, 6:109–121, 1996.

[78] C. Koscielny. NLPN Sequences over GF(q). *Quasigroups Relat. Syst.*, 4:89–102, 1997.

[79] C. Koscielny. Generating quasigroups for cryptographic applications. *Int. J. Appl. Math. Comput. Sci.*, 12(4):559–569, 2002.

[80] C. Koscielny. Stegano crypto graphy with maple 8. Technical report, Institute of Control and Computation Engineering, University of Zielona Gora,

http://www.mapleapps.com/categories/mathematics/Cryptography /html/stegcryp.html, 2003.

[81] C. Koscielny and G.L. Mullen. A quasigroup-based public-key cryptosystem. *Int. J. Appl. Math. Comput. Sci.*, 9(4):955–963, 1999.

[82] A. V. Kuznetsov and A.F. Danilchenko. Functionally complete quasigroups. In *First All-Union Simposium on quasigroup theory and its applications. Abstracts of reports and talks*, pages 17–19, Tbilisi, 1968.

[83] Charles F. Laywine and Gary L. Mullen. *Discrete Mathematics Using Latin Squares.* John Wiley & Sons, Inc., New York, 1998.

[84] S.S. Magliveras, D.R. Stinson, and Tran van Trung. New approach to designing public key cryptosystems using one-way function and trapdoors in finite groups. *J. Cryptology*, 15:285–297, 2002.

[85] A. Marini and V.A. Shcherbacov. On autotopies and automorphisms of n-ary linear quasigroups. *Algebra and Discrete Math.*, (2):51–75, 2004.

[86] S. Markovski, D. Gligoroski, and V. Bakeva. Quasigroups and hash functions. In *Res. Math. Comput. Sci.*, volume 6, pages 43–50, South-West Univ., Blagoevgrad, 2002.

[87] S. Markovski, D. Gligoroski, and B. Stojcevska. Secure two-way on-line communication by using quasigroup enciphering with almost public key. *Novi Sad J. Math.*, 30(2):43–49, 2000.

[88] S.I. Marnas, L. Angelis, and G.L. Bleris. All-or-nothing transforms using quasigroups. In *Proceedings of 1st Balkan Conference in Informatics*, pages 183–191, Thessaloniki, November 2003.

[89] A. Maturo and M. Zannetti. Redei blocking sets with two Redei lines and quasigroups. *J. Discrete Math. Sci. Cryptography*, 5(1):51–62, 2002.

[90] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, FL, 1997.

[91] L. Mittenhal. A source of cryptographically strong permutations for use in block ciphers. In *Proc. IEEE, International Sympos. on Information Theory, 1993, IEEE*, pages 17–22, New York, 1993.

[92] L. Mittenhal. Block substitutions using orthomorphic mappings. *Advances in Applied Mathematics*, 16:59–71, 1995.

[93] Mohamed Saied Emam Mohamed, Jintai Ding, and Johannes Buchmann. Algebraic Cryptanalysis of MQQ Public Key Cryptosystem by MutantXL, 2008. eprint.iacr.org/2008/451.pdf.

[94] D. A. Mojdeh and N.J. Rad. Critical sets in latin squares given that they are symmetric. *Univ. Beograd. Publ. Elektrotehn. Fak. Ser. Mat.*, 18:38–45, 2007.

[95] N.A. Moldovyan. *Problems and methods of cryptology.* S.-Petersburg University Press, S.-Petersburg, 1998. (in Russian).

[96] N.A. Moldovyan, A.A. Moldovyan, and M.E. Eremeev. *Cryptology. From primitives to syntez of algorithms.* S.-Petersburg University Press, S.-Petersburg, 2004. (in Russian).

[97] Yu. Movsisyan. Hyperidentities in algebras and varieties. *Russ. Math. Surv.*, 53(1):57–108, 1998.

[98] G.L. Mullen and V.A. Shcherbacov. On orthogonality of binary operations and squares. *Bul. Acad. Stiinte Repub. Mold., Mat.*, (2 (48)):3–42, 2005.

[99] V. A. Nosov and A. E. Pankratiev. Latin squares over abelian groups. *Fundamentalnaya i prikladnaya matematika*, 12:65–71, 2006.

[100] E. Ochadkova and V. Snasel. Using quasigroups for secure encoding of file system. In *Conference "Security and Protection of information", Abstract of Talks*, pages 175–181, Brno, May 2001.

[101] Adrian Petrescu. Applications of quasigroups in cryptography. In *"Interdisciplinarity in Engineering" Scientific International Conference Tg.Mures-Romania, 15-16 November 2007*, 2007. www.upm.ro/InterIng2007/Papers/Section6/16-Petrescu-Quasigroups-pVI- 16-1-5.pdf.

[102] H.O. Pflugfelder. *Quasigroups and Loops: Introduction.* Heldermann Verlag, Berlin, 1990.

[103] R.L. Rivest. All-or-nothing encryption and the package transform. In *Fast Software Encryption '97*, volume 1267 of *LNCS*. Springer, 1997.

[104] D.G. Sarvate and J. Seberry. Encryption methods based on combinatorial designs. *Ars Combinatoria*, 21A:237 – 245, 1986.

[105] M. Satti. A quasigroup based cryptographic system. Technical Report CR/0610017, arxiv.org, 2006.

[106] R. Schauffler. *Eine Anwendung zyklischer Permutationen und ihre Theorie.* PhD thesis, Philipps-Universität Marburg, 1948. (in German).

[107] R. Schauffler. Uber die Bildung von Codewörter. *Arch. Elektr. Übertragung*, 10:303 – 314, 1956.

[108] J. R. Seberry and A. P. Street. Strongbox secured secret sharing schemes. *Util. Math.*, 57:147 – 163, 2000.

[109] I. G. Shaposhnikov. Congruences of finite multibase universal algebras. *Diskret. Mat.*, 11(3):48 – 62, 1999.

[110] V.A. Shcherbacov. On linear quasigroups and their automorphism groups. *Mat. Issled.*, 120:104 – 113, 1991. (in Russian).

[111] V.A. Shcherbacov. Elements of quasigroup theory and some its applications in code theory, 2003. www.karlin.mff.cuni.cz/ drapal/speccurs.pdf.

[112] V.A. Shcherbacov. On some known possible applications of quasigroups in cryptology, 2003. www.karlin.mff.cuni.cz/ drapal/krypto.pdf.

[113] V.A. Shcherbacov. *On linear and inverse quasigroups and their applications in code theory, Doctor of Science thesis.* Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova, Chişinău, 2008. cnaa.acad.md/files/theses/2008/8175/victor_scerbacov_thesis.pdf.

[114] P.W. Shor. Quantum computing. In *Proc. Intern. Congress of Mathematicians*, pages 467–486, Berlin, 1998.

[115] G.J. Simmons. *Contemporary Cryptology - The Science of Information Integrity.* IEEE Press, New York, 1992.

[116] V. Snasel, A. Abraham, J. Dvorsky, P. Kromer, and J. Platos. Hash functions based on large quasigroups. In *ICCS 2009, Part I, LNCS 5544*, pages 521–529, Springer-Verlag, Berlin, 2009.

[117] Zoran Stojakovic and Djura Paunic. Self-orthogonal cyclic n-quasigroups. *Aequationes Math.*, 30(2-3):252–257, 1986.

[118] P.N. Syrbu. Self-orthogonal n-ary groups. *Matem. issled.*, 113:99–106, 1990. (in Russian).

[119] P. Vojtechovsky. Distances of groups of prime order. *Contrib. Gen. Algebra*, 11:225–231, 1999.

[120] M. Vojvoda. A survey of security mechanisms in mobile communication systems. *Tatra Mt. Math. Publ.*, 25:109–125, 2002.

[121] M. Vojvoda. Cryptanalysis of one hash function based on quasigroup. *Tatra Mt. Math. Publ.*, 29:173–181, 2004. MR2201663 (2006k:94117).

[122] M. Vojvoda. *Stream ciphers and hash functions - analysis of some new design approaches.* PhD thesis, Slovak University of Technology, July, 2004.

[123] Fajar Yuliawan. Studi mengenai aplikasi teori quasigroup dalam kriptografi, 2006. Program Studi Teknik Informatika, Institut Teknologi Bandung, www.informatika.org/rinaldi/Kriptografi/2006-2007/Makalah1/Makalah1- 037.pdf.

[124] H. Zbingen, N. Gisin, B. Huttner, A. Muller, and W. Tittel. Practical aspects of quantum cryptographical key distributions. *J. Cryptology*, 13:207–220, 2000.

V.A. Shcherbacov,

Received November 4, 2009

Institute of Mathematics and Computer Science
Academy of Sciences of Moldova
Academiei 5, Chişinău MD-2028 Moldova
E–mail: *scerb@math.md*