The Visualization of the Road Coloring Algorithm in the package TESTAS

A.N. Trahtman, T. Bauer, N. Cohen

Bar-Ilan University, Dep. of Math., 52900, Ramat Gan, Israel

Abstract

A synchronizing word of a deterministic automaton is a word in the alphabet of colors of its edges that maps the automaton to a single state. A coloring of edges of a directed graph is synchronizing if the coloring turns the graph into a deterministic finite automaton possessing a synchronizing word.

The road coloring problem is the problem of synchronizing coloring of a directed finite strongly connected graph with constant outdegree of all its vertices if the greatest common divisor of the lengths of all its cycles is one. A polynomial time algorithm of the road coloring has been based on recent positive solution of this old famous problem.

One can use our new visualization program for demonstration of the algorithm as well as for visualization of the transition graph of any finite automaton. The visual image presents some structure properties of the transition graph. This help tool is linear in the size of the automaton.

The road coloring problem originates in [2] and was stated explicitly in [1] by Adler, Goodwyn and Weiss over 30 years ago. The problem appears first in the context of symbolic dynamics for a complete strongly connected directed finite graph with constant outdegree of all its vertices where the greatest common divisor (gcd) of lengths of all its cycles is one. The edges of the graph being unlabelled, the task is to find a labelling of the edges that turns the graph into a deterministic finite automaton possessing a synchronizing word. So the road coloring problem is connected with the problem of the existence of synchronizing word for deterministic complete finite automata.
The road coloring problem is important in automata theory: a synchronizing coloring makes the behavior of an automaton resistant against input errors since, after detection of an error, a synchronizing word can reset the automaton back to its original state, as if no error had occurred.

Together with the Černý conjecture [12], [17], the road coloring problem was belong to the most fascinating problems in the theory of finite automata. The problem was mentioned in the popular Internet Encyclopedia ”Wikipedia” on the list of most interesting unsolved problems in mathematics. However, at the same time it was considered as a ”notorious open problem” [11], [5] and ”unfeasible” [9].

The solution of the road coloring problem [16], [17] is a basis of a polynomial time implemented algorithm of $O(n^3)$ complexity in the most worst case and quadratic in majority of cases. Some results from [16] are strengthened below or presented in another wording. The task of the algorithm is to check the necessary conditions of synchronizing coloring on an arbitrary graph and in positive case find a labelling of the edges that turns the graph into a deterministic finite automaton possessing a synchronizing word.

The realization of the algorithm is demonstrated by a new visualization program. It can analyze any kind of input graph. Crucial role in the visualization plays for us the correspondence of the layout to the human intuition, the perception of the structure properties of the graph and the rapidity of the appearance of the image. We use for this aim some known approaches [14], [18] together with some new productive ideas. The algorithm of the visualization is linear in the size of the automaton. The visualization of the transition graph of the automaton is a help tool of the study of the automata and therefore the linearity of the algorithm is comfortably and important.

**Preliminaries**

A finite directed strongly connected graph with constant outdegree of all its vertices where the gcd of lengths of all its cycles is one will be called AGW graph as aroused by Adler, Goodwyn and Weiss.

Let $A = (P, \Sigma, \delta)$ be a deterministic finite automaton, where $P$ denotes the state set, $\Sigma$ stands for the input alphabet, and $\delta : P \times \Sigma \rightarrow P$ is the transition function defining an action of the letters in $\Sigma$ on $P$. The transition function defines on the set of vertices $P$ the transition graph of the automaton.
We denote by $|P|$ the size of the subset $P$ of states of an automaton (of vertices of a graph).

If there exists a path in an automaton from the state $p$ to the state $q$ and the edges of the path are consecutively labelled by $\sigma_1, ..., \sigma_k$, then for $s = \sigma_1...\sigma_k \in \Sigma^+$ we shall write $q = ps$.

Let $Ps$ be the set of the states $ps$ for $p \in P$ and $s \in \Sigma^+$. For the transition graph $\Gamma$ of an automaton let $\Gamma s$ denote the map of the set of states of the automaton.

A word $s \in \Sigma^+$ is called a synchronizing word of the automaton with transition graph $\Gamma$ if $|\Gamma s| = 1$.

A coloring (a labelling of edges by colors) of a directed finite graph is synchronizing if the coloring turns the graph into a deterministic finite automaton possessing a synchronizing word.

The bold letters will denote the vertices of a graph and the states of an automaton.

A pair of distinct states $p, q$ of an automaton (of vertices of the transition graph) will be called synchronizing if $ps = qs$ for some $s \in \Sigma^+$. In the opposite case, if for any $s \in \Sigma^+$ $ps \neq qs$, we call the pair a deadlock.

A synchronizing pair of states $p, q$ of an automaton is called stable if for any word $u$ the pair $pu, qu$ is also synchronizing [5], [10].

We call the set of all outgoing edges of a vertex a bunch if all these edges are incoming edges of only one vertex.

The subset of states (of vertices of the transition graph $\Gamma$) of maximal size such that every pair of states from the set is a deadlock will be called an $F$-clique.

1 Some properties of $F$-cliques and of stable pairs

The road coloring problem was formulated for AGW graphs [1] and only such graphs are considered below.

Let us recall that a binary relation $\rho$ on the set of the states of an automaton is called congruence if $\rho$ is equivalence and for any word $u$, from $p \rho q$ follows $pu \rho qu$. Let us formulate an important result from [10], [5] in the following form:
Theorem 1 Let us consider a coloring of an AGW graph $\Gamma$. The stability of states is a binary relation on the set of states of the obtained automaton. Let us denote this relation by $\rho$.

Then $\rho$ is a congruence relation, $\Gamma/\rho$ is an AGW graph and a synchronizing coloring of $\Gamma/\rho$ implies synchronizing recoloring of $\Gamma$.

Lemma 1 Let $F$ be an $F$-clique of some coloring of an AGW graph $\Gamma$. For any word $s$ the set $Fs$ is also an $F$-clique and any state $p$ belongs to some $F$-clique.

Lemma 2 Let $A$ and $B$ (with $|A| > 1$) be distinct $F$-cliques of some coloring of an AGW graph $\Gamma$ such that $|A| - |A \cap B| = 1$. Then for all states $p \in A \setminus A \cap B$ and $q \in B \setminus A \cap B$ the pair $(p, q)$ is stable.

Proof. By the definition of an $F$-clique, $|A| = |B|$ and $|B| - |A \cap B| = 1$, too. If the pair of states $p \in A \setminus B$ and $q \in B \setminus A$ is not stable, then, for some word $s$, the pair $(ps, qs)$ is a deadlock. Any pair of states from the $F$-clique $A$ and from the $F$-clique $B$, as well as from the $F$-cliques $As$ and $Bs$, is a deadlock. So any pair of states from the set $(A \cup B)s$ is a deadlock. One has $|(A \cup B)s| = |As| + 1 = |A| + 1 > |A|$. So the size of the set $(A \cup B)s$ of deadlocks is greater than the maximal size of any $F$-clique. Contradiction.

Lemma 3 If some vertex of AGW graph $\Gamma$ has two incoming bunches then the beginnings of the bunches form a stable couple by any coloring.

2 The spanning subgraph of AGW graph

Definition 1 Let us call a subgraph $S$ of an AGW graph $\Gamma$, a spanning subgraph of $\Gamma$, if $S$ contains all vertices of $\Gamma$ and if each vertex has exactly one outgoing edge.

A maximal subtree of a spanning subgraph $S$ with its root on a cycle of $S$ and having no common edges with the cycles of $S$ is called a tree of $S$.

The length of path from a vertex $p$ through the edges of the tree of the spanning set $S$ to the root of the tree is called the level of $p$ in $S$.

A tree with vertex of maximal level let us call a maximal tree.

Remark 1 Any spanning subgraph $S$ consists of disjoint cycles and trees with roots on cycles. Any tree and cycle of $S$ is defined identically. The
level of the vertices belonging to some cycle is zero. The vertices of the trees except the roost have positive level. The vertices of maximal positive level have no incoming edge in $S$. The edges labelled by a given color defined by any coloring form a spanning subgraph. Conversely, for any spanning subgraph, there exists a coloring and a color such that the set of edges labelled with this color corresponds to this spanning subgraph.

**Lemma 4** [16] Let $N$ be a set of vertices of maximal level in some tree of the spanning subgraph $S$ of an AGW graph $\Gamma$. Then, via a coloring of $\Gamma$ such that all edges of $S$ have the same color $\alpha$, for any $F$-clique $F$ holds $|F \cap N| \leq 1$.

**Lemma 5** [16] Let $\Gamma$ be an AGW graph which is a union of cycles (without trees). Then the non-trivial graph $\Gamma$ has also a spanning subgraph with exactly one maximal tree.

**Lemma 6** Let us assume that any vertex of an AGW graph $\Gamma$ has not two incoming bunches. Let $R$ be a spanning subgraph of $\Gamma$. Let $T$ be a maximal tree with a vertex $p$ of maximal positive level and let $r$ be its root which belongs to a cycle $H$ of $R$. Let us consider the following transformations

1) replacing an edge from $R$ by an incoming edge of $p$ with the same origin,

2) replacing an incoming edge of $r$ which belongs to the path from $p$ to $r$ in $T$.

3) replacing an incoming edge of $r$ which belongs to the cycle $H$.

Then after a finite number of the above transformations in the new spanning subgraph either the number of edges in cycles is growing or $R$ has a unique maximal tree.

Proof. Let the edge $\bar{b} = b \rightarrow r \in T$ belongs to the path of the maximal length $L$ from $p$ to $r$ in $T$. Suppose $\bar{c} = c \rightarrow r \in H$. There exists also an edge $\bar{a} = a \rightarrow p$ that does not belong to $R$ because $\Gamma$ is strongly connected and $p$ has no incoming edge in $R$. Let $w = a \rightarrow d$ belongs to $R$.  

5
We consider the path from $p$ to $r$ of maximal length $L$ in $T$. Our aim is to increase the maximal level with an extension of the tree $T$ much more than the maximal level of vertex of other trees from $R$.

We plan to use the three aforesaid ways. If one of the ways does not succeed, let us go to the next, assuming that the situation in which the previous fails, and excluding the successfully studied cases. Let us begin with

1) Suppose first $a \not\in H$. If $a$ belongs to the path in $T$ from $p$ to $r$ then a new cycle with part of the path and the edge $a \to p$ is added to $R$ extending the number of vertices in its cycles. In the opposite case the level of $a$ is $L + 1$ in the new spanning subgraph and the vertex $r$ is the root of the new tree containing all vertices of maximal level (the vertex $a$ or its ancestors in $R$).

So let us assume $a \in H$ and suppose $\bar{w} = a \to d \in H$. In this case the vertices $p$, $r$ and $a$ belong to a cycle $H_1$ obtained by removing $\bar{w}$ and adding $a$. We denote by $R_1$ the new spanning subgraph. So we have the cycle $H_1 \in R_1$ instead of $H \in R$. If the length of the path from $r$ to $a$ in $H$ is $r_1$ then $H_1$ has length $L + r_1 + 1$. A path from $d$ to $r$ in the cycle $H$ remains in $R_1$. Suppose that its length is $r_2$. So the length of the cycle $H$ is $r_1 + r_2 + 1$.

If the length of the cycle $H_1$ is greater than the length of $H$ then the number of edges in the cycles of the spanning subgraph grows. In the opposite case one has $r_1 + r_2 + 1 \geq L + r_1 + 1$, whence $r_2 \geq L$. If $r_2 > L$, then the length $r_2$ of the path from $d$ to $r$ in a tree of $R_1$ (equal to the level of $d$) is greater than $L$. The tree containing $d$ (or some ancestor of $d$ in a tree of $R_1$) is the desired unique maximal level.

So we can assume for further consideration that $L = r_2$ and $a \in H$. Analogously, for any vertex of maximal level which belongs to a tree whose root is in $H$ and has an incoming edge starting at a vertex $a_1$, the proof can be reduced to the case $a_1 \in H$ and $L = r_2'$ for the corresponding value $r_2'$.

2) Suppose that the set of outgoing edges of the vertex $b$ is not a bunch. So one can replace in $R$ the edge $\bar{b}$ from the vertex $b$ by an edge $\bar{v}$ from $b$ to
a vertex \( v \neq r \).

If the vertex \( v \) belongs to \( T \) then a new cycle is added to \( R \), whence the number of vertices in the cycles of the spanning subgraph grows.

If the vertex \( v \) belongs to another tree of \( R \) but not to a cycle, then \( T \) is a part of a new tree \( T_1 \) with a new root of a new spanning subgraph \( R_1 \) and the path from \( p \) to the new root has a length greater than \( L \).

Therefore the tree \( T_1 \) is the unique maximal tree in \( R_1 \).

If \( v \) belongs to some cycle \( H \neq H \) in \( R \), then together with replacing \( \bar{b} \) by \( \bar{v} \), we also replace the edge \( \bar{w} \) by \( \bar{a} \). So we extend the path from \( p \) to the new root \( v \) of \( H_2 \) at least by the edge \( \bar{a} = a \rightarrow p \) and there is a unique maximal tree of level \( L_1 > L \) which contains either the vertex \( d \) or some of its ancestors in the old spanning subgraph \( R \).

Let \( r'_2 \) be the length of the path from \( d \) to \( v \in H \).

For the spanning subgraph \( R_1 \), one can obtain \( L = r'_2 \) just as it was done in the step 1) for \( R \). From \( v \neq r \) follows \( r'_2 \neq r_2 \), though \( L = r'_2 \) and \( L = r_2 \).

So for further consideration suppose that the set of outgoing edges of the vertex \( b \) is a bunch to \( r \).

3) The set of outgoing edges of the vertex \( c \) is not a bunch because \( r \) has another bunch from \( b \) in virtue of the lemma condition.

Let us replace in \( R \) the edge \( \bar{c} \) by an edge \( \bar{u} = c \rightarrow u \) such that \( u \neq r \). The vertex \( u \) could not belong to the tree \( T \) because one has in this case a cycle with all vertices from \( H \) and some vertices of \( T \) whence its length is greater than \( |H| \) and so the number of vertices in the cycles of a new spanning subgraph grows.

If the vertex \( u \) does not belong to \( T \), then the tree \( T \) is a part of a new tree with a new root. The path from \( p \) to the new root is extended at least by a part of \( H \) starting at the former root \( r \). The new level of \( p \) therefore is greater than the level of any vertex in another tree.

**Lemma 7** Let \( \Gamma \) be an AGW graph such that no vertex has two incoming bunches. Let \( R \) be a spanning subgraph which has a unique maximal tree \( T \) with a root on a cycle \( C \). Let \( p \) be some vertex of \( T \) of maximal level \( L \).

Then coloring \( R \) with some color \( \alpha \) makes the pair \( p\alpha^{L-1},\ p\alpha^{L+|C|-1} \) stable.
Proof. Let us color the edges of \( R \) with the color \( \alpha \) and color the remaining edges of \( \Gamma \) by other colors arbitrarily.

By Lemma 1 since \( \Gamma \) is a strongly connected graph, for every word \( s \) and any \( F \)-clique \( F \) of size \( |F| > 1 \), the set \( F \)'s also is an \( F \)-clique of the same size and for any state \( p \) there exists an \( F \)-clique \( F \) such that \( p \in F \).

In particular, some \( F \)-clique \( F \) has a non-empty intersection with the set \( N \) of vertices of maximal level \( L \). The set \( N \) belongs to the tree \( T \), whence by Lemma 4 this intersection has only one vertex. The word \( \alpha^{L-1} \) maps \( F \) on an \( F \)-clique \( F_1 \) of size \( |F| \). One has \( |F_1 \setminus C| = 1 \) because any sequence of length \( L-1 \) of edges of color \( \alpha \) in any tree of \( R \) leads to the cycle \( C \). For the set \( N \) of vertices of maximal level holds \( N \alpha^{L-1} \not\subseteq C \). So \( |N \alpha^{L-1} \cap F_1| = |F_1 \setminus C| = 1 \), \( p \alpha^{L-1} \in F_1 \setminus C \) and \( |C \cap F_1| = |F_1| - 1 \).

Let the integer \( m \) be a common multiple of the lengths of all considered cycles colored by \( \alpha \). So for any \( r \) in \( C \) as well as in \( F \cap C \) holds \( r \alpha^m = r \). Let \( F_2 \) be the \( F \)-clique \( F_1 \alpha^m \). We have \( F_2 \subseteq C \) and \( C \cap F_1 = F_1 \cap F_2 \).

Thus the two \( F \)-cliques \( F_1 \) and \( F_2 \) of size \( |F_1| > 1 \) have \( |F_1| - 1 \) common vertices. So \( |F_1 \setminus (F_1 \cap F_2)| = 1 \), whence by Lemma 2 the pair of states \( p \alpha^{L-1} \) from \( F_1 \setminus (F_1 \cap F_2) \) and \( q \) from \( F_2 \setminus (F_1 \cap F_2) \) is stable. Evidently that \( q = p \alpha^{L+m-1} = p \alpha^{L+|C|-1} \).

Theorem 2 [16] Every AGW graph has a synchronizing coloring.

Theorem 3 [16] Let every vertex of a strongly connected directed graph \( \Gamma \) have the same number of outgoing edges. Then \( \Gamma \) has synchronizing coloring if and only if the greatest common divisor of lengths of all its cycles is one.

Lemma 8 Let \( \Gamma \) be an AGW graph such that two cycles \( C_u \) and \( C_v \) either with one common vertex \( p_1 \) or with a common sequence \( p_1, \ldots, p_k \), such that all incoming edges of \( p_i \) form a bunch from \( p_{i+1} \) \( (i < k) \). Let \( u \in C_u \) and \( v \in C_v \) be the edges of the cycles leaving \( p_1 \). Let \( T \) be a maximal subtree of \( \Gamma \) whose root is \( p_1 \) and whose edges are the union of \( C_u \) and \( C_v \) except \( u \) and \( v \).

Then the subtree \( T \) obtained by adding one of the edges \( u \) or \( v \) turns in a spanning subgraph with unique maximal tree.

Proof. Let us add to \( T \) either \( u \) or \( v \) and then find the maximal level of vertices in both these cases. The vertex \( p_i \) for \( i > 1 \) cannot be the root of a tree. If any tree of spanning subgraph with vertex of maximal level has the
root $p_1$ then the lemma holds in any of both these opportunities. If some tree of the spanning subgraph with some vertex of maximal level has its root only on $C_u$ [resp.$C_v$] then let us choose the addition of $v$ [resp. $u$]. So the level of the considered vertex is growing. The new tree with root $p_1$ is the unique maximal tree.

3 The algorithm for synchronizing coloring

Let us start with an arbitrary coloring of an AGW graph $\Gamma$ with $n$ vertices and constant outdegree $d$. The considered $d$ colors define $d$ spanning subgraphs of the graph.

We keep images of vertices and colored edges in a generic graph by any transformation and homomorphism.

If there exists a loop in $\Gamma$ around a state $r$, then let us color the edges of a tree with the same color as the color of the loop. The other edges may be colored arbitrarily. The coloring is synchronizing [1].

In the case of two incoming bunches of a same vertex the beginnings of these bunches form a stable pair ($p, q$) by any coloring (Lemma 3). We merge $p$ and $q$ in the homomorphic image of the graph (Theorem 1) and obtain according the theorem a new AGW graph of the size $|\Gamma| - 1$.

The linear search of two incoming bunches and of loop can be made at any stage of the algorithm.

1) If there are two cycles with one common vertex (or sequence from Lemma 8) then we use this lemma and find a spanning subgraph $R_1$ such that any vertex $p$ of maximal level $L$ belongs to one tree whose root belongs to a cycle $H$. Then after coloring edges of $R_1$ by color $\alpha$ we find a stable pair $q = p\alpha^{L-1+|H|}$ and $s = p\alpha^{L-1}$ (Lemma 7) and go to the step 3).

2) Let us consider the spanning subgraph of every given color and find its parameters: levels of all vertices, the number of vertices (edges) in cycles, for any vertex let us keep its tree and the cycle attached to the root of the tree. We form the set of vertices of maximal level and choose from the set of trees a tree $T$ containing a vertex $p$ of maximal level. This step is linear and used by any recoloring step. If $T$ is unique maximal tree the pair of ancestors of the root ($p, s$) is stable (Lemma 7) and let us go to the step 3).

Let us consider now the three replacements from Lemma 6 and find the number of edges of the cycles and other parameters of the spanning subgraph of the given color. If the number of edges of the cycles is growing, then the
new spanning subgraph must be considered and the new parameters of the subgraph must be found. In the opposite case, after at most $3d$ steps, by Lemma 6, there exists a unique maximal tree $T_1$ of a spanning subgraph $R_1$.

Let us color the edges of $R_1$ by color $\alpha$ and finish the coloring sooner. The pair of vertices $q = p\alpha^{L-1+|H_1|}$ and $s = p\alpha^{L-1}$ is stable (Lemma 7).

3) Let us find the subsequent stable pairs of the stable pair $(s, q)$ using appropriate recoloring. Then we go to the homomorphic image $\Gamma_i/\rho$ (Theorem 1) of considered graph $\Gamma_i$ (with a $O(|\Gamma_i|m_i d)$ complexity where $m_i$ is the size of the graph $\Gamma_i$). Then we repeat the procedure with the new graph $\Gamma_{i+1}$ which has a smaller size. So the overall complexity of this step of the algorithm is $O(n^2 d)$ in majority of cases and $O(n^3 d)$ in the worst case when the number of edges in cycles grows slowly, $m_i$ decreases also slowly, loops do not appear and the case of two ingoing bunches appears rarely.

Let $\Gamma_{i+1} = \Gamma_i/\rho_{i+1}$ at some stage $i + 1$ has a synchronizing coloring. For every stable pair $q, p$ of vertices from $\Gamma_i$ there exists a pair of corresponding outgoing edges that reaches either another stable pair or one vertex. This pair of edges is mapped on one image edge in $\Gamma_{i+1}$. So let us give the color of the image to preimages and obtain on this way a synchronizing coloring of $\Gamma_i$. This step is linear. So the overall complexity of the algorithm is $O(n^3 d)$ in the most worst case and $O(n^2 d)$ (or less) in most cases.

4 Visualization algorithm

Our main objective is the visual representation of the transition graph of a deterministic finite automaton on the basis of structure properties of the graph. Any kind of finite directed deterministic graph is accepted. A crucial role in the visualization plays the correspondence of the layout to the human intuition, the perception of the structure properties of the graph and the rapidity of the appearance of the image. The graphical image created by computer must resemble the last one of the human being. The considered visualization is a help tool for any program dealing with transition graph of DFA and also for the road coloring algorithm.

Therefore we have changed some priorities of the layout and, in particular, eliminate the goal of reducing the number of intersection of the edges as it was an important in some algorithms [15], [13]. The intersections of the edges are even not considered in the algorithm. Nevertheless the amount of the intersections is relatively less and they are far from the vertices because
of the circular approach [18], [14] we use. The intersections must only not to stir to imagine the structure of the graph.

Among the important visual properties of the graph and its inner structure one can mention paths, cycles and strongly connected components of the graph. We consider as an important aim the eduction and selection of the strongly connected components. It is clear that the curve edges (used, for instance, in the package GraphViz [7], [13]) hinder to recognize the cycles and paths and we use therefore only direct and desirably short edges. The problem of the placing of the labels near corresponding edges is sometimes very complicated and frequently the connection between edge and its label is unclear. Our solution is to use colors on the edges instead of labels and exclude the placing of labels.

The visual placement is based on the inner structure of the graph considered as a union of the set of strongly connected components. The quick linear algorithm for finding SCC [3] is implemented in the program. The vertices of every SCC are presented by a cycle in the graph layout and therefore strongly connected components can be easily recognized by the observer. The periphery of a circle of SCC is the most desirable area for placing the edges because the edges in this case are short and we choose the order of the vertices on the circle according to this purpose. All SCC are placed on the periphery of a big circle. so the pictorial diagram demonstrates the inner structure of the graph.

Lemma 9  The visualization algorithm described above is linear in the sum of states and edges of the transition graph of the automaton.

Proof. The algorithm for finding strongly connected components (SCC) of the graph is linear in the sum of states and edges [3]. One can find in a linear time the order of vertices on a SCC circle for to place just a little more edges on the periphery of the circle. The step for finding on the big circle of the graph the places for all SCC together with radius of the big circle is linear in the number of SCC. The placing of direct edge is defined only by its borders and does not change the complexity.

The linearity of the algorithm ensures the momentary appearance of the layout. See package TESTAS (www.cs.biu.ac.il/~trakht/syn.html).
References

[1] R.L. Adler, L.W. Goodwyn, B. Weiss. *Equivalence of topological Markov shifts*, Israel J. of Math. 27(1977), 49-63.

[2] R.L. Adler, B. Weiss. *Similarity of automorphisms of the torus*, Memoirs of the Amer. Math. Soc., Providence, RI, 98(1970).

[3] A. Aho, J. Hopcroft, J. Ulman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[4] M.P. B'eat, D. Perrin, *A quadratic algorithm for road coloring*, arXiv: 0803.0726v2 [cs.DM].

[5] K. Culik II, J. Karhumaki, J. Kari. *A note on synchronized automata and Road Coloring Problem*, Developments in Language Theory (5th Int. Conf., Vienna, 2001), Lecture Notes in Computer Science, 2295(2002), 175-185.

[6] J. Černy. *Poznamka k homogenym eksperimentom s konechnymi automatami*, Math.-Fyz. Čas., 14(1964) 208-215.

[7] J. Ellson, E. Gansner, L. Koutsofios, et al., *GraphViz - open source graph drawing tools*, Graph Drawing, 2265, 2002, 483-484.

[8] P. Frankl. *An extremal problem for two families of sets*, Eur. J. Comb., 3(1982), 125-127.

[9] R. Hegde, K. Jain, *Min-Max theorem about the Road Coloring Conjecture* EuroComb 2005, DMTCS proc., AE, 2005, 279 - 284.

[10] J. Kari. *Synchronizing finite automata on Eulerian digraphs*, Springer, Lect. Notes in Comp. Sci., 2136(2001), 432-438.

[11] D. Lind, B. Marcus. *An Introduction of Symbolic Dynamics and Coding*, Cambridge Univ. Press, 1995.

[12] J.E. Pin. *On two combinatorial problems arising from automata theory*, Annals of Discrete Math., 17(1983), 535-548.

[13] M. Simonato, An Introduction to GraphViz, 2004.
[14] JM. Six, IG. Tollis, A framework for user-grouped circular drawings
Lect. Notes in Comp. Sci., 1731(1999), 107-116.

[15] K. Sugiyama, S. Tagawa, M. Toda: Methods for visual understanding
of hierarchical systems. IEEE Transactions on Systems, Man and Cy-
bernetics, 11, (1981) 109-125.

[16] A.N. Trahtman. *Synchronizing Road Coloring*. 5-th IFIP WCC-TCS,
Springer, 273(2008), 43-53.

[17] A.N. Trahtman. *The Road Coloring and Cerny Conjecture*. Proc. of
Prague Stringology Conference. 2008, 1-12.

[18] R. Wiese, M. Eiglsperger, M. Kaufmann. A framework for circular draw-
ings of networks. Lect. Notes in Comp. Sci., 2265(2002), 453-454.