ON OBDD TRANSFORMATIONS REPRESENTING FINITE STATE AUTOMATA

S.Kryvyy, W.Grzywacz

Abstract: *We present OBDD transformation problem representing finite labeled transition systems corresponding to some congruence relation. Transformations are oriented toward obtaining the OBDD of a minimized transition system for this congruence relation.*

Keywords: *finite automata, OBDD, congruence relation, minimization, transformation.*

Introduction

Ordered Binary Decision Diagrams (OBDDs) are widely used in many domains of computer science. One of the most important applications of OBDDs is compact representation of finite transition systems (TSs), which allows one to extend verification possibilities of TSs (the Model Checking method). An important class of TSs is the class of finite TSs, which is the class of finite state automata in the case where the numbers of states and transitions of a TS is finite.

In this paper, we consider the following problem. Let a finite state automaton *A*, its representation as an OBDD, and a congruence relation *R* on the set of states of the automaton *A* be given. It is necessary to construct an OBDD that represents quotient automaton *A/R*. This problem can be solved in a traditional way as follows: first, to construct the quotient automaton *A/R* and, second, the OBDD that represents this quotient automaton. However, the relation *R* often can be formulated after the representation of the automaton A as an OBDD and, as a result, this OBDD can be easily transformed into the OBDD that represents the *A/R* automaton.

Preliminary notes

A Finite state X-automaton is understood to be a quadruple $A = (A, X, f, F)$, where A is a finite set of states, $X = \{0, 1\}$ is the alphabet of input symbols or the input alphabet of the automaton, $f : A \times X \rightarrow A$ is a transition function, and $F \subseteq A$ is a subset of terminal states.

Let $R \subseteq A \times A$ be some congruence relation with respect to the transition function, i.e., we have $a\, R\, b \Leftrightarrow (\forall x \in X)\, (f(a, x)\, R\, f(b, x))$. The automaton whose set of states consists of equivalent classes of relation *R* is called quotient automaton *A* and is denoted by *A/R*. An important congruence relation is the so-called automata equivalence relation R_A that, for *X*-automata, is defined in the following way:

$$a\, R_A\, b \Leftrightarrow (\forall p \in F(X))\, (f(a, p) \in F \Leftrightarrow f(b, p) \in F),$$

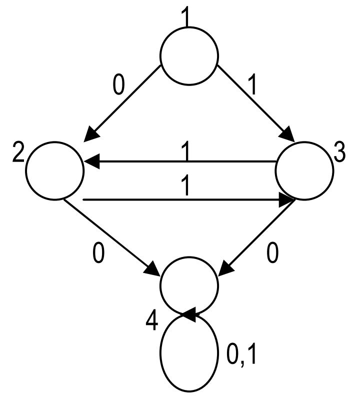
where *F(X)* is the semigroup of all words of finite length in the alphabet *X*, $a, b \in A$, and *F* is the set of final states. An automaton is called reduced or minimal if all its states are pairwise nonequivalent. As is well known, the quotient automaton A/R_A is minimal in the class of all the automata that are equivalent to automaton *A* [4].

An OBDD is a graphical representation (an acyclic graph) of a Boolean function [1]. The use of OBDDs for representation of finite state automata is based on the representation its transition function *f* in the form of a ternary relation R_f defined on codes of states of an automaton. Let, for example, for states $a, b \in A$ and some $x \in X$, we have $f(a, x) = b$. If $\underline{x}$, $\underline{a}$, and $\underline{b}$ are the corresponding codes of a symbol $x \in X$ and states $a, b \in A$, then we have

$$R_f(\underline{x}, \underline{a}, \underline{b}) = \begin{cases} 1, & if \quad f(a,x) = b, \\ 0, & if \quad f(a,x) \neq b, \\ d, & in \quad other \quad cases \end{cases}$$

Now, it is obvious that this form of the relation R_f corresponds to a Boolean function $g_f(\underline{x}, \underline{a}, \underline{b})$ that equals 1 if and only if $R_f(\underline{x}, \underline{a}, \underline{b}) = 1$.

Consider an example. Let A = (A= {1,2,3,4}, X = {0,1}, f, F = {4}) be a finite state X-automaton whose transition function *f* is specified by the transition graph given below.



We assume that the symbols of the input alphabet *X* are coded in a natural way by an identical mapping and that the states of the automaton are coded as follows:

$$1 \to 00,\ 2 \to 01,\ 3 \to 10,\ 4 \to 11.$$

Then the relation $R_f(x, a_1, a_2, b_1, b_2)$ assumes the form:

$$R_f(0,0,0,0,1) = 1,\ R_f(1,0,0,1,0) = 1,\ R_f(0,0,1,1,1) = 1,\ R_f(1,0,1,1,0) = 1,$$
$$R_f(0,1,0,1,1) = 1,\ R_f(1,1,0,0,1) = 1,\ R_f(0,1,1,1,1) = 1,\ R_f(1,1,1,1,1) = 1.$$

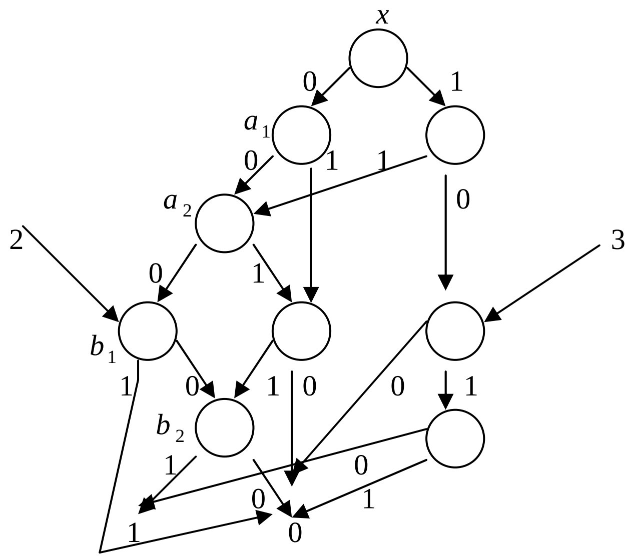
The above relation corresponds to the following Boolean function g_f:

$$g_f(x, a_1, a_2, b_1, b_2) = \bar{x}\bar{a}_1\bar{a}_2\bar{b}_1 b_2 \vee x\bar{a}_1\bar{a}_2 b_1\bar{b}_2 \vee \bar{x}\bar{a}_1 a_2 b_1 b_2 \vee x\bar{a}_1 a_2 b_1\bar{b}_2 \vee$$
$$\vee \bar{x} a_1\bar{a}_2 b_1 b_2 \vee x a_1\bar{a}_2\bar{b}_1 b_2 \vee \bar{x} a_1 a_2 b_1 b_2 \vee x a_1 a_2 b_1 b_2,$$

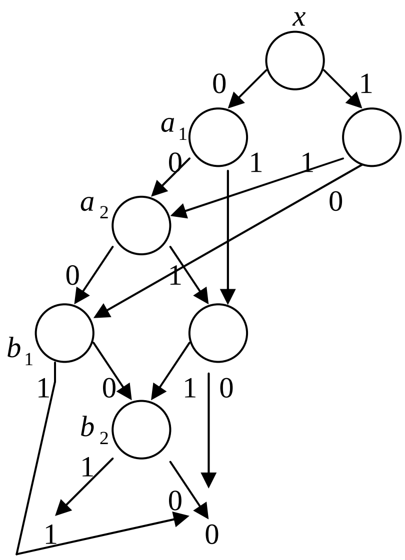
where x, a_1, a_2, b_1, and b_2 are codes that represent input symbols from the alphabet X and a and b are states from A, (we write a_i if $a_i = 1$ and $\bar{a}_i$ if $a_i = 0$, the same is true for b_i, i = 1,2).

Transformation of OBDD representing of non minimal automaton

We now describe transformations of OBDD that lead to the representation of the minimal A/R_A automaton. Let $K_1, K_2, \ldots, K_m$ be the equivalence classes of the set A in the sense of relation R_A. We can choose one representative element in every class K_i, i = 1,..., m. Let it be states $b_1, b_2, \ldots, b_m$, where $b_i \in K_i$. These states can be represented using the following codes: $b_{11} b_{12} \ldots b_{1k}, b_{21} b_{22} \ldots b_{2k}, \ldots, b_{m1} b_{m2} \ldots b_{mk}$, where $k = \log |A|$. The first step of the transformation is based on the reorganization of paths in the acyclic diagram representing OBDD. All the paths leading to a state from K_i must be replaced by ones that lead to their representative element. At the next step, we apply standard OBDD operations to remove unused nodes. At the third step, we introduce new nodes since it is necessary to correctly reorient the transitions so that unused nodes are eliminated. We do not have to introduce new nodes if there are transitions that make it possible to reach states from a class K_i, i = 1, ..., m (that are equivalent to state b_i). In the minimal automaton, these states will be eliminated. The described method will be explained using an example given below. The figure below presents the initial OBDD graph.

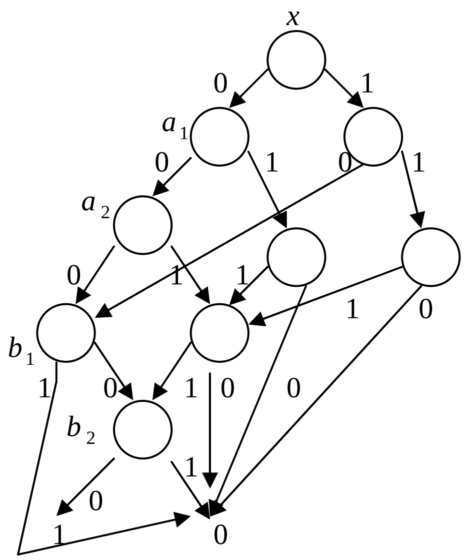


In this automaton, states 2 and 3 (see the above figure) are equivalent. Their codes are 01 and 10, respectively. We choose, for example, state 2 as a representative element in the equivalence class. We perform the first step and reorganize paths that lead to state 2. At the same time, we eliminate unused nodes. As a result, we obtain the OBDD presented below:

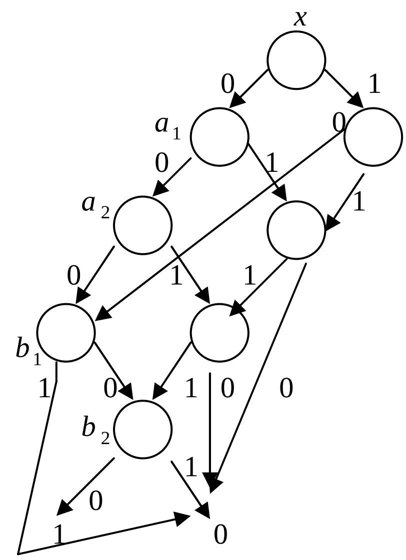


This graph does not correspond to the minimal automaton yet. This is because all the transitions to state 3 are eliminated, but the transitions outgoing from state 3 are not eliminated. To eliminate the latter transitions, it is necessary to introduce a new node (in this case, only one node). The explanation is as follows:

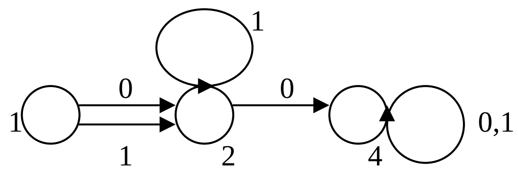
In the OBDD, 2 nodes exist that correspond to the unessential transition $xa_1 \overline{a}_2 \overline{b}_1 b_2$. These nodes are labeled by a_1. For these nodes, we introduce a new node labeled by a symbol a_2 and eliminate unessential transitions. All these transformations are showed below.



After an evident reduction, we obtain the final OBDD that represents the minimal automaton A/ R_A of the automaton A.



To show that this OBDD really represents A/ R_A automaton, we present the following automaton obtained from the above diagram:



It is obvious that the automaton obtained is a reduced automaton of the automaton A.

It follows from the above that the obtaining of the OBDD of the minimal automaton is reduced to the following transformations:

(a) elimination of transitions leading to equivalent states, which reduces to changing the paths leading to this state to lead to node, which is main representative state of equivalence class;

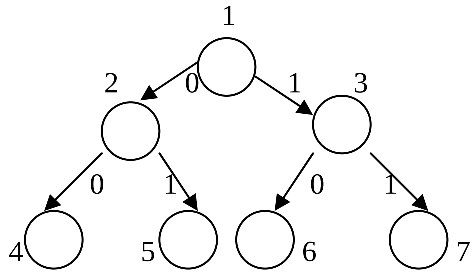
(b) elimination of transitions from equivalent states with eventual reproducing of necessary nodes in OBDD;

(c) elimination of unavailable nodes in OBDD graph.

Acyclic automata

The OBDD transformations described above can be applied to OBDD transformations representing an acyclic automaton. In this case, the relation R need not be a congruence relation, it must be an equivalent relation. To

obtain a reduced automaton, a congruent closure R^* of the relation R should be constructed. In this case, the closure R^* is defined as follows: $a\, R^*\, b \Leftrightarrow (a\, R\, b) \vee (\forall\ x \in X)\ (f(a, x)\ R^*\ f(b, x))$.
If an equivalence relation R is given, then we can obtain a reduced automaton from the OBDD.
In the general case, the relation R^* can lead to a cycle. To assure the acyclicity of R^*, we will consider the case where R relates the states on the same level. This constraint guarantees the acyclicity of relation R^* and quotient automaton A/R^*. All the above facts are illustrated by the following example of acyclic automaton A whose graph is presented below:



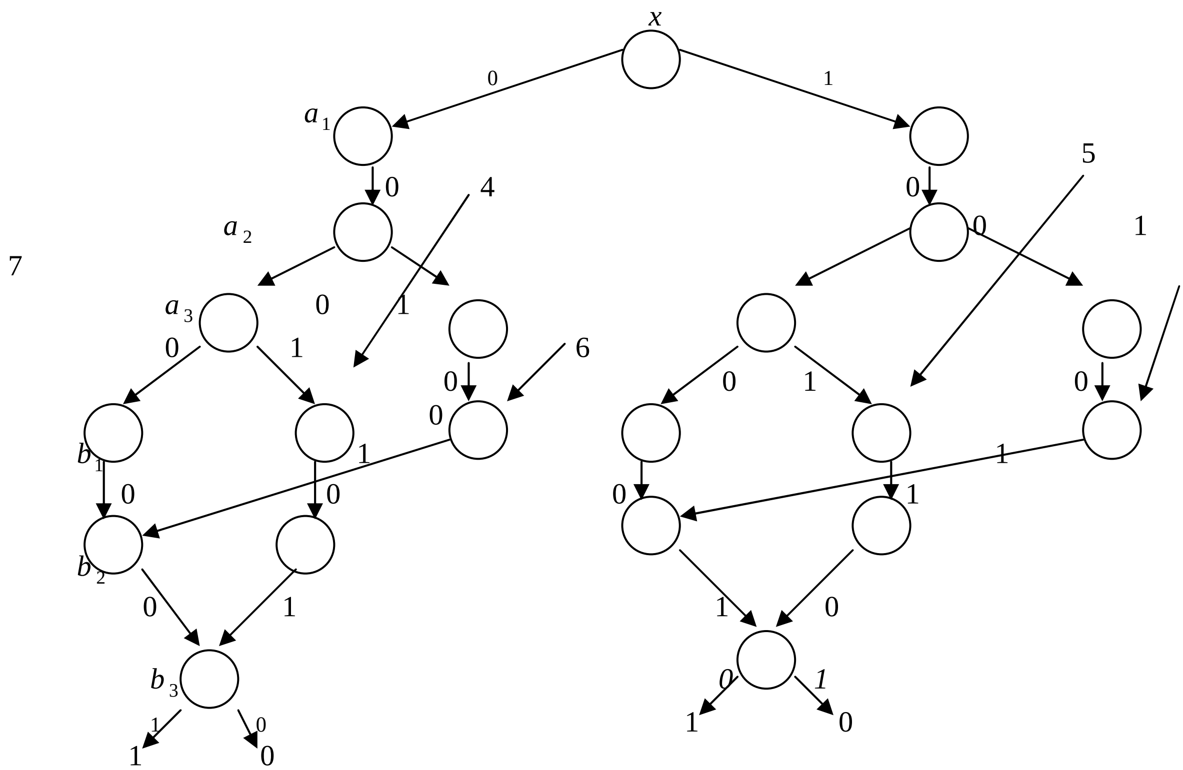
Let the states of this automaton are coded as follows:

$$1 \to 000,\ 2 \to 001,\ 3 \to 010,\ 4 \to 011,\ 5 \to 100,\ 6 \to 101,\ 7 \to 110.$$

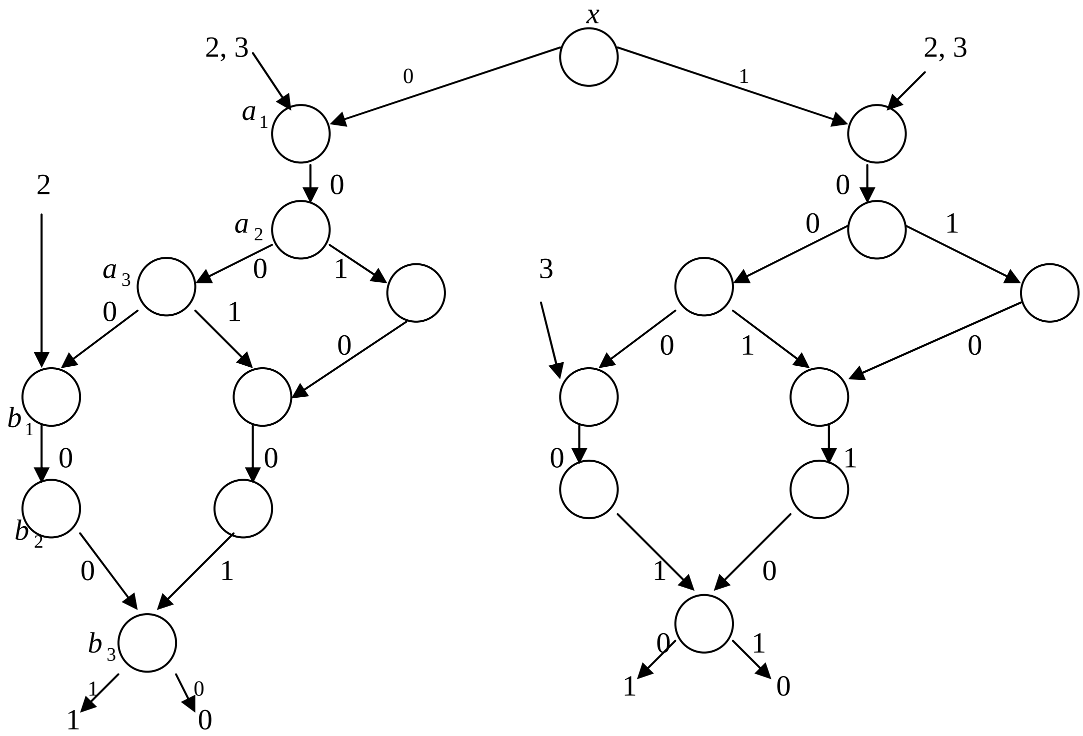
A Boolean function g_f corresponding to the transition relation R_f assumes the form

$$g_f\,(x, a_1, a_2, a_3, b_1, b_2, b_3) = \bar{x}\,\bar{a}_1\,\bar{a}_2\,\bar{a}_3\,\bar{b}_1\,\bar{b}_2\,b_3 \vee x\bar{a}_1\,\bar{a}_2\,\bar{a}_3\,\bar{b}_1 b_2\,\bar{b} \vee$$
$$\vee\ \bar{x}\,\bar{a}_1\,\bar{a}_2\,a_3\,\bar{b}_1 b_2 b_3 \vee x\bar{a}_1\,\bar{a}_2\,a_3\,b_1\bar{b}_2\,\bar{b}_3 \vee \bar{x}\,\bar{a}_1 a_2\,\bar{a}_3 b_1\bar{b}_2\,b_3 \quad \vee$$
$$x\bar{a}_1 a_2\,\bar{a}_3 b_1 b_2\,\bar{b}_3.$$

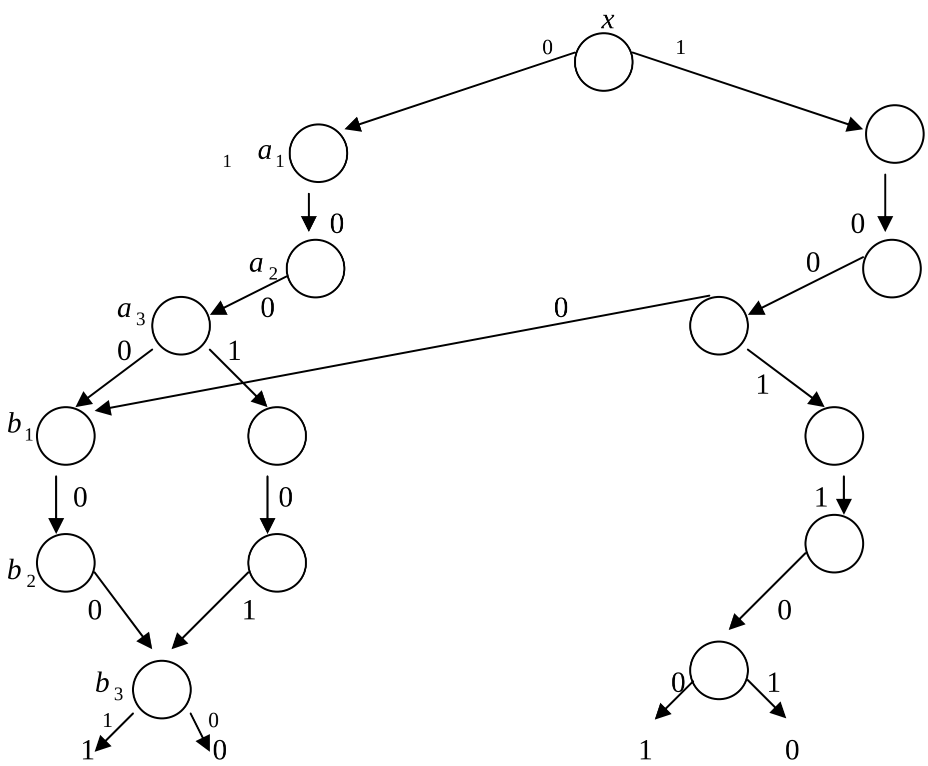
The OBDD that corresponds to this function assumes the form (to reduce the size of the diagram, we assume that all the missing paths lead to the node labeled by 0):



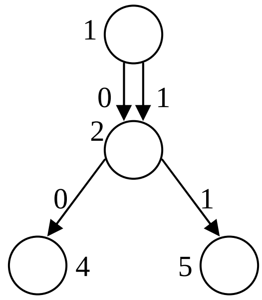
Let the relation R = {(4,6), (5,7)} be given. Then we choose the representatives of the equivalence classes, e.g., states 4 and 5, respectively, and reorient the transitions to these states. Thus, after this step of transformation, we obtain the following diagram:



As a result of this transformation, we obtain that the states 2 (001) and 3 (010) are equivalent. We choose state 2 as a representative of the equivalence class and eliminate the transitions from and to state 3. Repeating the first transformation step, we obtain the following OBDD:



The OBDD presented above corresponds to the following acyclic automaton:



Conclusions

The method presented in the paper allows one to reduce the number of nodes in OBDDs. The use of an equivalence relation instead of a congruence relation allows us to work with acyclic graphs. This makes it possible to simplify the solution of the common subexpressions problem.

This work on OBDDs deals with algorithms of minimization of finite state automata. This research shows the expediency of transformations of OBDDs.

Bibliography

[Bryant R.E., 1] Bryant R.E. Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams. Scool of Computer Science, Carnegi Mellon University, Pittsburg. 1992 (june). - 34 P.

[Clarke E.M., Schlingloff B.-H., 2] Clarke E.M., Schlingloff B.-H. Model Checking. In ed. A. Robinson and A. Voronkov Handbook of Automated Reasoning, Elsevier Science Publishers B.V. - 2001. - P. 1369 - 1522.

[McMillan K.I.,3] McMillan K.I. Symbolic Model Checking: an approach to state explosion problem. PhD thesis. Scool of Computer Science, Carnegi Mellon University. -1992. - 212 P.

[Glushkov V.M., Leticievskij A.A., Godlievskij A.B., 4] Glushkov V.M., Lieticievskij A.A., Godlievskij A.B. Mathematical biology methods. Book 6. Methods of synthesis of discret biological systems. Kiev: University. - 1983. - 264 pages.

Information about authors

Krivoi Sergey – Glushkov Institute of Cybernetics NSA Ukraine, Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: krivoi@i.com.ua

Grzywacz Wioletta – Technical University of Czestochowa, Institute of Theoretical and Aplied Informatics, Czestochowa, Poland, e-mail: wiola@icis.pcz.czest.pl