The Road Coloring and Černy Conjecture

Avraham N. Trahtman

Bar-Ilan University, Dep. of Math., 52900, Ramat Gan, Israel
trakht@macs.biu.ac.il
http://www.cs.biu.ac.il/~trakht

Abstract. A synchronizing word of a deterministic automaton is a word in the alphabet of colors (considered as letters) 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 lengths of all its cycles is one. The problem was posed by Adler, Goodwyn and Weiss over 30 years ago and evoked noticeable interest among the specialists in the theory of graphs, deterministic automata and symbolic dynamics.
The positive solution of the road coloring problem is presented.
Some consequences on the length of the synchronizing word are discussed.

Keywords: road coloring problem, graph, deterministic finite automaton, synchronization

Introduction

The road coloring problem originates in [2] and was stated explicitly in [1] for a 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 are 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 existence of synchronizing word for deterministic complete finite automaton.

The condition on gcd is necessary [1], [6]. It can be replaced by the equivalent property that there does not exist a partition of the set of vertices on subsets V_1, V_2, ..., $V_{k+1} = V_1$ $(k > 1)$ such that every edge which begins in V_i has its end in V_{i+1} [6], [20]. The outdegree of the vertex can be considered also as the size of an alphabet where the letters denote colors.

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. The problem appeared first in the context of symbolic dynamics and is important also in this area.

Together with the Černy conjecture [22], [24], the road coloring problem belongs to the most fascinating problems in the theory of finite automata. The problem was discussed even in "Wikipedia" – the popular Internet Encyclopedia. However, at the same time it was considered as a "notorious open problem" [18], [6] and "unfeasible" [13]. For some positive results in this area see [4], [5], [11], [12], [13], [15], [16], [20], [21].

Avraham N. Trahtman: The Road Coloring and Černy Conjecture, pp. 1–12.
Proceedings of PSC 2008, Jan Holub and Jan Žďárek (Eds.), ISBN 978-80-01-04145-1 © Czech Technical University in Prague, Czech Republic

The road coloring conjecture is settled in the affirmative: A finite strong digraph with constant outdegree has a synchronizing coloring if and only if the greatest common divisor of the lengths of its cycles is 1.

The concept of a stable pair of states [6], [16] of Culik, Karhumaki and Kari with corresponding results and consequences is used below. The first version of our paper had also used results from [11]. However, we are now able to simplify the proof using idea from [3], [25] and [26].

A problem of the minimal length of synchronizing word, best known as Černy's conjecture, was raised independently by distinct authors. Jan Černy found in 1964 [7] n-state complete DFA with shortest synchronizing word of length $(n-1)^2$ for alphabet size $q = 2$. He conjectured that it is an upper bound for the length of the shortest synchronizing word for any n-state complete DFA. The best known upper bound is now equal to $(n^3 - n)/6$ [10], [17]. The conjecture holds true for a lot of automata, but in general the problem still remains open. Moreover, the examples of automata with shortest synchronizing word of length $(n-1)^2$ are infrequent. After the sequence found by Černy and example of Černy, Piricka and Rosenauerova [8] of 1971 for $q = 2$, the next such example was found by Kari [16] only in 2001 for $n = 6$ and $q = 2$. Roman [23] had found an analogous example for $n = 5$ and $q = 3$ in 2004. There are no examples of automata for the time being such that the length of the shortest synchronizing word is greater than $(n-1)^2$.

We use a new efficient algorithm for finding a synchronizing word. The known algorithm of Eppstein [9] finds a synchronizing word for n-state DFA in $O(n^3 + n^2 q)$ time. The actual running time of our algorithm $(O(n^2 q))$ on a lot of examples proved to be less than in the case of $O(n^3 q)$ time complexity (the worst case). It gives a chance to extend noticeably the class of considered DFA.

The program had studied all automata with strongly connected transition graph of size $n \leq 10$ for $q = 2$, of size $n \leq 8$ for $q \leq 3$ and of size $n \leq 7$ for $q \leq 4$. All known together with some new examples of DFA with shortest synchronizing word of length $(n-1)^2$ from this class of automata were obtained. So all examples of DFA with shortest synchronizing word of length $(n-1)^2$ in this area are known for today. The size of the alphabet of the examples is two or three. The contradictory examples for the Černy conjecture do not exist in this class of automata. Moreover, the program does not find examples of DFA with reset word of length $(n-1)^2$ for $n > 4$ as well as for $q > 3$. No such examples exist also for alphabet of size four if $n \leq 7$ and of size three if $n \leq 8$.

All examples on the Černy border $(n-1)^2$ except one have loops and therefore by some recoloring have shortest synchronizing word of length not greater than $n-1$. It supports the conjecture that by some coloring every synchronizing automaton has synchronizing word of length less than $(n-1)^2$.

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.

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

If there exists a path in an automaton from the state $\mathbf{p}$ to the state $\mathbf{q}$ and the edges of the path are consecutively labelled by $\sigma_1, ..., \sigma_k$, then for $s = \sigma_1 ... \sigma_k \in \Sigma^+$ let us write $\mathbf{q} = \mathbf{p}s$ and $\mathbf{p} \succeq \mathbf{r}$.

Let Ps be the map of the subset P of states of an automaton by help of $s \in \Sigma^+$ and let Ps^{-1} be the maximal set of states Q such that $Qs \subseteq P$. For the transition graph Γ of an automaton let Γs denote the map of the set of states of the automaton.

$|P|$ – the size of the subset P of states from an automaton (of vertices from a graph).

A word $s \in \Sigma^+$ is called a *synchronizing* (or*2-reset*) word of the automaton with transition graph Γ if $|\Gamma s| = 1$.

A coloring of a directed finite graph is *synchronizing* if the coloring turns the graph into a deterministic finite automaton possessing a synchronizing word.

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

A synchronizing pair of states $\mathbf{p}$, $\mathbf{q}$ of an automaton is called *stable* if for any word u the pair $\mathbf{p}u, \mathbf{q}u$ is also synchronizing [6], [16].

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 Γ) of maximal size such that every pair of states from the set is deadlock will be called an *F-clique*.

A state [a vertex] $\mathbf{r}$ is called *sink* of an automaton [of a graph] if $\mathbf{p} \succeq \mathbf{r}$ for all states $\mathbf{p}$.

The direct product Γ^2 of two copies of the graph Γ over the alphabet Σ consists of vertices $(\mathbf{p}, \mathbf{r})$ and edges $(\mathbf{p}, \mathbf{r}) \to (\mathbf{p}\sigma, \mathbf{r}\sigma)$ labelled by σ. Here $\mathbf{p}, \mathbf{r} \in \Gamma$, $\sigma \in \Sigma$.

1 Some properties of F-clique

The road coloring problem was formulated for *AGW* graphs [1] and only such graphs are considered below. We exclude from the consideration also the primitive cases of graphs with loops and of only one color [1], [20].

Let us recall that a binary relation ρ on the set of the states of an automaton is called congruence if ρ is equivalence and for any word u from $\mathbf{p}\ \rho\ \mathbf{q}$ follows $\mathbf{p}u\ \rho\ \mathbf{q}u$. Let us formulate an important result from [16] in the following form:

Theorem 1. *[16] Let us consider a coloring of AGW graph Γ. Stability of states is a binary relation on the set of states of the obtained automaton; denote this relation by ρ. Then ρ is a congruence relation, Γ/ρ presents an AGW graph and synchronizing coloring of Γ/ρ implies synchronizing recoloring of Γ.*

Lemma 2. *Let F be F-clique via some coloring of AGW graph Γ. For any word s the set Fs is also an F-clique and any state [vertex] $\mathbf{p}$ belongs to some F-clique.*

Proof. Any pair $\mathbf{p}$, $\mathbf{q}$ from an F-clique F is a deadlock. To be deadlock is a stable binary relation, therefore for any word s the pair $\mathbf{p}s$, $\mathbf{q}s$ from Fs also is a deadlock. So all pairs from Fs are deadlocks.

For any $\mathbf{r}$ from a strongly connected graph Γ, there exists a word u such that $\mathbf{r} = \mathbf{p}u$ for $\mathbf{p}$ from the F-clique F, whence $\mathbf{r}$ belongs to the F-clique Fu.

Lemma 3. *Let A and B ($|A| > 1$) be distinct F-cliques via some coloring without stable pairs of the AGW graph Γ. Then $|A| - |A \cap B| = |B| - |A \cap B| > 1$.*

Proof. Let us assume the contrary: $|A| - |A \cap B| = 1$. By the definition of F-clique, $|A| = |B|$ and $|B| - |A \cap B| = 1$, too.

The pair of states $\mathbf{p} \in A \setminus B$ and $\mathbf{q} \in B \setminus A$ is not stable. Therefore for some word s the pair $(\mathbf{p}s, \mathbf{q}s)$ is a deadlock. Any pair of states from the F-clique A and from the F-clique B as well as from 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 $|(A \cup B)s| = |A| + 1 > |A|$ in spite of maximality of the size of F-clique A among the sets of states such that every pair of its states is deadlock.

Lemma 4. *Let some vertex of AGW graph Γ have two incoming bunches. Then any coloring of Γ has a stable couple.*

Proof. If a vertex $\mathbf{p}$ has two incoming bunches from vertices $\mathbf{q}$ and $\mathbf{r}$, then the couple $\mathbf{q}$, $\mathbf{r}$ is stable for any coloring because $\mathbf{q}\alpha = \mathbf{r}\alpha = \mathbf{p}$ for any letter (color) $\alpha \in \Sigma$.

2 The spanning subgraph of cycles and trees with maximal number of edges in the cycles

Définition 1 *Let us call a subgraph S of the AGW graph Γ a* spanning subgraph *of Γ if to S belong all vertices of Γ and exactly one outgoing edge of every vertex.*

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

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

Remark 5. 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 vertex from cycle is zero, the vertices of trees except root have positive level, the vertex of maximal positive level has no incoming edge from S.

Lemma 6. *Let L be a set of vertices of level l from some tree of the spanning subgraph S of AGW graph Γ and let all edges of S have a color α by some coloring of Γ. Then for any F-clique F of the coloring holds $|F \cap L| \leq 1$.*

Proof. Some power of α synchronizes all states of given level of the tree and maps them into the root. Any couple of states from an F-clique could not be synchronized and therefore could not belong to L.

Lemma 7. *Let AGW graph Γ have a spanning subgraph R of only disjoint cycles (without trees). Then Γ also has another spanning subgraph with exactly one vertex of maximal positive level.*

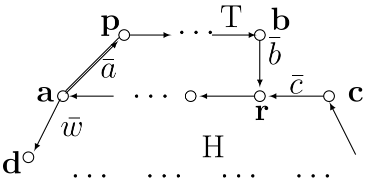
Proof. The spanning subgraph R has only cycles and therefore the levels of all vertices are equal to zero. In view of gcd =1 in the strongly connected graph Γ, not all edges belong to a bunch. Therefore there exist two edges $u = \mathbf{p} \rightarrow \mathbf{q} \notin R$ and $v = \mathbf{p} \rightarrow \mathbf{s} \in R$ with common first vertex $\mathbf{p}$ but such that $\mathbf{q} \neq \mathbf{s}$. Let us replace the edge $v = \mathbf{p} \rightarrow \mathbf{s}$ from R by u. Then only the vertex $\mathbf{s}$ has maximal level $L > 0$ in the new spanning subgraph.

Lemma 8. *Let any vertex of an AGW graph Γ have no two incoming bunches. Then Γ has a spanning subgraph such that all its vertices of maximal positive level belong to one non-trivial tree.*

Proof. Let us consider a spanning subgraph R with a maximal number of vertices [edges] in its cycles. In view of Lemma 7, suppose that R has non-trivial trees and let $L > 0$ be the maximal value of the level of a vertex.

Further consideration is necessary only if at least two vertices of level L belong to distinct trees of R with distinct roots.

Let us consider a tree T from R with vertex $\mathbf{p}$ of maximal level L and edge $\bar{b}$ from vertex $\mathbf{b}$ to the tree root $\mathbf{r} \in T$ on the path of length L from $\mathbf{p}$. Let the root $\mathbf{r}$ belong to the cycle H of R with the edge $\bar{c} = \mathbf{c} \to \mathbf{r} \in H$. There exists also an edge $\bar{a} = \mathbf{a} \to \mathbf{p}$ that does not belong to R because Γ is strongly connected and $\mathbf{p}$ has no incoming edge from R.



Let us consider the path from $\mathbf{p}$ to $\mathbf{r}$ of maximal length L in T. Our aim is to extend the maximal level of the vertex on the extension of the tree T much more than the maximal level of vertex of other trees from R. We plan to use the following three changes:

1) replace the edge $\bar{w}$ from R with first vertex $\mathbf{a}$ by the edge $\bar{a} = \mathbf{a} \to \mathbf{p}$,

2) replace the edge $\bar{b}$ from R by some other outgoing edge of the vertex $\mathbf{b}$,

3) replace the edge $\bar{c}$ from R by some other outgoing edge of the vertex $\mathbf{c}$.

If one of the ways does not succeed let us go to the next assuming the situation in which the previous way fails and excluding the successfully studied cases. So we diminish the considered domain. We can use sometimes two changes together. Let us begin with

1) Suppose first $\mathbf{a} \notin H$. If $\mathbf{a}$ belongs to a path in T from $\mathbf{p}$ to $\mathbf{r}$ then a new cycle with part of the path and edge $\mathbf{a} \to \mathbf{p}$ is added to R extending the number of vertices in its cycles in spite of the choice of R. In opposite case the level of $\mathbf{a}$ in the new spanning subgraph is $L + 1$ and the vertex $\mathbf{r}$ is a root of the new tree containing all vertices of maximal level (in particular, the vertex $\mathbf{a}$ or its ancestors in R).

So let us assume $\mathbf{a} \in H$ and suppose $\bar{w} = \mathbf{a} \to \mathbf{d} \in H$. In this case the vertices $\mathbf{p}$, $\mathbf{r}$ and $\mathbf{a}$ belong to a cycle H_1 with new edge $\bar{a}$ of a new spanning subgraph R_1. So we have the cycle $H_1 \in R_1$ instead of $H \in R$. If the length of path from $\mathbf{r}$ to $\mathbf{a}$ in H is r_1 then H_1 has length $L + r_1 + 1$. A path to $\mathbf{r}$ from the vertex $\mathbf{d}$ of the cycle H remains in R_1. Suppose its length is r_2. So the length of the cycle H is $r_1 + r_2 + 1$. The length of the cycle H_1 is not greater than the length of H because the spanning subgraph R has maximal number of edges in its cycles. So $r_1 + r_2 + 1 \geq L + r_1 + 1$, whence $r_2 \geq L$. If $r_2 > L$, then the length r_2 of the path from $\mathbf{d}$ to $\mathbf{r}$ in a tree of R_1 (and the level of $\mathbf{d}$) is greater than L and the level of $\mathbf{d}$ (or of some other ancestor of $\mathbf{r}$ in a tree from R_1) is the desired unique maximal level.

So assume for further consideration $L = r_2$ and $\mathbf{a} \in H$. Analogously, for any vertex of maximal level L with root in the cycle H and incoming edge from a vertex $\mathbf{a}_1$ the proof can be reduced to the case $\mathbf{a}_1 \in H$ and $L = r_2$ for the corresponding new value of r_2.

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

The vertex $\mathbf{v}$ could not belong to T because in this case a new cycle is added to R and therefore a new spanning subgraph has a number of vertices in the cycles greater than in R.

If the vertex $\mathbf{v}$ belongs to another tree of R but not to 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 $\mathbf{p}$ to the new root is extended. So only the tree T_1 has states of new maximal level.

If $\mathbf{v}$ belongs to some cycle $H_2 \neq H$ from R, then together with replacing $\bar{b}$ by $\bar{v}$, we replace also the edge $\bar{w}$ by $\bar{a}$. So we extend the path from $\mathbf{p}$ to the new root $\mathbf{v}$ at least by the edge $\bar{a} = \mathbf{a} \to \mathbf{p}$ and by almost all edges of H. Therefore the new maximal level $L_1 > L$ has either the vertex $\mathbf{d}$ or its ancestors from the old spanning subgraph R.

Now there remains only the case when $\mathbf{v}$ belongs to the cycle H. The vertex $\mathbf{p}$ also has level L in new tree T_1 with root $\mathbf{v}$. The only difference between T and T_1 (just as between R and R_1) is the root and the incoming edge of the root. The new spanning subgraph R_1 has also a maximal number of vertices in cycles just as R. Let r_3 be the length of the path from $\mathbf{d}$ to the new root $\mathbf{v} \in H$.

For the spanning subgraph R_1, one can obtain $L = r_3$ just as it was done on the step 1) for R. From $\mathbf{v} \neq \mathbf{r}$ follows $r_3 \neq r_2$, though $L = r_3$ and $L = r_2$.

So for further consideration suppose that the set of outgoing edges of the vertex $\mathbf{b}$ is a bunch to $\mathbf{r}$.

3) The set of outgoing edges of the vertex $\mathbf{c}$ is not a bunch to $\mathbf{r}$ because $\mathbf{r}$ has another bunch from $\mathbf{b}$.

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

So remains the case $\mathbf{u} \notin T$. Then the tree T is a part of a new tree with a new root and the path from $\mathbf{p}$ to the new root is extended at least by a part of H from the former root $\mathbf{r}$. The new level of $\mathbf{p}$ therefore is maximal and greater than the level of any vertex in some another tree.

Thus anyway there exists a spanning subgraph with vertices of maximal level in one non-trivial tree.

Theorem 9. *Any AGW graph Γ has a coloring with stable couple.*

Proof. By Lemma 4, in the case of vertex with two incoming bunches Γ has a coloring with stable couples. In opposite case, by Lemma 8, Γ has a spanning subgraph R such that the vertices of maximal positive level L belong to one tree of R.

Let us give to the edges of R the color α and denote by C the set of all vertices from the cycles of R. Then let us color the remaining edges of Γ by other colors arbitrarily.

By Lemma 2, in a strongly connected graph Γ for every word s and F-clique F of size $|F| > 1$, the set Fs also is an F-clique of the same size and for any state $\mathbf{p}$ there exists an F-clique F such that $\mathbf{p} \in F$.

In particular, some F has non-empty intersection with the set N of vertices of maximal level L. The set N belongs to one tree, whence by Lemma 6 this intersection has only one vertex. The word α^{L-1} maps F on an F-clique F_1 of size $|F|$. One has $|F_1 \setminus C| = 1$ because the sequence of edges of color α from any tree of R leads to the root of the tree, the root belongs to a cycle colored by α from C and only for the set

N with vertices of maximal level holds $N\alpha^{L-1} \not\subseteq C$. So $|N\alpha^{L-1} \cap F_1| = |F_1 \setminus C| = 1$ and $|C \cap F_1| = |F_1| - 1$.

Let the integer m be a common multiple of the lengths of all considered cycles from C colored by α. So for any $\mathbf{p}$ from C as well as from $F_1 \cap C$ holds $\mathbf{p}\alpha^m = \mathbf{p}$. Therefore for an F-clique $F_2 = F_1\alpha^m$ holds $F_2 \subseteq C$ and $C \cap F_1 = F_1 \cap F_2$.

Thus two F-cliques F_1 and F_2 of size $|F_1| > 1$ have $|F_1| - 1$ common vertices. So $|F_1 \setminus (F_1 \cap F_2)| = 1$. Consequently, in view of Lemma 3, there exists a stable couple in the considered coloring.

Theorem 10. *Every AGW graph Γ has synchronizing coloring.*

The proof follows from Theorems 9 and 1.

3 Some auxiliary properties

Lemma 11. *Suppose $\mathbf{p} \notin \Gamma s$. Then $\mathbf{p} \notin \Gamma us$ for any word u.*

Proof follows from $\Gamma u \subseteq \Gamma$.

Lemma 12. *Suppose $\mathbf{p} \notin \Gamma s$ for a word s and a state $\mathbf{p}$ of transition graph Γ of DFA.*

Then there exist two minimal integer k and r such that $\mathbf{p}s^k = \mathbf{p}s^{k+r}$. The pair of states $\mathbf{p}, \mathbf{p}s^r$ has 2-reset word s^k and for every $i < k$ the pair of states $\mathbf{p}s^i, \mathbf{p}s^{r+i}$ has 2-reset word s^{k-i}. The word s^k is a 2-reset word for at least k different pairs of states.

In the case $r = 1$, the word s^k maps the set of states $\mathbf{p}, \mathbf{p}s, ..., \mathbf{p}s^k$ on $\mathbf{p}s^k$.

Proof. The sequence $\mathbf{p}s, \mathbf{p}s^2, ..., \mathbf{p}s^t, ...$ is finite and belongs to Γs. Therefore such k and r exist. Two states $\mathbf{p}s^i$ and $\mathbf{p}s^{r+i}$ are mapped by the power s^{k-i} on $\mathbf{p}s^k = \mathbf{p}s^{k+r}$ as well as the states $\mathbf{p}$ and $\mathbf{p}s^r$ are mapped by the power s^k on $\mathbf{p}s^k$. All states $\mathbf{p}s^i$ are distinct for $i \leq k$, whence the word s^k unites at least k distinct pairs of states.

In the case $r = 1$, $\mathbf{p}s^k = \mathbf{p}s^j s^k$ for any j. All states $\mathbf{p}s^i$ are distinct for $0 \geq i \leq k$, whence the word s^k unites in this case at least $k + 1$ distinct states.

Lemma 13. *Suppose $\mathbf{r}\alpha = \mathbf{t}\alpha$ for a letter α and two distinct states $\mathbf{r}$, $\mathbf{t}$ of transition graph Γ of DFA and let the states $\mathbf{r}$ and $\mathbf{r}\alpha$ be consecutive states of a cycle C of Γ.*

Then there exists a word s of length of the cycle C such that $\mathbf{r}s = \mathbf{r}$ and $|\Gamma s| < |\Gamma|$. For some state $\mathbf{p} \in \Gamma \setminus \Gamma s$ there exists a minimal integer k such that $\mathbf{p}s^k = \mathbf{p}s^{k+1}$. The pair of states $\mathbf{p}, \mathbf{p}s^k$ has 2-reset word s^k and for every $i < k$ the pair of states $\mathbf{p}s^i, \mathbf{p}s^k$ has 2-reset word s^{k-i}. The word s^k unites at least $k + 1$ distinct states.

Proof. A word s with first letter α can be obtained from consecutive letters on the edges of the cycle C. Therefore $|s|$ is equal to the length of the cycle and $\mathbf{r}s = \mathbf{r}$. $|\Gamma s| < |\Gamma|$ follows from $\mathbf{r}\alpha = \mathbf{t}\alpha$.

From $\mathbf{r}s = \mathbf{r} \neq \mathbf{t}$ and $\mathbf{r}\alpha = \mathbf{t}\alpha$ follows that $\mathbf{t}s = \mathbf{r} \neq \mathbf{t}$, whence $\mathbf{r} = \mathbf{t}s^i \neq \mathbf{t}$ for any integer i. In the case $\mathbf{t} \in \Gamma \setminus \Gamma s$ suppose $\mathbf{p} = \mathbf{t}$, and so the state $\mathbf{p}$ is defined.

In opposite case the state $\mathbf{t}$ has by mapping s some preimage $\mathbf{t}s^{-1}$ and in view of $\mathbf{t}s^i \neq \mathbf{t}$ for all i there exists an integer k (only one) such that the state $\mathbf{t}s^{-k}$ belongs to $\Gamma \setminus \Gamma s$. Now suppose $\mathbf{p} = \mathbf{t}s^{-k}$. One has $\mathbf{p}s^k = \mathbf{p}s^{k+1} = \mathbf{r}$ for $\mathbf{p}$ from $\Gamma \setminus \Gamma s$.

So the pair of states $\mathbf{p}, \mathbf{p}s^k$ has 2-reset word s^k and for every $i < k$ the pair of states $\mathbf{p}s^i, \mathbf{p}s^k$ has 2-reset word s^{k-i}. The states $\mathbf{p}s^i$ for $i \leq k$ and $\mathbf{p}$ are distinct because k is unique. The word s^k maps all these states on the state $\mathbf{r}$.

Lemma 14. *Let Γ be strongly connected graph of synchronizing automaton with transition semigroup S. Suppose $\Gamma a = \Gamma b$ for reset words a and b. Then $a = b$. Any reset word is an idempotent.*

Proof. The elements a and b from S induce equal mappings on the set of states of Γ. S can be embedded into the semigroup of all functions on the set of states under composition. Therefore $a = b$ in S. $\Gamma a = \Gamma a^2$, whence $a = a^2$ for any reset word a and the element $a \in S$ is an idempotent.

4 Synchronizing Algorithms

The following help construction was supposed by Eppstein [9]. Let us keep for any pair of states $\mathbf{r}, \mathbf{p}$ the first letter α of the minimal 2-reset word w of the pair together with the length of the word w. The second letter of w is the first letter of the analogical word of the pair of states $\mathbf{r}\alpha, \mathbf{p}\alpha$. Therefore the 2-reset word w of minimal length can be restored on this way. The time and space complexity of this preprocessing is $O(n^2)$ [9] for n-state automaton.

4.1 Checking synchronizability

A help algorithm with $O(n^2q)$ time complexity in the worst case verifies whether or not a given n-state DFA of alphabet size q is synchronizing. The algorithm follows [9]. Our modification of the algorithm finds first all SCC of the graph (the first-depth search is a linear) and then checks the minimal SCC Γ_s of sink states of the graph (if exists). If there is no sink state then the automaton is not synchronizing. Exactly one sink state implies synchronizability. The time and space complexity of the algorithm in both these cases are linear.

Let us consider the graph Γ_s with at least two sink states. The next step is the consideration of Γ_s^2. We unite any pair of states $(\mathbf{p}, \mathbf{r})$ and $(\mathbf{r}, \mathbf{p})$, all states $(\mathbf{r}, \mathbf{r})$ are united in one state $(0, 0)$. Then let us mark sink state $(0, 0)$ and all ancestors of $(0, 0)$ using the first-depth search on the reverse of the obtained graph G. The graph Γ is synchronizing if any node of G will be marked.

4.2 An efficient algorithm for reset word

An efficient semigroup algorithm, essential improvement of the algorithm [9], based on the properties of transition semigroup and inspired mostly by results from the previous section plays a central role in the program.

We consider the square Γ^2 and the reverse graph I of Γ. The graph I is not deterministic for synchronizing graph Γ. Suppose that the graph Γ is synchronizing, all sink states are found on the stage of checking of the synchronizability, the graph Γ^2 and the reverse graph I were build.

Let us find by help of the reverse graph I for any pair of states $\mathbf{r}, \mathbf{p}$ from Γ^2 the first letter of the minimal 2-reset word w of the pair and the length of w [9]. So for any pair $\mathbf{r}, \mathbf{p}$ can be restored a 2-reset word w of minimal length.

Let us order the set of states $(\mathbf{r}, \mathbf{p})$ according to the length of the word w. The ordering can be made linear in the size of the set in the following way:

Let us find first the number c_i of all states $(\mathbf{r}, \mathbf{p})$ with given length i of minimal 2-reset word for any i, then adjust the intervals of size c_i for to place the pairs and then allocate in every interval the pairs with common length. It needs $O(n^2)$ time.

We use also a complementary idea for to reorder the pairs of states. If a word w unites at least two states let us find the number of states united by powers of w and use this value for *complementary* order.

The important part of the preprocessing supposed by Eppstein was the computing of the mapping Γw of the graph Γ induced by the minimal 2-reset word w of the pair of states $\mathbf{r}, \mathbf{p}$. This stage begins from the shortest words w and therefore is linear for any considered pair of states $\mathbf{r}, \mathbf{p}$. Nevertheless, the time complexity of the stage is $O(n^3)$. For to avoid the extremes of this step, our algorithm stops on linear number of pairs. The obtained set G of 2-reset words is considered as a set of generators of some subsemigroup from A and will be marked together with corresponding pairs of states. The time complexity of this step is therefore $O(n^2)$. Let us reorder G in the *complementary* order and use the mapping of the graph induced by powers of generators.

Let Γ_i be consecutive images of the graph $\Gamma = \Gamma_0$ such that for $w_i \in A$ holds $\Gamma_i w_{i+1} = \Gamma_{i+1}$ and $|\Gamma_i| > |\Gamma_{i+1}|$. Let A_i be a semigroup generated by the set w_1, ... w_i. Let us check pairs of states corresponding to the words from G. If the pair belongs to Γ_i then the corresponding minimal reset word w_{i+1} together with its powers may be used for to find the image Γ_{i+1}.

In the case no minimal 2-reset word of a pair from Γ_i was marked, let us consider the products of marked words. If some product unites a pairs of states of Γ_i, then let us use the mapping, mark the product of words and the pair of states. Let us notice that on this step are considered not all marked pairs. The number of considered products must be linear in the size of Γ. The product of two mappings can be found in linear time. Therefore the time complexity of this stage is $O(nk)$ for the defect k of the mapping of Γ_i.

If two considered stages still do not find a reset word, then the new generator must be added to considered subsemigroup A_i. Let us take a pair of states $\mathbf{r}, \mathbf{p}$ from Γ_i with reset word w_i. Suppose $w_i = u_i v_i$ such that the word v_i was marked. Then the mapping w_i can be found in $n|u_i|$ time. Let us notice that only on this step the time complexity may by greater than quadratic.

Lemma 15. *Let Γ_i be consecutive images of the graph $\Gamma = \Gamma_0$ such that for v_i from semigroup A $\Gamma_i v_{i+1} = \Gamma_{i+1}$, $|\Gamma_i| > |\Gamma_{i+1}|$ and $|\Gamma_s| = 1$ for some integer s. Let A_i be a semigroup generated by the set w_1, ... w_i such that $w_i = u_i v_i$ is a reset word for some pair of states from Γ_{i-1} and v_i is a marked element of the subsemigroup A_{i-1}.*

Then the considered algorithm has $max(O(|\Gamma|^2 q), O(|\Gamma||u_1...u_s|)$ time complexity.

Proof. The time complexity of the step of the building of Γ^2 is $O(|\Gamma|^2 q)$. So $O(|\Gamma|^2 q)$ is a lower bound for the complexity of the considered algorithm.

Let the set w_1, ... w_i generate A_i. The creation of the mapping w_i needs $|\Gamma||u_i| + 1$ steps because for the marked element v_i the mapping is known.

The element will be marked and used only if it is either a generator from A_i or a product of two marked elements. With a marked semigroup element will be associated the mapping of Γ defined by the element. The finding of the mapping of the product of two elements with known images is linear in the size of the graph.

We repeat the process with the obtained image Γ_i. The defect of the mapping is growing on every step. After not over than $|\Gamma| - 1$ steps Γ will be synchronized.

As for complexity of the algorithm, let us notice that the length of the synchronizing word found by the algorithm was less than n^2 in all considered cases. The stage of adding of new generators was used only in a small number of cases, only some percents of considered automata. The number of generators of the semigroup A is usually small. For instance, for Černy graphs there are only two generators. Therefore the time complexity of the algorithm is $O(n^2q)$ in majority of cases and the algorithm can be considered as subquadratic.

4.3 An algorithm for reset word of minimal length

A straightforward algorithm for finding synchronizing word of minimal length is used by the program on its last stage. The algorithm is not polynomial in the most worst case (the finding of the synchronizing word of minimal length is NP-hard [9], [19]). The size of the transition semigroup is in general not polynomial in the size of the transition graph. The program for search of minimal reset word uses this algorithm relatively rare.

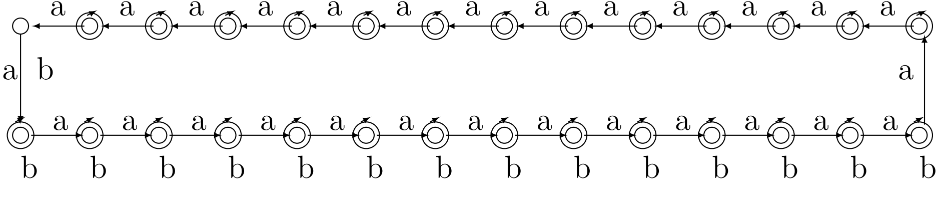
We find mappings of the graph of the automaton induced by the letters of the alphabet of the labels. Mappings with the same set of states are identified. It essentially simplifies the process. Distinct mappings are saved. For this aim, any two mappings must to be compared, so we have $O(s(s-1)/2)$ steps for s mappings.
The mappings correspond to semigroup elements. With any mapping let us connect a previous mapping and the letter that creates the mapping. On this way, the path on the graph of the automaton can be constructed. The time complexity of the considered procedure is $O(nqs^2)$ with $O(ns)$ space complexity.

Proposition 16. *The algorithm finds a list of all words (elements of transition semigroup) of length k where k is growing. The first synchronizing word of the list has minimal length.*

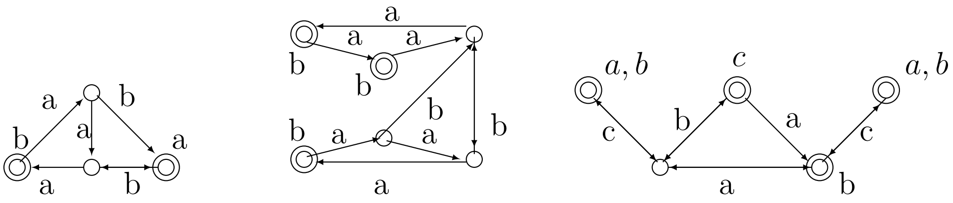
5 Experimental data

The considered synchronization algorithms were used in a program for search of automata with minimal reset word of relatively great length. The program has investigated all complete DFA for $n \leq 10$, $q = 2$ and for $n \leq 7$, $q \leq 4$.
An automaton with k states outside sink SCC A of the transition graph can be mapped on A by word of length not greater than $k(k-1)/2$. Therefore only automata with strongly connected transition graphs need investigation. The graphs with synchronizing proper subgraph obtained by moving off letters from the alphabet are omitted too. In particular, there are no synchronizing 3-state automata for $q \geq 3$ such that by removing any letter the obtained automata are not synchronizing. Therefore such automata are not studied and in the table below for $n = 3$ appears zero.

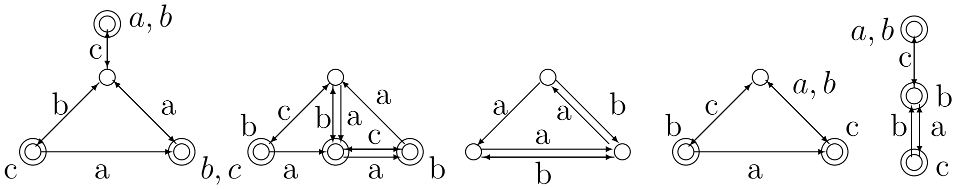
The known n-state automata with minimal reset word of length $(n-1)^2$ are presented by sequence of Černy [7] (here n=28):



by automata supposed by Černy, Piricka and Rosenauerova [8], by Kari [16] and Roman [23].



Our program has found five new following examples on the border $(n-1)^2$.



The corresponding reset words of minimal length are: *abcacabca*, *acbaaacba*, *baab*, *acba*, *bacb*. All considered algorithms have found the same reset word for every example. The size of the transition semigroup found by the package TESTAS is 148, 180, 24, 27 and 27 correspondingly.

There are no contradictory examples for the Černy conjecture in considered class of automata. Moreover, there are no new examples of automata with reset word of length $(n-1)^2$ for $n > 4$ and $q > 3$ in this class. And what is more, the examples with minimal length of reset word disappear even for values near the Černy bound $(n-1)^2$ with growth of the size of the automaton and of the size of the alphabet. The following table displays this noteworthy trend for the maximum of lengths of minimal reset words of length less than $(n-1)^2$. By $*$ are denoted here non-isomorphic automata having minimal reset words of length $(n-1)^2$ that do not belong to Černy sequence.

size of the automaton	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10
$(n-1)^2$	4	9	16	25	36	49	64	81
max length, 2 letters	$3*$	$8*$	15	$23*$	32	44	58	74
max length, 3 letters	$0**$	$8**$	$15*$	23	31	≤ 44	–	–
max length, 4 letters	0	8	15	22	30	–	–	–

The gap between $(n-1)^2$ and the maximum of considered length of the minimal reset word grows with n and q. This gap supports the following funny

Conjecture *The set of n-state DFA with minimal reset word of length not less than* $(n-1)^2$ *contains only the sequence of Černy and the eight automata mentioned above, three of size 3, three of size 4, one of size 5 and one of size 6.*

and also

Conjecture *Any AGW graph has coloring with minimal reset word of length less than* $(n-1)^2$.

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. M.P. B'eal, D. Perrin: *A quadratic algorithm for road coloring.* arXiv:0803.0726v2 [cs.DM].
4. G. Budzban, A. Mukherjea: *A semigroup approach to the Road Coloring Problem*, Probability on Algebraic Structures. Contemporary Mathematics, 261(2000), 195–207.

5. A. Carbone: *Cycles of relatively prime length and the road coloring problem*, Israel J. of Math., 123(2001), 303–316.
6. 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.
7. J.Černy: Poznamka k homogenym eksperimentom s konechnymi automatami, Math.-Fyz. Čas., 14(1964) 208–215.
8. J. Černy, A. Piricka, B. Rosenauerova: On directable automata, Kybernetika 7(1971), 289–298.
9. D. Eppstein: Reset sequences for monotonic automata. SIAM J. Comput., 19(1990), 500–510.
10. P. Frankl: An extremal problem for two families of sets, Eur. J. Comb., 3(1982), 125–127.
11. J. Friedman: *On the road coloring problem*, Proc. of the Amer. Math. Soc. 110(1990), 1133–1135.
12. E. Gocka, W. Kirchherr, E. Schmeichel: *A note on the road-coloring conjecture.* Ars Combin. 49(1998), 265–270.
13. R. Hegde, K. Jain: *Min-Max theorem about the Road Coloring Conjecture* EuroComb 2005, DMTCS proc., AE, 2005, 279–284.
14. P.M. Higgins: The range order of a product of I-transformation from a finite full transformation semigroup, Semigroup Forum, 37(1988), 31–36.
15. N. Jonoska, S. Suen: *Monocyclic decomposition of graphs and the road coloring problem*, Congressum numerantium, 110(1995), 201–209.
16. J. Kari: *Synchronizing finite automata on Eulerian digraphs*, Springer, Lect. Notes in Comp. Sci., 2136(2001), 432–438.
17. A.A. Kljachko, I.K. Rystsov, M.A. Spivak: An extremely combinatorial problem connected with the bound on the length of a recurrent word in an automata. Kybernetika. 2(1987), 16–25.
18. D. Lind, B. Marcus: *An Introduction of Symbolic Dynamics and Coding*, Cambridge Univ. Press, 1995.
19. A. Mateescu, A. Salomaa: *Many-Valued Truth Functions, Černy's Conjecture and Road Coloring*, Bull. of Eur. Ass. for TCS, 68(1999), 134–148.
20. G.L. O'Brien: *The road coloring problem*, Isr. J. of Math., 39(1981), 145–154.
21. D. Perrin, M.P. Schützenberger: *Synchronizing prefix codes and automata, and the road coloring problem*, In Symbolic Dynamics and Appl., Contemp. Math., 135(1992), 295–318.
22. J.E. Pin: *On two combinatorial problems arising from automata theory*, Annals of Discrete Math., 17(1983), 535–548.
23. A. Roman: *Synchronization of finite automaton. Computations for different alphabet sizes*, Workshop on words and automata. S-Petersburg. 2006.
24. A.N. Trahtman: *Notable trends concerning the synchronization of graphs and automata*, CTW06, El. Notes in Discrete Math., 25(2006), 173-175.
25. M.V. Volkov: A private letter.
26. W.H. Wheeler: A note on Trakhtman's proof of the road coloring theorem. Submitted.