Linear Visualization of a Road Coloring Algorithm

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

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

Abstract

The visualization has become essential in many application areas. The finite graphs and automata undoubtedly belong to such areas. A problem of a visual image of a directed finite graph has appeared in the study of the *road coloring conjecture*.

Given a finite directed graph, a coloring of its edges turns the graph into finite-state automaton. The visual perception of the structure properties of automata is an important goal. 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 conjecture [1], [2] was stated about forty yeas ago 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 that turns the graph into a deterministic finite automaton possessing a synchronizing word. Such graph has according to the conjecture a synchronizing coloring.

The problem belonged to the most fascinating problems in the theory of finite automata [9], [4] and was mentioned in the popular Internet Encyclopedia "Wikipedia" on the list of the most interesting unsolved problems in mathematics. The positive solution of the road coloring problem [13] is a basis of a polynomial-time implemented algorithm of $O(n^3)$ complexity in the worst case.

The realization of the considered algorithm is demonstrated by a high-speed visualization program. The visibility of inner structure of a digraph without doubt is a matter of interest not only for road coloring, the range of the application may be significantly wider.

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 [11], [14] together with some new productive ideas. Our algorithm for the visualization is linear in the size of the automaton. This algorithm not complicated at first sight successfully solves a whole series of tasks of the disposal of the objects.

The visualization of the transition graph of the automaton is a help tool of the

study of the automata. Thus the linearity of the algorithm is comfortably and important. Both the road coloring algorithm and the visualization algorithm are implemented in the package TESTAS (www.cs.biu.ac.il/∼trakht/syn.html).

As usual, we regard a directed graph with colors assigned to its edges as a finite automaton, whose input alphabet consists of these colors. The graph is called *transition graph* of the automaton.

An automaton is *deterministic* if no state has two outgoing edges of the same color. In *complete* automaton each state has outgoing edges of any color.

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

Let Ps be the set of states $\mathbf{p}s$ for $\mathbf{p} \in P$ $s \in \Sigma^+$. For the transition graph Γ of an automaton let Γs denote the map of the set of states of the automaton.

A word $s \in \Sigma^+$ is called a *k-synchronizing* word of the automaton with transition graph Γ if both $|\Gamma s| = k$ and for all words $t \in \Sigma^*$ holds $|\Gamma t| \geq k$.

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 $s \in \Sigma^+$.

A synchronizing pair of states $\mathbf{p}$, $\mathbf{q}$ of an automaton is called *stable* if for every word u the pair $\mathbf{p}u, \mathbf{q}u$ is also synchronizing [4], [**?**].

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

Imagine a map with roads which are colored in such a way that a fixed sequence of colors, called a synchronizing sequence, leads to a fixed place whatever is the starting point. Finding such a coloring is called em road coloring problem. The roads of the map are considered as edges of a directed graph. The visual presentation of a road coloring algorithm is essentially based on the paths of the graph. The paths must be visible as well as cycles, bunches and other structure components of the graph. In particular, the notion of the bunch according to the following lemma plays some role in the road coloring algorithm.

Lemma 1 *[13] If some vertex of graph Γ has two incoming bunches then there exists a stable pair by any coloring.*

The role of the length of a path is also important.

Lemma 2 *[13] Let any vertex of the graph Γ have no two incoming bunches. Then a subgraph of Γ of some color has maximal subtree.*

A crucial role in the visualization plays in our opinion 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 automatically drawn graphical image must resemble the last one of a human being. The considered visualization is a help tool for any program dealing with transition graph of DFA and in particular for the road coloring algorithm.

Our main objective is the visual representation of the transition graph of a deterministic finite automaton based on the structure properties of the graph. Any deterministic finite automaton is accepted by the algorithm.

Among the important visual properties of a graph one can mention paths, cycles, strongly connected components, cliques, bunches etc. These important properties reflect the inner structure of the digraph. The special significance plays here the strongly connected components (SCC). Thus our first step is the eduction and selection of the SCC. We choose to place SCC according to a cyclic layout[11], [14]. According to approach the vertices are placed at the periphery of a circle. Our modification of the approach considered two levels of circles, the first level consists of strongly connected components, the second level corresponds to the whole graph with SCC at the periphery of the circle. The visual placement is based on the structure of the graph considered as a union of the set of strongly connected components.

It is clear that the curve edges (used, for instance, in the package GraphViz [6], [10]) hinder to recognize the cycles and paths. Therefore, we use only direct and, hopefully, short edges. We have changed some priorities of the layout and, in particular, eliminate the goal of reducing the number of intersections of the edges as it was an important aim in some algorithms [10]. The intersections of the edges are even not considered in our algorithm. This approach gives us an opportunity to simplify essentially the algorithm and to reduce its complexity. Our main intent is only not to stir by the intersections of the edges to conceive the structure of the graph. The intersections are placed in our algorithm far from the vertices due to the cyclic layout [14], [11] we use. The area of vertices differs of the area of the majority of intersections.

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 not clear. Our solution is to use colors on the edges instead of labels and exclude the placing of labels.

The quick linear algorithm for finding SCC [3] is implemented in the program. The vertices of every SCC belong to a cycle in the graph layout. So strongly connected components can be easily recognized by observer. All SCC are placed on the periphery of a big circle. So the pictorial diagram demonstrates the structure of the graph and the visualization can be considered as a kind of structure visualization.

The periphery of a circle of SCC is the most desirable area for placing the edges because the edges in this case are short. We choose the order of the vertices of the SCC on the circle according to this purpose. The length of some edges can be reduced in a such way. It also helps an observer to recognize paths and cycles on the screen.

The linearity of the algorithm ensures the momentary appearance of the layout. It is favorably also for educational purposes because the road coloring conjecture can be stated in simple terms and initial explorations can be done immediately. It can be understood by any student with a little experience in the graph theory. "The Road Coloring Conjecture makes a nice supplement to any discrete mathematics course" [7].

The complexity of the algorithm describes the following

Lemma 3 *The time and space complexity of the visualization algorithm described above is linear in the sum of states and edges of the transition graph of automaton.*

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 Analisys of Computer Algorithms*, 1974, Addison-Wesley.

[4] K. Culik II, J. Karhumaki, J. Kari. A note on synchronized automata and Road Coloring Problem, Lect. Notes in Comput. Sci., 2295, 2002, 175-185.

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

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

[7] J. V. Rauff, Way back from anywhere: exploring the road coloring conjecture. Math. and Comput. Education. 01, 2009.

[8] A. Roman. Synchronizing finite automata with short reset words. Applied Math. and Comput. 1, 209, 2009, 125-136.

[9] A. Mateescu, A.Salomaa, Many-valued truth function. Černy conjuncture and road coloring, EATCS Bulletin, 68(1999) 134-150.

[10] M.Simonato, 2004, *An Introduction to GraphViz*.

[11] JM. Six, IG. Tollis, A framework for user-grouped circular drawings Lect. Notes in Comp. Sci., 1731, 1999, 107-116.

[12] A.N. Trahtman, Notable trends concerning the synchronization of graphs and automata. El. Notes in Discr. Math., 25, 2006, 173-175

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

[14] R. Wiese, M. Eiglsperger, M. Kaufmann. 2002, A framework for circular drawings of network