ON THE SYNCHRONIZATION OF CIRCULAR SEMI-FLOWER AUTOMATA

SHUBH N. SINGH AND ANKIT RAJ

ABSTRACT. Pin proved that every circular automaton with a prime number of states containing a non-permutation is synchronizing. In this paper, we investigate the synchronization of circular semi-flower automata. We first prove that every semi-flower automaton is a one-cluster automaton. Subsequently, we prove that every semi-flower automaton containing a 1-cycle is synchronizing. Further, we prove that every circular semi-flower automaton with an odd number of states containing a 2-cycle is synchronizing.

INTRODUCTION

An automaton is called synchronizing if there exists a word, called a synchronizing word, that sends all its states to a single state. The concept of synchronization of automata has many practical applications in such areas as robotics, manufacturing, coding theory, bio-computing, model-based testing and many others [20].

A lot of investigations have already been done in the area of synchronization of automata. Pin [9] proved that every circular automaton with a prime number of states containing a non-permutation letter is synchronizing. Perles et al. [15] observed that the class of definite automata is a sub-class of synchronizing automata. It is verified that every strongly connected aperiodic automaton is synchronizing. For a given automaton, it is often very challenging to prove that the automaton is synchronizing or not.

Besides of the applications of synchronizing automata, there is a famous conjecture, known as Černý conjecture, related to synchronizing automata. Černý conjecture states that every \(n\)-state synchronizing automaton has a synchronizing word of length at most \((n-1)^2\) [8]. In this connection, Pin [9] proved that the Černý conjecture is satisfied for circular automata with a prime number of states. Further, Dubuc [12] showed that all circular automata satisfies the Černý conjecture. Steinberg [2] proved the Černý conjecture for one-cluster automata with prime length cycle. For another special classes of synchronizing automata, the Černý conjecture has also been verified, or sharper bounds than the general bounds have been proven, see for instance [1, 3, 4, 10, 11, 17, 18, 21].

This paper investigates the synchronization of circular semi-flower automata. Circular automata have been studied in various contexts [9, 12]. Semi-flower automata have been introduced to study the finitely generated submonoids of a free monoid [13, 22]. Using semi-flower automata, the rank and intersection problem of certain finitely generated submonoids of a free monoid have been investigated.

2010 Mathematics Subject Classification. 20M35, 68Q45, 68Q70, 68R10.

Key words and phrases. circular automaton, semi-flower automaton, one-cluster automaton, synchronizing automaton, Černý conjecture.
Semi-flower automata have also been studied in many different contexts, see for instance [6, 14, 21, 25, 26]. The remaining part of the paper is organized as follows. In Section 1, we introduce the notation and briefly give the required background. In Section 2, we investigate the synchronization of circular semi-flower automata. In Section 3, we conclude the paper and provide some future directions for our work.

1. Preliminaries and Notation

In this section we first briefly introduce the notations used in this paper. Throughout this paper, $n$ is an integer greater than 1. Let $P$ be a non-empty finite set. The number of elements in $P$ is denoted by $|P|$. We write argument of a transformation $\alpha$ of $P$ on its left so that $p \alpha$ is the value of $\alpha$ at the argument $p \in P$. The composition of transformations is designated by concatenation, with the leftmost transformation understood to apply first, so that $p(\alpha \beta) = (p\alpha)\beta$. We denote by $T_n$ the full transformation monoid of a set with $n$ elements.

Let $D$ be a (labeled) digraph. The vertex set of $D$ is denoted by $V(D)$. A path in $D$ is an alternating finite sequence $v_0, e_1, v_1, \ldots, v_k, e_k, v_{k+1}$ of distinct vertices and (labeled) edges such that, for $1 \leq i \leq k$, the tail and the head of edge $e_i$ are vertices $v_{i-1}$ and $v_i$, respectively. A path with at least one edge is called a cycle if its initial vertex and terminal vertex are the same. The length of a path is the number of its edges. A $k$-cycle is a cycle of length $k$. If there is a path from vertex $u$ to vertex $v$, then the distance from $u$ to $v$ is the length of shortest path from $u$ to $v$.

An alphabet is a non-empty finite set. The elements of an alphabet are referred to as letters. We denote an alphabet by the symbol $A$. We use symbols $A^*$ and $\varepsilon$ to denote the set of all words over $A$ and the empty word, respectively. We consider an automaton (over $A$) as a quintuple $A = (Q, A, \delta, q_0, F)$, where $Q$ is the non-empty finite set of states, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and $\delta : Q \times A \rightarrow Q$ is the transition (total) function. An automaton with $n$ states is called an $n$-state automaton. The canonical extension of $\delta$ to the set $Q \times A^*$ is still denoted by $\delta$.

Let $A$ be an $n$-state automaton. Each word $x \in A^*$ has a natural interpretation as a transformation of $T_n$ and we do not distinguish between the word $x$ and its interpretation. A letter $a \in A$ is called permutation if its interpretation is a permutation; otherwise, it will be called non-permutation. $A$ is called circular if there exists a permutation letter which induces a circular permutation on its set of states. The set $M(A) = \{x \in T_n \mid x \in A^*\}$ forms a submonoid of the full transformation monoid $T_n$ called the transition monoid of $A$. The automaton $A$ is called synchronizing if there exists a word $x \in A^*$ such that the image of $Q$ under the transformation $x \in M(A)$ is a singleton.

Let $A$ be an automaton. By denoting states as vertices and transitions as (labeled directed) edges, $A$ can be represented by (labeled) digraph, denoted by $D(A)$, in which initial state and final states shall be distinguished appropriately. For $a \in A$, we define an $a$-edge as an edge labeled by the letter $a$. A path (respectively, cycle) in $A$ is a path (respectively, a cycle) in $D(A)$. A state $q$ is said to be accessible (respectively, co-accessible) if there exists a path from $q_0$ to $q$ (respectively, a path from $q$ to some final state). $A$ is called semi-flower automaton (in short, SFA) if $F = \{q_0\}$, every state is both accessible and co-accessible, and all cycles in $A$ visit
ON THE SYNCHRONIZATION OF CSFA

the unique initial-final state \( q_0 \). For further basic definitions concerning digraphs and automata we refer [7, 19].

Let \( G \) be a finite group with identity element \( e \). We write \(|G|\) to denote the order of \( G \). Let \( g \) be an element of \( G \). The order of the element \( g \) is the smallest positive integer \( t \) such that \( g^t = e \). The cyclic subgroup generated by the element \( g \) is denoted by \( \langle g \rangle \). The group \( G \) is called cyclic if \( G = \langle g \rangle \) for some \( g \in G \). In this case, \( g \) is called a generator of \( G \). It is well known that any two finite cyclic groups of the same order are isomorphic. In a finite group, the order of a group element divides the order of its group. Therefore, if \(|G|\) is odd, the order of each element of \( G \) is also odd. All further unexplained notation and terminology of groups we refer [5].

2. Main Results

In this section we investigate the synchronization of circular semi-flower automata (in short, CSFA). In order to investigate the synchronization of CSFA, we recall the concept of one-cluster automata introduced by Béal and Perrin in [16].

Let \( A \) be an automaton and let \( R \) be the sub-digraph of \( D(A) \) made of the \( b \)-edges. A connected component of \( R \) is called a \( b \)-cluster. Note that each \( b \)-cluster contains a unique cycle, called a \( b \)-cycle, with possible trees attached to \( b \)-cycle at their root. A one-cluster automaton with respect to a letter \( b \) is an automaton which has exactly one \( b \)-cluster.

The following theorem proves that every SFA is a one-cluster automaton with respect to each letter.

**Theorem 2.1.** Every SFA is a one-cluster automaton with respect to each letter.

**Proof.** Let \( A \) be an SFA and let \( b \in A \). Consider the sub-digraph \( R \) of the digraph \( D(A) \) made of the \( b \)-edges. It is sufficient to prove that the underlying graph of \( R \) is connected. Let \( p \) be a state such that \( p \neq q_0 \). We claim that \( p \overset{b^r}{\rightarrow} q_0 \) for some positive integer \( r \). Clearly, the state \( p \) belongs to exactly one \( b \)-cluster. Recall that each \( b \)-cluster contains a unique cycle. Since \( A \) is an SFA, the initial-final state \( q_0 \) belongs to each cycle, and subsequently the state \( q_0 \) belongs to each \( b \)-cluster. Therefore \( p \overset{b^r}{\rightarrow} q_0 \) for some positive integer \( r \), and consequently the underlying graph of \( R \) is connected. This proves our theorem. \( \square \)

In view of Theorem 2.1 we get that every SFA contains a unique \( b \)-cycle, where \( b \in A \).

**Notation 2.2.** We shall denote by \( C \) the \( b \)-cycle in an SFA. Further, the set of states in the \( b \)-cycle \( C \) shall be denoted by \( V(C) \).

Obviously \( q_0 \in V(C) \). We now recall the definition of the level of automata from [16]. Let \( A \) be an automaton and let \( b \in A \). The level of a state \( q \) in a \( b \)-cluster is defined as the distance between \( q \) and the root of the tree containing \( q \). If \( q \) belongs to the cycle, its level is defined as zero. The level of \( A \) is the maximal level of its states.

**Notation 2.3.** We shall denote the level of an automaton by symbol \( l \).

Note that \( Qb^l = V(C) \). If the length of the cycle \( C \) is 1, the following theorem asserts about the synchronization of an SFA.

**Theorem 2.4.** Let \( A \) be an SFA. If \( C \) is a 1-cycle in \( A \), then \( A \) is synchronizing.
Proof. Recall that \( q_0 \in V(C) \). Since \(|V(C)| = 1 \), it follows that \( V(C) = \{q_0\} \).
Hence, \( Qb^l = V(C) = \{q_0\} \). This completes our proof. \( \square \)

We now recall the necessary result from [25].

**Theorem 2.5** ([25]). Let \( A \) be an SFA.

(i) For \( a \in A \), if \( a \) is a permutation, then \( a \) is a circular permutation.

(ii) For \( a, b \in A \), if \( a \) and \( b \) are permutations, then the permutations \( a \) and \( b \) are same.

Unless otherwise stated, in what follows, \( A \) denotes an \( n \)-state CSFA. In view of
Theorem 2.5 there is a unique circular permutation induced by letters. For the rest
of paper, we fix the following regarding \( A \). Assume that the letter \( a \in A \) induces
the circular permutation, and accordingly \( q_0, q_1, \ldots, q_n-1 \) is the cyclic ordering of \( Q \)
with respect to \( a \). We use symbol \( G \) to denote the cyclic subgroup of the transition
monoid \( M(A) \) generated by the permutation \( a \).

For completeness, we state the following simple result from [26].

**Remark 2.6** ([26]). Let \( A \) be an \( n \)-state circular semi-flower automaton. Then

(i) \( |G| = n \).

(ii) \( G \) is the group of units of \( M(A) \).

For an odd integer \( n \), if the length of the cycle \( C \) is 2, the following theorem
proves that an \( n \)-state circular semi-flower automaton \( A \) is synchronizing.

**Theorem 2.7.** For an odd integer \( n \), let \( A \) be an \( n \)-state circular semi-flower
automaton. If \( C \) is a 2-cycle in \( A \), then \( A \) is synchronizing.

Proof. Recall that \( q_0 \in V(C) \). Since \(|V(C)| = 2 \), let \( q_m \) \((1 \leq m \leq n-1) \) be another
state such that \( q_m \in V(C) \), and so \( V(C) = \{q_0, q_m\} \). Then \( Qb^l = V(C) = \{q_0, q_m\} \).
Clearly, \( q_m \cdot b = q_0 \) and \( q_0 \cdot b = q_m \). It follows that

\[
q_0 \cdot b^2 = q_0 \implies q_0 \cdot b^2 = q_0 \implies q_0 \cdot b^{1+2l} = q_m \cdot b^2 = q_m,
\]

and

\[
q_m \cdot b^2 = q_m \implies q_m \cdot b^2 = q_m \implies q_m \cdot b^{1+2l} = q_0 \cdot b^2 = q_0.
\]

Thus, \( b^{1+2l} \) maps \( Q \) into \( \{q_0, q_m\} \) and swaps \( q_0 \) and \( q_m \).

Note that \( q_m \cdot a^{(n-m)} = q_0 \). We now consider the sequence \( \langle w_k \rangle \) of words, where
\( w_k := a^{k(n-m)}b^{1+2l} \). From the suffix word \( b^{1+2l} \), it is clear that \( \{q_0, q_m\} : w_k \subseteq \{q_0, q_m\} \) for each \( k \). If there exists a \( k \) such that \( \{q_0, q_m\} : w_k \) is a singleton, then
the automaton \( A \) is synchronizing.

Otherwise, for \( k = 1 \), we have

\[
q_m \cdot a^{(n-m)}b^{1+2l} = q_0 \cdot b^{1+2l} = q_m, \quad \text{and therefore } q_0 \cdot a^{(n-m)}b^{1+2l} = q_0.
\]

By induction, we get that

\[
q_0 \cdot a^{k(n-m)}b^{1+2l} = \begin{cases} q_0 & \text{if } k \text{ is odd} \\ q_m & \text{if } k \text{ is even}. \end{cases}
\]

Since by assumption \( n \) is an odd number and the letter \( a \) induces a circular permuta-
tion, the order of \( a^{(n-m)} \) is also an odd number, say \( t \). Now for this number
\( t \), we have that \( a^{t(n-m)} \) induces the identity transformation. Hence

\[
q_0 \cdot a^{t(n-m)}b^{1+2l} = q_0 \cdot b^{1+2l} = q_m,
\]

a contradicting to the previous statement. This completes our proof. \( \square \)
The following example shows that Theorem 2.7 is not necessarily true when circular semi-flower automaton has an even number of states.

**Example 2.8.** Consider 6-state circular semi-flower automaton $A_1$ over $A = \{a, b\}$ given in the Figure 1. Clearly, $V(C) = \{q_0, q_3\}$. By using the computer algebra system GAP–Groups, Algorithms and Programming [27] we observe that the transition monoid $M(A_1)$ does not contain a constant transformation. Hence, $A_1$ is non-synchronizing.

![Figure 1. 6-state non-synchronizing CSFA $A_1$ with $|V(C)| = 2$](image)

For an odd integer $n$, let $A$ be an $n$-state circular semi-flower automaton. If $C$ is a 3-cycle in $A$, then $A$ is not necessary synchronizing as shown in the following example.

**Example 2.9.** Consider 9-state circular semi-flower automaton $A_2$ given in the Figure 2. Clearly, $V(C) = \{q_0, q_3, q_6\}$. By using the computer algebra system GAP–Groups, Algorithms and Programming [27] we observe that the transition monoid $M(A_2)$ does not contain a constant transformation. Hence, $A_2$ is non-synchronizing.

![Figure 2. 9-state non-synchronizing CSFA $A_2$ with $|V(C)| = 3$](image)

**Conclusion and further directions**

This work investigated the synchronization of circular semi-flower automata (CSFA). We proved that every semi-flower automaton is one-cluster automaton,
and subsequently we observed that every semi-flower automaton containing a 1-cycle is synchronizing. Further, for an odd number \( n \), we proved that every \( n \)-state CSFA containing a 2-cycle is synchronizing. Using the computer algebra system GAP, we finally provided examples of 6-state non-synchronizing CSFA and 9-state non-synchronizing CSFA containing, respectively, 2-cycle and 3-cycle.

In the present work, we observed that an \( n \)-state CSFA is in general not synchronizing. Therefore, one could investigate on the sufficient conditions for the synchronization of an \( n \)-state CSFA.

References

[1] A. Trahtman. The Černý conjecture for aperiodic automata, *Discrete Math. Theoret. Comput. Sci.*, 9(2): 3–10, 2007.
[2] B. Steinberg. The Černý conjecture for one-cluster automata with prime length cycle, *Theor. Comput. Sci.*, 412(39): 5487–5491, 2011.
[3] D. Eppstein. Reset sequences for monotonic automata, *SIAM J. Comput.*, 19(3): 500–510, 1990.
[4] D. S. Ananichev and M. V. Volkov. Synchronizing monotonic automata, *Theor. Comput. Sci.*, 327(3): 225–239, 2004.
[5] D. S. Dummit and R. M. Foote. *Abstract algebra*. John Wiley & Sons Inc., 2004.
[6] E. V. Pribavkina. Slowly synchronizing automata with zero and noncomplete sets. *Mathematical Notes*, 90: 411–417, 2011.
[7] J. Bang-Jensen and G. Gutin. *Digraphs* - *theory, algorithms and applications*, Springer, 2002.
[8] J. Černý. Poznámka k homogénnym experimentom s konečnými automatmi, *Mat. fyz. čas SAV*, 14: 208–215, 1964.
[9] J.-E. Pin. Sur un cas particulier de la conjecture de Černý. In *Automata, languages and programming (Fifth Internat. Colloq., 1978)*, volume 62 of *Lecture Notes in Comput. Sci.*, pp.345–352, 1978.
[10] J.-E. Pin. On two combinatorial problems arising from automata theory, *Ann. Discrete Math.*, 17: 535–548, 1983.
[11] J. Kari. Synchronizing finite automata on Eulerian digraphs, *Theor. Comput. Sci.*, 295: 223–232, 2003.
[12] L. Dubuc. Sur les automates circulaires et la conjecture de Černý. *RAIRO Inform. Théor. Appl.*, 32(1-3): 21–34, 1998.
[13] L. Giambruno. *Automata-theoretic Methods in Free Monoids and Free Groups*. PhD thesis, Universit degli Studi di Palermo, Italy, 2007.
[14] L. Giambruno and A. Restivo. An automata-theoretic approach to the study of the intersection of two submonoids of a free monoid. *Theor. Inform. Appl.*, 42(3): 503–524, 2008.
[15] M. A. Perles and M. O. Rabin and E. Shamir. The theory of definite automata, *IEEE Trans. Electronic Computers*, 12(3): 233–243, 1963.
[16] M. Béal and D. Perrin. A quadratic upper bound on the size of a synchronizing word in one-cluster automata, In *Developments in Language Theory, DLT 2009*, volume 5583 of *Lecture Notes in Comput. Sci.*, pp.81–90, 2009.
[17] M. Béal and M. V. Berlinkov and D. Perrin. A quadratic upper bound on the size of a synchronizing word in one-cluster automata, *Int. J. Found. Comput. Sci.*, 22(2): 277–288, 2011.
[18] M. Grech and A. Kisielewicz. The Černý conjecture for automata respecting intervals of a directed graph, *Discrete Math. Theoret. Comput. Sci.*, 15(3): 61–72, 2013.
[19] M. V. Lawson. *Finite automata*. Chapman & Hall/CRC, 2004.
[20] M. V. Volkov. Synchronizing automata and the Černý conjecture, In *Language and Automata Theory and Applications, LATA 2008*, volume 5196 of *Lecture Notes in Comput. Sci.*, pp.11–27, 2008.
[21] M. V. Volkov. Synchronizing automata preserving a chain of partial orders, *Theor. Comput. Sci.*, 410(37): 3513–3519, 2009.
[22] S. N. Singh. *Semi-Flower Automata*. PhD thesis, IIT Guwahati, India, 2012.
[23] S. N. Singh and K. V. Krishna. The rank and Hanna Neumann property of some submonoids of a free monoid. *Ann. Math. Inform.*, 40: 113–123, 2012.
[24] S. N. Singh and K. V. Krishna. A sufficient condition for the Hanna Neumann property of submonoids of a free monoid. *Semigroup Forum*, 86(3): 537–554, 2013.

[25] S. N. Singh and K. V. Krishna. The holonomy decomposition of some circular semi-flower automata. *Acta Cybernet.*, 22(4): 791–805, 2016.

[26] S. N. Singh and K. V. Krishna. On syntactic complexity of circular cemi-flower automata, In *Implementation and Application of Automata, CIAA 2018*, volume 10977 of *Lecture Notes in Comput. Sci.*, pp.312–323, 2018.

[27] The GAP Group. *GAP — Groups, Algorithms, and Programming*, Version 4.9.2, 2018, [http://www.gap-system.org](http://www.gap-system.org)

[28] V. V. Gusev and M. I. Maslennikova and E. V. Pribavkina, Principal ideal languages and synchronizing automata, *Fundam. Inform.*, 132(1): 95–108, 2014.

Department of Mathematics, Central University of South Bihar, Gaya, India

E-mail address: shub@cub.ac.in

Department of Mathematics, Central University of South Bihar, Gaya, India

E-mail address: ankitraj@cusb.ac.in