Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 13, No. 4, 2021 Article ID IJIM-1493, 12 pages
Research Article

Zero-forcing Finite Automata

M. Shamsizadeh [*†], M. M. Zahedi [‡], M. Golmohamadian [§], KH. Abolpour [¶]

Received Date: 2021-02-13 Revised Date: 2021-06-21 Accepted Date: 2021-08-01

Abstract

The current study aims to establish a connection between graphs and automata theory, which apparently demonstrate different mathematical structures. Through searching out some properties of one of these structures, we try to find some new properties of the other structure as well. This will result in obtaining some unknown properties. At first, a novel automaton called zero-forcing (Z-F) finite automata is defined according to the notion of a zero-forcing set of a graph. It is shown that for a given graph and for some zero forcing sets, various Z-F-finite automata will be obtained. In addition, the language and the closure properties of Z-F-finite automata, in particular; union, connection, and serial connection are studied. Moreover, considering some properties of graphs such as the closed trail, connected and complete; some new features for Z-F-finite automata are presented. Further, it is shown that there is not any finite graph such that f be a part of the language of its Z-F-finite automata. Actually, it is proved that for every given graph, the Z-F-finite automata of it does not show any closed trail containing all edges for every zero forcing set, but if the graph G has been a closed trail containing all edges, then the Z-F-finite automata of it has a weak closed trail containing all edges. Some examples are also given to clarify these new notions.

Keywords : Graph; Zero forcing set; Automata; Graph automata; Language of automata.

1 Introduction

An automaton is a mathematical theory in investigates behavior, structure and their relationship to discrete systems. Directable automata were introduced by P. H. Strake in [24] and J. Cerny in [10], and also definite automata by S. C. Kleene in [16]. Today, finite automata have many applications in plenty of areas of computer science such as databases, functional languages, bisimulation, and biology, for more information see [1, 9, 11, 14, 18, 21, 22, 23].

For more than one hundred years, the development of graph theory was inspired and guided mainly by the Four-Colour Conjecture. The resolution of the conjecture by K. Appel and W. Haken in 1976, the year in which our first book Graph Theory with Applications appeared, marked a turning point in its history. Since then, the subject has experienced explosive growth, due in large measure to its role as an essential structure underpinning modern applied mathematics.

*Corresponding author. shamsizadeh.m@gmail.com, Tel:+98(917)3084435.
†Department of Mathematics, Behbahan Khatam Alanbia University of Technology, Khouzestan, Iran.
‡Department of Mathematics, Graduate University of Advanced Technology, Kerman, Iran.
§epartment of Mathematics, Tarbiat Modares University, Tehran, Iran.
¶Department of Mathematics, Shiraz Branch, Islamic Azad University, Shiraz, Iran.

Graph theory is a delightful playground for the exploration of proof techniques in discrete mathematics and its results have applications in many areas of computing, social and natural science. Computer science and combinatorial optimization, in particular, draw upon and contribute to the development of the theory of graphs. Moreover, in a world where communication is of prime importance, the versatility of graphs makes them indispensable tools in the design and analysis of communication networks. A graph is a pair represented by $G = (V, E)$, where V is the set of vertices and E is the set of edges, what we call a graph is sometimes called a simple graph. The order of a graph G, denoted by $|G|$, is the number of vertices of G. The notion of a zero forcing set of a simple graph was introduced in [2] to bound the minimum rank for numerous families of graphs. The zero forcing parameters have been considered by some expert researchers, for additional sources on this topic see [3, 5, 12, 15, 19]. Independently, physicists have studied this parameter, referred to as the graph infection number, in conjunction with control of quantum systems [7, 8, 6, 20]. It also arises in computer science in the context of fast-mixed searching [20]. Various models of graph generation have been introduced in the literature and have been investigated in many directions [13]. However a general theory of graph automata (i.e., automata with a graph as input and output) is still missing from formal graph language theory. Historically, Arbib and Give'on were the first who extended tree automata to operate on planner ordered acyclic graphs [4].
This current study aims to establish a connection between graphs and automata theory, which apparently show different mathematical structures. Through searching out some properties of one of these structures, we attempt to find some new properties of the other structures as well. Accordingly, this will result in obtaining some unknown properties. Also, the zero forcing set and zero forcing number of a graph are defined and a novel automaton by using the zero forcing set of a graph is presented. This automata is called zero forcing finite automata (Z-F-finite automata). Later for a denoted graph we show that it is possible for various zero forcing set, the induced Z-F-finite automata of them could be different. In addition, the relevant behavior is further discussed and some of the closure properties of the Z-F-finite automata such as union, connection, and serial connection are presented. Moreover, considering some properties of graphs such as the closed trail, connected and complete; we present some new features for Z-F-finite automata. Further, we show that if G is a complete graph and the vertices of G be more than three, then for every zero forcing set $Z(G)$, the language of Z-F-finite automata of it is n^*f. After that, we prove that for every graph G, Z-F-finite automata $\mathcal{A}(Z(G))$ does not have a closed trail containing all edges, for every zero forcing set $Z(G)$. Also, we show that for every given graph, Z-F-finite automata of it does not have a closed trail containing all edges for every zero forcing set, but if the graph G has been a closed trail containing all edges, the Z-F-finite automata of it has a weak closed trail containing all edges. Finally, we prove that in the language of Z-F-finite automata, f^* does not appear where f belongs to the set of alphabet.

2 Preliminaries

In this part, we first review some notions and definitions which will be essential for the other sections.

Definition 2.1. *[2]*

- *Color-change rule: If G is a graph with each vertex colored either white or black, u is a black vertex of G, and exactly one neighbor v of u is white, then change the color of v to black.*
- *Given a coloring of G, the derived coloring is resulted by the color-change rule until no more changes are possible.*
- *A zero forcing set for a graph G is a subset of vertices Z such that if initially the vertices in Z are colored black and the remaining vertices are colored white, the derived coloring of G is all black.*
- *$Z(G)$ refers to the minimum of $|Z|$ over all zero forcing sets $Z \subseteq V(G)$.*

In this note, we say that vertex u forces vertex v if v got black with u.

In this note, if black vertex u of G changes the color of vertex v to black, then we say that vertex u forces vertex v.

Definition 2.2. *[17] Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two automata. We say that $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent if they have the same languages.*

Definition 2.3. *[25] A homomorphism from a simple graph $G = (V_G, E_G)$ to a simple graph $H = (V_H, E_H)$ is a surjection $f : V_G \to V_H$ such that $uv \in E_G$ if and only if $f(u)f(v) \in E_G$. f is called an isomorphism if and only if f is a homomorphism that is one-one.*

Definition 2.4. *[25] The graph G is bipartite if V_G is the union of two disjoint (possibly empty) independent sets called partite sets of G.*

Definition 2.5. *[25] A graph G is connected if each pair of vertices in G belongs to a path.*

Definition 2.6. *[25] A walk is a list $v_0, e_1, v_1, ..., e_k, v_k$ of vertices and edges such that for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i. A trail is a walk with no repeated edge.*

Definition 2.7. *[25] A graph is Eulerian if it has a closed trail containing all edges*

Theorem 2.1. *[25] A graph G is Eulerian if and only if it has at most one nontrivial component and its vertices all have even degree.*

Definition 2.8. *[25] The degree of vertex v in a graph G, writhen $d_G(v)$ or $d(v)$ is the number of edges incident to v, except that each loop at v counts twice. The maximum degree is $\triangle(G)$, the minimum degree is $\delta(G)$ and G is regular if $\triangle(G) = \delta(G)$. It is k-regular if the common degree is k.*

Definition 2.9. *[25] A complete graph is a simple graph in which every pair of distinct vertices is connected by a unique edge.*

Definition 2.10. *[25] A path is a simple graph whose vertices can be ordered such that two vertices are adjacent if and only if they are consecutive in the list.*

3 Z-F-finite automata of graphs

At first, by definition of zero forcing set of a graph, we present zero forcing finite automata (Z-F-finite automata).

Definition 3.1. *Let $G = (V, E)$ be a graph. A zero forcing finite automata (Z-F-finite automata) is a five-tuple machine denoted by $\mathcal{A} = (Q, A, \varphi, I, T)$, where*

1. *$Q = V$ is the finite set of states,*
2. *$A = \{f, n\}$ is the set of alphabet,*
3. *$\varphi : Q \times A \to P(Q)$ is the transition function, where if vertex u forces vertex v in G, then define $\varphi(u, f) = v$ in $\mathcal{A}$ and if $uv \in E$ and u and v do not force each other, then $\varphi(u, n) = v$ and $\varphi(v, n) = u$,*
4. *$I = Z(G)$ is the set of initial states,*
5. *T is the set of final states, which $u \in T$ if and only if u does not force any vertex.*

Naturally, φ can be extended to $\varphi^* : Q \times A^* \to P(Q)$.

Note that if $\mathcal{A}(Z(G))$ is a Z-F-finite automata, then $\mathcal{A}(Z(G))$ recognizes a word w in A^* if $\varphi^*(i, w) \cap T \neq \emptyset$, for some $i \in I$.

Example 3.1. *Let graph $G = (V, E)$ be as in Figure 1, where $V = \{q_1, q_2, q_3, q_4, q_5\}$. Consider the different zero forcing set $Z(G)$ of this graph: (i) $Z_1(G) = \{q_1, q_2\}$. Then Z-F-finite automata $\mathcal{A}(Z_1(G)) = (Q, A, \varphi, I, T)$ is as in Figure 2, where $I = \{q_1, q_2\}$, $A = \{f, n\}$, $T = \{q_3\}$ and*

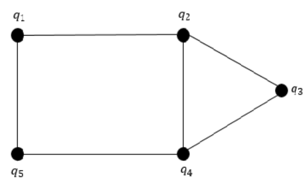

Figure 1: The graph G of Example 3.1

$$\varphi(q_1, n) = q_2, \qquad \varphi(q_1, f) = q_5$$
$$\varphi(q_2, n) = \{q_1, q_4\}, \qquad \varphi(q_2, f) = q_3$$
$$\varphi(q_5, f) = q_4, \qquad \varphi(q_4, n) = q_2$$
$$\varphi(q_4, f) = q_3,$$

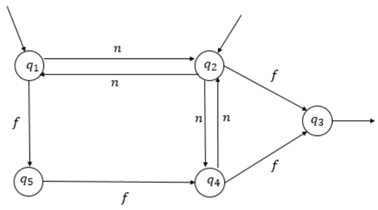

Figure 2: The Z-F-finite automata $\mathcal{A}(Z_1(G))$of Example 3.1

Also,

$$\mathcal{L}(\mathcal{A}(Z_1(G))) = n^*f \cup n^*f^2(n^2n^*f^2)^*n^*f.$$

(ii) $Z_2(G) = \{q_1, q_5\}$. *Then the Z-F-finite automata* $\mathcal{A}(Z_2(G))$ *is as in Figure 3, where* $I = \{q_1, q_5\}, A = \{f, n\}, T = \{q_3\}$ *and*

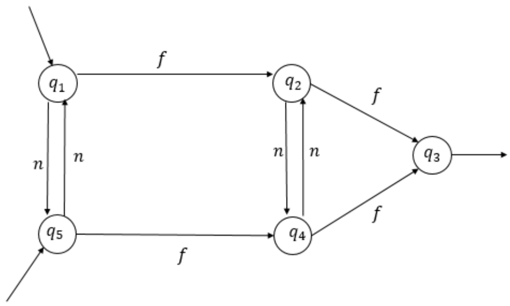

Figure 3: The Z-F-finite automata $\mathcal{A}(Z_2(G))$ of Example 3.1

$\mathcal{L}(\mathcal{A}(Z_2(G))) = n^*fn^*f.$

(iii) $Z_3(G) = \{q_4, q_5\}$. *Then the Z-F-finite automata* $\mathcal{A}(Z_3(G))$ *is as Figure 4, and*

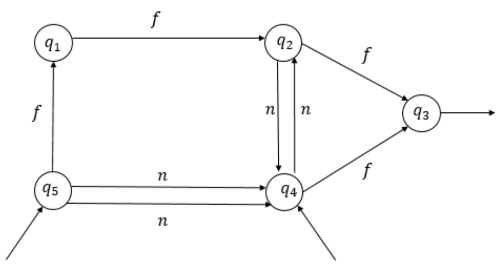

Figure 4: The Z-F-finite automata $\mathcal{A}(Z_3(G))$ of Example 3.1

$$\mathcal{L}(\mathcal{A}(Z_3(G))) = n^*f \cup n^*f^2(n^2n^*f^2)^*n^*f.$$

(iv) $Z_4(G) = \{q_2, q_3\}$. *Then the Z-F-finite automata* $\mathcal{A}(Z_4(G))$ *is as in Figure 5, where*

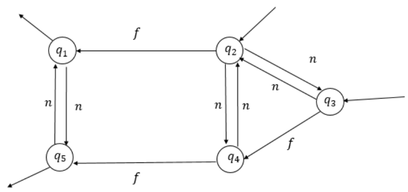

Figure 5: The Z-F-finite automata $\mathcal{A}(Z_4(G))$ of Example 3.1

$I = \{q_2, q_3\}$, $T = \{q_1, q_5\}$ *also,* $\mathcal{L}(\mathcal{A}(Z_4(G))) = n^*fn^*fn^* \cup n^*fn^*$. *(v)* $Z_5(G) = \{q_3, q_4\}$. *Then the Z-F-finite automata* $\mathcal{A}(Z_5(G))$ *is as in Figure 6, where* $I = \{q_3, q_4\}$, $T = \{q_1, q_5\}$ *and,* $\mathcal{L}(\mathcal{A}(Z_4(G))) = n^*fn^*fn^* \cup n^*fn^*$.

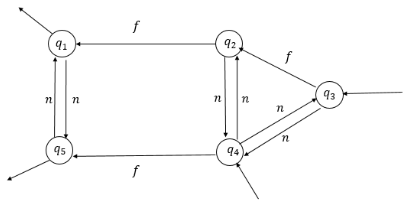

Figure 6: The Z-F-finite automata $\mathcal{A}(Z_5(G))$ of Example 3.1

Example 3.1 shows that for some different zero forcing set of a given graph; the Z-F-finite automata of them are different, too. Also, we see that for some different zero forcing set for the given graph G, we have the same language for Z-F-finite automata of them. For example from the above discussion we see that $\mathcal{L}(\mathcal{A}(Z_1(G))) = \mathcal{L}(\mathcal{A}(Z_3(G)))$ and $\mathcal{L}(\mathcal{A}(Z_4(G))) = \mathcal{L}(\mathcal{A}(Z_5(G)))$. Hence, $\mathcal{A}(Z_1(G))$ is equivalent to $\mathcal{A}(Z_3(G))$ and $\mathcal{A}(Z_4(G))$ is equivalent to $\mathcal{A}(Z_5(G))$.

Example 3.2. *Let graph* G *be as Figure 7 and choosing* $Z_1(G) = \{p_2, p_3, p_4, p_5\}$ *and* $Z_2(G) = \{p_2, p_3, p_5, p_6\}$. *Then the Z-F-finite automata of them have the same language.*

As we see in Example 3.1, it is clear that in a given symmetric graph, for some different symmetric zero forcing sets like $Z_1(G)$ and $Z_3(G)$, the Z-F-finite automata of them are equivalent. The graph of Example 3.2 is a nonsymmetric graph. In this example, we show that for some nonsymmetric graphs we also can find zero forcing sets in which the Z-F-finite automata of them have the same language.

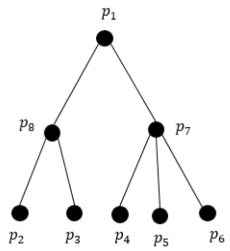

Figure 7: The graph G of Example 3.2

Lemma 3.1. *Let G be a symmetric graph. For every symmetric zero forcing sets $Z_1(G)$ and $Z_2(G)$, $\mathcal{A}(Z_1(G))$ and $\mathcal{A}(Z_2(G))$ are isomorphic. Also, $\mathcal{L}(\mathcal{A}(Z_1(G))) = \mathcal{L}(\mathcal{A}(Z_2(G)))$.*

Notice that in Figure 8, if we consider $Z_1(G) = \{u_1, v_1\}$ and $Z_1(G) = \{v_1, v_2\}$, Z-F-finite automata of them do not have the same language. So, symmetric is essential.

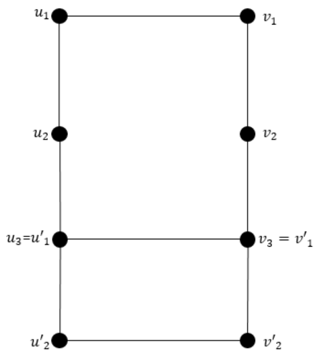

Figure 8: The graph G

Example 3.3. *Consider symmetric graph G as Figure 1. The zero forcing sets $Z_1(G)$ and $Z_3(G)$ are symmetric, too. By Example 3.1 and Lemma 3.1, $\mathcal{L}(\mathcal{A}(Z_1(G))) = \mathcal{L}(\mathcal{A}(Z_3(G)))$. Similarly, $\mathcal{A}(Z_4(G)) = \mathcal{A}(Z_5(G))$ are isomorphic and $\mathcal{L}(\mathcal{A}(Z_4(G))) = \mathcal{L}(\mathcal{A}(Z_5(G)))$.*

Definition 3.2. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two automata. A homomorphism from $\mathcal{A}_1$ onto $\mathcal{A}_2$ is a function g from Q_1 onto Q_2 such that for every $q, q' \in Q_1$ and $u \in A$, the following conditions hold:*

- *$q \in I_1$ if and only if $g(q) \in I_2$,*
- *$\varphi(q, u) = q'$ if and only if $\varphi(g(q), u) = g(q')$,*
- *$q \in T_1$ implies that $g(q) \in T_2$.*

g is called an isomorphism if and only if g is homomorphism that is one-one and $q \in T_1$ if and only if $g(q) \in T_2$.

Theorem 3.1. *Let G and H be two graphs and G be isomorphic to H. If $Z(G)$ is a zero-forcing set for G, then $g(Z(G)) = \{g(u)|u \in Z(G)\}$ is a zero-forcing set for H.*

Proof. Let G be a graph and $Z(G)$ be a zero-forcing set of it. Consider $Z = \{g(u)|u \in Z(G)\}$. Now, we show that Z is a zero forcing set for H. At first, let $u \in Z(G)$ and u forces v. Then $g(u) \in Z$. For every $w' \in H$ such that $g(u)w' \in E_H$, there exists $w \in G$, where $g(w) = w'$ and $uw \in E_G$. Since u forces v, then for every $v \neq w \in G$ in which $uw \in E_G$, w is black. So, $g(w)$ is black for every $v \neq w \in G$. Then $g(u)$ forces $g(v)$. We continue this process. Hence, u forces v if and only if $g(u)$ forces $g(v)$. □

Lemma 3.2. *Let G and H be two graphs and G be homomorphic to H. Then $|Z(H)| \leq |Z(G)|$.*

Example 3.4. *Let graphs G and H be as Figures 9 and 10, respectively. Consider homomorphism*

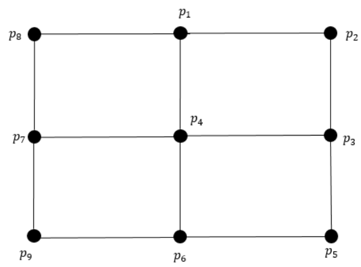

Figure 9: The graph G of Example 3.4

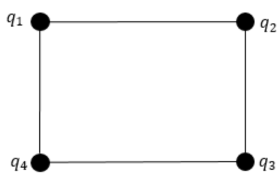

Figure 10: The graph H of Example 3.4

g as follow: $g : G \to H$, where

$$g(p_1) = q_1 = g(p_6),$$
$$g(p_2) = q_2 = g(p_8) = g(p_5),$$
$$g(p_3) = q_3 = g(p_7),$$
$$g(p_4) = q_4 = g(p_9).$$

Obviously, $\{p_1, p_4, p_8\}$ *is a zero forcing set for graph* G *and* $\{q_1, q_2\}$ *is a zero forcing set for* H.

Example 3.4, shows that two graphs G and H are homomorphic and $|Z(H)| < |Z(G)|$.

Theorem 3.2. *Let* G *and* H *be two graphs such that* G *and* H *be isomorphic. Then there exist zero forcing sets* $Z(G)$ *and* $Z(H)$ *such that* $\mathcal{A}(Z(G))$ *and* $\mathcal{A}(Z(H))$ *are isomorphic.*

Proof. Let G and H be two graphs in which they are isomorphic. Then there exists a bijective $g : V(G) \to V(H)$. Let $Z(G)$ be a zero-forcing set of G. Then $g(Z(G)) = \{g(u) | u \in Z(G)\}$ is a zero-forcing set of H. So, if u forces v in G, then $g(u)$ forces $g(v)$ in H. Therefore, $\mathcal{A}(Z(G))$ and $\mathcal{A}(Z(H))$ have the same form. Now, we show that $\mathcal{A}(Z(G))$ and $\mathcal{A}(Z(H))$ are isomorphic. Let $\varphi(u, f) = v$. Then $\varphi(g(u), f) = g(v)$. Clearly, $\varphi(u, n) = v$ if and only $\varphi(g(u), n) = g(v)$. Now, let $u \in T_{\mathcal{A}(Z(G))}$ and $g(u) \notin T_{\mathcal{A}(Z(H))}$. Since $g(u) \notin T_{\mathcal{A}(Z(H))}$, then there exists $v \in V(H)$ such that $\varphi(g(u), f) = v$. So, $g(u)v \in E_H$. The function g is onto, then there is $v' \in G$ such that $g(v') = v$. So, $uv' \in E_G$, it implies that u forces v'. Then it is contradiction with $u \in T_{\mathcal{A}(Z(G))}$. Similarly, if $g(u) \in T_{\mathcal{A}(Z(H))}$, then $u \in T_{\mathcal{A}(Z(G))}$. Hence, $\mathcal{A}(Z(G))$ and $\mathcal{A}(Z(H))$ are isomorphic. □

Theorem 3.3. *Let* G *be a complete* $X - Y$ *bipartite graph and* $|X| \geq 2, |Y| \geq 2$. *Then for every zero forcing sets* $Z(G)$, $\mathcal{L}(\mathcal{A}(Z(G))) = n^* f n^*$.

Proof. Let G be a complete $X - Y$ bipartite graph. We have $|Z(G)| = |X| + |Y| - 2$. Consider $Z(G) = \{u_1, u_2, ..., u_{|x|-1}\} \cup \{v_1, v_2, ..., v_{|Y|-1}\}$ such that $u_i \in X$ and $v_i \in Y$. Clearly, $u_1, u_2, ..., u_{|x|-1}$ force $v_{|Y|}$ and $v_1, v_2, ..., v_{|Y|-1}$ force $u_{|X|}$. By considering Definition 3.1, u_k does not force v_l, for $k = 1, 2, ..., |X| - 1$ and $l = 1, ..., |Y| - 1$. Similarly, v_l does not force u_k, where $l = 1, ..., |Y| - 1$ and $k = 1, ..., |X| - 1$. Also, $u_{|X|}$ does not force $v_{|Y|}$. Hence, $\mathcal{L}(\mathcal{A}(Z(G))) = n^* f n^*$. □

Example 3.5. *Let the complete* $X - Y$ *bipartite* G *be as Figure 11. By considering* $Z(G) = \{p_1, p_2, p_5, p_6\}$, *Z-F-finite automata* $\mathcal{A}(Z(G))$ *is as Figure 12. By Figure 12 and Theorem 3.3, it is obvious that* $\mathcal{L}(\mathcal{A}(Z(G))) = n^* f n^*$

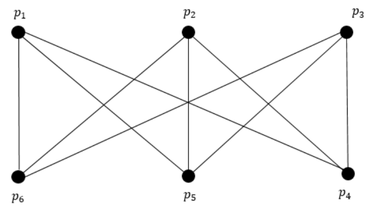

Figure 11: The graph G of Example 3.5

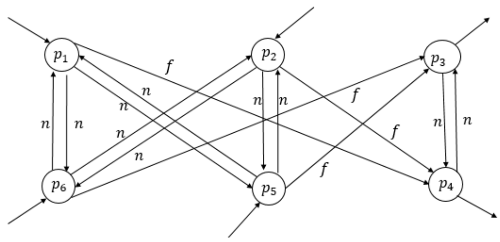

Figure 12: The Z-F-finite automata $\mathcal{A}(Z(G))$ of Example 3.5

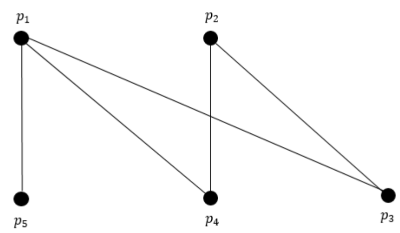

Figure 13: The graph G of Example 3.6

Example 3.6. *Let graph* G *be as Figure 13. By* $Z(G) = \{p_4, p_5\}$, $\mathcal{A}(Z(G))$ *is as Figure 13. Also, the language of Z-F-finite automata* $\mathcal{A}(Z(G))$ *is* $n^* f n^* \cup f n^* f n^*$.

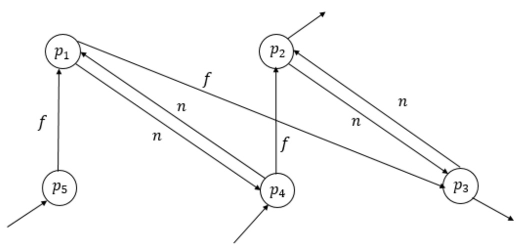

Figure 14: The Z-F-finite automata $\mathcal{A}(Z(G))$ of Example 3.6

Example 3.6 shows that Theorem 3.3 does not hold for every $X - Y$ bipartite graphs.

Theorem 3.4. *Let* G *be a connected graph. Then for every* $Z(G)$, $\mathcal{A}(Z(G))$ *is accessible.*

Proof. By considering Definitions 2.1 and 3.1, the proof is clear. □

Example 3.7. *Let graph G be as in Example 3.6. It is a connected graph. Clearly, the Z-F-finite automata $\mathcal{A}(Z(G))$, as in Example 3.6, is accessible.*

Theorem 3.5. *For every $\mathcal{L} = n^*f^ln^*$, where $l \geq 1$, there exists a graph G such that the language Z-F-finite automata of it is $\mathcal{L}$.*

Proof. Let $\mathcal{L} = n^*f^ln^*$, where $l \geq 1$. We construct the graph G as follows: at first, consider two adjacent vertices u_1 and v_1 such that u_1 is adjacent u_2 and v_1 is adjacent v_2. We continue this way until u_l is adjacent u_{l+1} and v_l ia adjacent v_{l+1}. Finally, let u_{l+1} and v_{l+1} are adjacent. By considering $Z(G) = \{u_1, v_1\}$ obviously, $\mathcal{L}(\mathcal{A}(Z(G))) = n^*f^ln^*$. □

Example 3.8. *Let $\mathcal{L} = n^*f^2n^*$ and Graph G_1 be as in Figure 15. By considering $Z(G_1) =$*

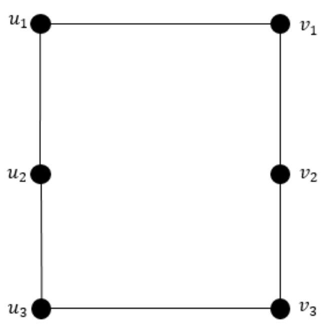

Figure 15: The graph G_1 of Example 3.8

$\{u_1, v_1\}$, $\mathcal{A}(Z(G_1))$ is as in Figure 16. By $\mathcal{A}(Z(G_1))$ as in Figure 16, $\mathcal{L}(\mathcal{A}(Z(G_1))) = n^*f^2n^*$.

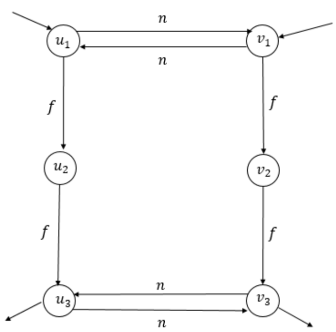

Figure 16: The Z-F-finite automata $\mathcal{A}(Z(G_1))$ of Example 3.8

Theorem 3.6. *Let G be a connected 2-regular graph. If $|G|= 2m + 1$, m is an integer, then $\mathcal{L}(\mathcal{A}(Z(G))) = n^*f^m$ and if $|G|= 2m + 2$, then $\mathcal{L}(\mathcal{A}(Z(G))) = n^*f^mn^*$.*

Proof. Since G is a 2-regular graph, then for every $q \in V_G$, degree of q is 2. Let $q_1, q_1' \in Z(G)$, $V_G = \{q_1, q_1', ..., q_m, q_m', q_{m+1}, q_{m+1}'\}$, q_1 forces q_2 and q_1' forces q_2'. We continue this manner until q_{m-1} forces q_m and q_{m-1}' forces q_m'. Now, we have two cases. The first one, let $|G|= 2m + 1$ and $q_{m+1} = q_{m+1}'$. Then q_m' and q_m force q_{m+1}. So, clearly $\mathcal{L}(\mathcal{A}(Z(G))) = n^*f^m$. The last one, if $|G|= 2m + 2$, then q_m forces q_{m+1} and q_m' forces q_{m+1}'. Since q_{m+1} and q_{m+1}' are adjacent and they do not force anything, then $q_{m+1}, q_{m+1}' \in T$ and $\varphi(q_{m+1}, n) = q_{m+1}'$ and $\varphi(q_{m+1}', n) = q_{m+1}$. Obviously, $\mathcal{L}(\mathcal{A}(Z(G))) = n^*fn^*$. □

Corollary 3.1. *For every connected 2-regular graph G_1 and G_2 such that $|G_1|= |G_2|= 2m + 1$, the Z-F-finite automata of them are equivalent.*

Corollary 3.2. *Let G_1 and G_2 be two connected 2-regular graph and $|G_1|= |G(2)|= 2m + 2$. Then the Z-F-finite automata of them are equivalent.*

Example 3.9. *Let G_1 be as in Figure 17. By $Z(G_1) = \{p_1, p_2\}$, the Z-F-finite automata $\mathcal{A}(Z(G_1))$ is as Figure 18. Clearly,*

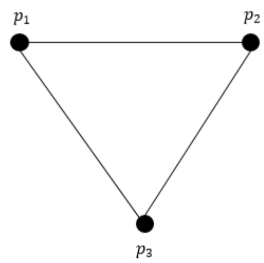

Figure 17: The graph G_1 of Example 5.3

$\mathcal{L}(\mathcal{A}(Z(G_2))) = n^*f$. Now, let graph G_2 be as in Figure 19. Considering $Z(G_2) = \{p_1, p_2\}$ the Z-F-finite automata $\mathcal{A}(Z(G_2))$ is as in Figure 20. Clearly, $\mathcal{L}(\mathcal{A}(Z(G_2))) = n^*fn^*$.

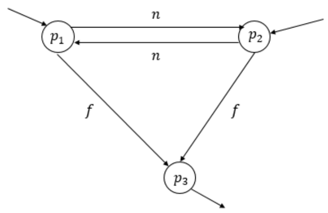

Figure 18: The Z-F-finite automata $\mathcal{A}(Z(G_1))$ of Example 5.3

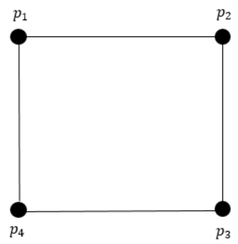

Figure 19: The graph G_2 of Example 3.9

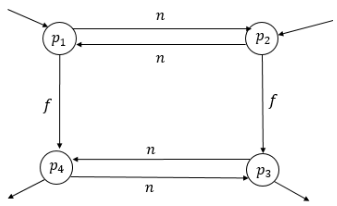

Figure 20: The Z-F-finite automata $\mathcal{A}(Z(G_2))$ of Example 3.9

4 Closure Properties of zero-forcing finite automata

In this section, we define the notions of union, connection, and the serial connection for zero forcing finite automata.

Definition 4.1. *(Connection) Given $\mathcal{L}_1$ and $\mathcal{L}_2$ the subsets of A^*, we define their product $\mathcal{L}_1.\mathcal{L}_2$ such that $\mathcal{L}_1 = n^*f^ln^*, \mathcal{L}_2 = n^*f^{l'}n^*$ in an obvious way, $\mathcal{L}_1.\mathcal{L}_2 = n^*f^ln^*f^{l'}n^*$.*

Theorem 4.1. *Let $\mathcal{L}_1 = n^*f^ln^*$ and $\mathcal{L}_2 = n^*f^{l'}n^*$. Then there exist a graph G and a zero forcing set $Z(G)$ for it such that $\mathcal{L}(\mathcal{A}(Z(G))) = n^*f^ln^*f^{l'}n^*$.*

Proof. By considering the proof of Theorem 3.5, there exist two graphs G_1, G_2 and zero forcing sets $Z(G_1)$ and $Z(G_2)$ such that $\mathcal{L}(\mathcal{A}(Z(G_1))) = \mathcal{L}_1$ and $\mathcal{L}(\mathcal{A}(Z(G_2))) = \mathcal{L}_2$. In graph G_1, let u_1 and v_1 be adjacent and $u_iu_{i+1}, v_iv_{i+1} \in E_{G_1}$, $1 \leq i \leq l$ and also $v_{l+1}u_{l+1} \in E_{G_1}$. Similarly, in graph G_2 suppose that $u'_1v'_1 \in E_{G_2}$ and $u'_iu'_{i+1}, v'_iv'_{i+1} \in E_{G_2}$, where $1 \leq i \leq l'$ and $u'_{l'+1}v'_{l'+1} \in E_{G_2}$. Now, for construction the graph G follow this way: $V_G = \{u_1, ..., u_{l+1}, v_1, ..., v_{l+1}, u'_1, ..., u'_{l'+1}, v'_1, ..., v'_{l'+1}\}$. At first, consider graph G_1 and let $u'_1 = u_{l+1}$ and $v'_1 = v_{l+1}$. Also, $zz' \in E_G$ if and only if $zz' \in E_{G_1}$ or $zz' \in E_{G_2}$, where $z, z' \in V(G)$. Let $Z(G) = \{u_1, v_1\}$. Clearly, $\mathcal{L}(\mathcal{A}(Z(G)) = n^*f^ln^*f^{l'}n^*$. □

Example 4.1. *Let $\mathcal{L}_1 = n^*f^2n^*$, $\mathcal{L}_2 = n^*fn^*$ and two graphs G_1 and G_2, like Figures 15 and 21. By considering zero forcing sets $Z(G_1) = \{u_1, v_1\}$ and $Z(G_2) = \{u'_1, v'_1\}$, $\mathcal{L}(\mathcal{A}(Z(G_1))) = n^*f^2n^*$ and $\mathcal{L}(\mathcal{A}(Z(G_1))) = n^*fn^*$. Now, by the proof of Theorem 4.1, we construct graph G as in Figure 8. Now, let $Z(G) = \{u_1, v_1\}$. Then the Z-F-finite automata $\mathcal{A}(Z(G))$ is as in Figure 22. Clearly, $\mathcal{L}(\mathcal{A}(Z(G))) = n^*f^2n^*fn^*$*

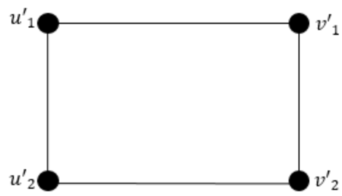

Figure 21: The graph G_2 of Example 4.1

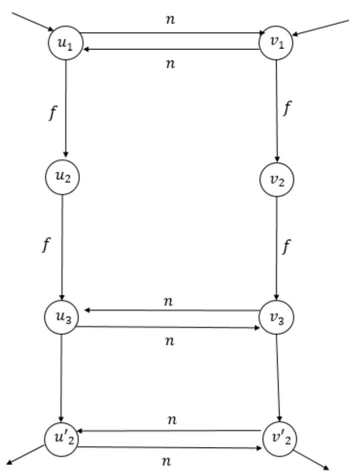

Figure 22: The Z-F-finite automata $\mathcal{A}(Z(G))$ of Example 4.1

Corollary 4.1. *(Serial Connection) Let* $A = \{\mathcal{L} | \mathcal{L} = n^* f^l n^*, l \geq 1\}$. *Then*

$$\mathcal{L}^m = \underbrace{\mathcal{L}.\mathcal{L}.\ldots\mathcal{L}}_{m}$$

is a recognizable Z-F-finite automata.

Definition 4.2. *(Union) Let* $G_1 = (V_{G_1}, E_{G_1})$ *and* $G_2 = (V_{G_2}, E_{G_2})$ *be two graphs such that* $V_{G_1} \cap V_{G_2} = \emptyset$. *The direct sum of graphs* G_1 *and* G_2*, writhen* $G_1 \cup G_2$*, is the graph with vertex set* $V_{G_1} \cup V_{G_2}$ *and edge set* $E_{G_1} \cup E_{G_2}$.

Definition 4.3. *The set* $\mathcal{L} \subseteq A^*$ *is called Z-F-recognizable if there exists a graph like* G *such that* $\mathcal{L}(A(Z(G))) = \mathcal{L}$*, for some* $Z(G)$.

Theorem 4.2. *Let* $\mathcal{L}_1$ *and* $\mathcal{L}_2$ *be two Z-F-recognizable. Then* $\mathcal{L}_1 \cup \mathcal{L}_2$ *is Z-F-recognizable.*

Proof. Since $\mathcal{L}_1$ and $\mathcal{L}_2$ are Z-F-recognizable, then there exist $G_1 = (V_{G_1}, E_{G_1})$, $G_2 = (V_{G_2}, E_{G_2})$, $Z(G_1)$ and $Z(G_2)$ such that $\mathcal{L}(\mathcal{A}(Z(G_1))) = \mathcal{L}_1$ and $\mathcal{L}(\mathcal{A}(Z(G_2))) = \mathcal{L}_2$. Without loss of generality, suppose that $V_{G_1} \cap V_{G_2} = \emptyset$. Consider $G = (V_G, E_G)$ such that $V_G = V_{G_1} \cup V_{G_2}$ and $E_G = E_{G_1} \cup E_{G_2}$. Since $V_{G_1} \cap V_{G_2} = \emptyset$, then $Z(G_1) \cup Z(G_2)$ is a zero forcing set for G. Clearly, $\mathcal{L}(\mathcal{A}(Z(G_1) \cup Z(G_2))) = \mathcal{L}_1 \cup \mathcal{L}_2$. □

5 Properties of languages of Z-F-finite automata

In this section, we present definitions of the closed trail, connected and complete for Z-F-finite automata. Further, we discuss the language of Z-F-finite automata.

Theorem 5.1. *There is not any finite graph such that* f^* *be a part of the language of its Z-F-finite automata.*

Proof. By considering the definitions of zero-forcing set and Z-F-finite automata, two vertices can not force each other. So, we can obtain f^* if the Z-F-finite automata have been the infinity states. Then it is contraction. □

Theorem 5.2. *Let* $\mathcal{L}$ *be a recognizable Z-F-finite automata. Then* $\mathcal{L} = n^* f^l n^* (n^* f n^*)^*$ *or* $\mathcal{L} = f^l$ *or* $\mathcal{L} = f^l n^*$.

Proof. Let $w \in \mathcal{L}$ and G be a graph such that $\mathcal{L}(\mathcal{A}(Z(G))) = \mathcal{L}$, for some $Z(G)$. Then $\varphi^*(i, w) \in T$. Let $w = a_1 a_2 \ldots a_n$. Then the path is as follows:

$$i \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \rightarrow \ldots \xrightarrow{a_n} q_n \in T,$$

where $i, q_1, q_2, \ldots, q_n \in Q$. There exist two cases for two adjacent vertices i and q_1 in graph G: in the first case, let vertex i forces q_1, then we have $a_1 = f$. In the last case, let vertices i and q_1 do not force each other, then $a_1 = n$. Also, by considering Definition 3.1, $\varphi(q_1, n) = i$ and $\varphi(i, n) = q_1$. So, with two vertices i and q_1 we can make n^*. Similarly, for two vertices q_1 and q_2 we have $\varphi(q_1, f) = q_2$ or $\varphi(q_1, n) = q_2$. Hence, the claim holds. □

Theorem 5.3. *Let* G *be a complete graph and* $|G| \geq 3$. *Then for every* $Z(G)$, $\mathcal{L}(\mathcal{A}(Z(G))) = n^* f$.

Proof. Since G is complete, then $\delta(G) = l - 1$, where $|G| = l$. By considering $|Z(G)| \geq \delta(G)$, we have $|Z(G)| = l - 1$. Since $|G| \geq 3$ and $|Z(G)| = l - 1$, then the all members of $Z(G)$ are adjacent and they make n^*, also the all members of $Z(G)$ force the last vertex. Hence, $\mathcal{L}(\mathcal{A}(Z(G))) = n^* f$. □

Example 5.1. *Let* G *be as Figure 23. By*

Figure 23: The graph G of Example 5.1

$Z(G) = \{p_1, p_2, p_3\}$, *Z-F-finite automata* $\mathcal{A}(Z(G))$ *is as Figure 24. Clearly,* $\mathcal{L}(\mathcal{A}(Z(G))) = n^* f$.

Definition 5.1. *A path in a Z-F-finite automata is a list* $p_0, a_1, p_1, \ldots, a_k, p_k$*, such that* $\varphi(p_{i-1}, a_i) = p_i$*, for* $1 \leq i \leq k$. *A trail in Z-F-finite automata is a path with no repeats edges.*

Theorem 5.4. *Let* G *be a path and* $|G| = l$. *Then* $\mathcal{L}(\mathcal{A}(Z(G))) = f^{l-1}$*, for every* $Z(G)$.

Proof. Since G is a path, then $|Z(G)| = 1$. So, every vertex forces the next vertex in the list

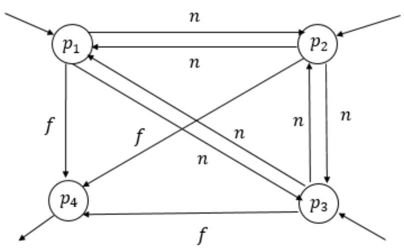

Figure 24: The Z-F-finite automata $\mathcal{A}(Z(G))$ of Example 5.1

and the last vertex is the only terminal member. Therefore, $\mathcal{L}(\mathcal{A}(Z(G))) = f^{l-1}$, where $l-1$ is the number of edges in graph G. □

Example 5.2. *Let graph G be as in Figure 25. Consider $Z(G) = \{p_1\}$, Z-F-finite automata*

Figure 25: The graph G of Example 5.2

$\mathcal{A}(Z(G))$ is as Figure 26. Also, $\mathcal{L}(\mathcal{A}(Z(G))) =$

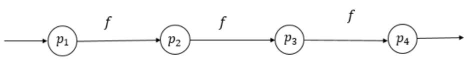

Figure 26: The Z-F-finite automata $\mathcal{A}(Z(G))$ of Example 5.2

f^3.

Theorem 5.5. *For every graph G, $\mathcal{A}(Z(G))$ does not have a closed trail containing all edges, for every zero forcing set $Z(G)$.*

Proof. Let G be a graph and $Z(G)$ be a zero forcing set of it. If G does not have a closed trail containing all edges, then clearly $\mathcal{A}(Z(G))$ does not have these trail. Now, suppose G has a closed trail containing all edges. Then G is an Eulerian graph. By Theorem 1.2.26, in [25], G has at most one nontrivial component and all its vertices have even degree. So, $|Z(G)| \geq 2$, then for two adjacent vertices u and v belong to $Z(G)$, we have two cases. The first one, u and v force w. The other one, u forces u' and v forces v'. Obviously $\mathcal{A}(Z(G))$ has not a closed trail containing all edges for both cases. Hence, the claim holds. □

Example 5.3. *Let graph G_1 be as Figure 17. Clearly, graph G_1 has a closed trail, but $\mathcal{A}(Z(G_1))$ has not a closed trail containing all edges, for every zero forcing set for G_1. By choosing $Z(G_1) = \{p_1, p_2\}$. Z-F-finite automata $\mathcal{A}(Z(G_1))$ is as Figure 18.*

Clearly, there is not any closed trail containing all edges in the Z-F-finite automata.

Definition 5.2. *A weak path in a Z-F-finite automata is a list $p_0, a_1, p_1, ..., a_k, p_k$ in which $\varphi(p_{i-1}, a_i) = p_i$ or $\varphi(p_i, a_i) = p_{i-1}$, where $1 \leq i \leq k$. A weak trail in Z-F-finite automata is a weak path with no repeat edges.*

Theorem 5.6. *Let G be a graph and has been a closed trail containing all edges. Then $\mathcal{A}(Z(G))$ has a closed weak trail containing all edges, for every $Z(G)$.*

Theorem 5.5 shows that for every given graph, Z-F-finite automata of it does not have a closed trail containing all edge, for every zero forcing set, but Theorem 5.6 says that if the graph G has been a closed trail containing all edges, the Z-F-finite automata of it has a weak closed trail containing all edges.

Theorem 5.7. *Let G be a graph. Then for every $p \in Q_{\mathcal{A}(Z(G))}$, p is accessible ad coaccessible, for every $Z(G)$. In the other words, $\mathcal{A}(Z(G))$ is trim, for every $Z(G)$.*

Proof. By Definitions 2.1 and 3.1, the proof is obvious. □

Theorem 5.8. *Let G be a connected graph and has no cycle. Then $\mathcal{A}(Z(G))$ is accessible and has no cycle with length at least 3, for every $Z(G)$.*

Proof. By Theorem 3.4, since G is connected, then $\mathcal{A}(Z(G))$ is accessible and by Definitions 2.1 and 3.1 the proof is obvious. □

Corollary 5.1. *Let G be a connected tree. Then $\mathcal{A}(Z(G))$ is accessible and has no cycle with length at least 3, for every $Z(G)$.*

Notice that trees have not cycle, but it does not hold for Z-F-finite automata of it. Some Z-F-finite automata of a tree have a cycle with length 2.

Corollary 5.2. *Let G be an n-vertex graph with $n-1$ edges.*

1. *Then $\mathcal{A}(Z(G))$ is accessible and has no cycle with length at least 3, for every $Z(G)$.*

2. *If G is an n-vertex tree, then $\mathcal{A}(Z(G))$ is accessible and has no cycle with length at least 3, for every $Z(G)$.*

6 Conclusion

In the current study, we presented the notion of zero forcing finite automata by using the notion of zero forcing set. After that, we proved that two Z-F-finite automata are isomorphic when the graphs of them are isomorphic. Afterwards, we discussed the behavior of Z-F-finite automata. In addition, the notions of union, connection, and serial connection for Z-F-finite automata were given. Moreover, by considering some properties of graphs such as the closed trail, connected and complete, we demonstrated some novel features for Z-F-finite automata.

Now, a question is raised here: If nodes and edges of a graph have different labels, how can we obtain the graph automata?

References

[1] L. Aceto, A. Ingolfsdottir, K. G. Larsen, J. I. Srba, Reactive systems: modelling, specifcation and verifcation, *Cambridge University Press*, 2007.

[2] AIM Minimum Rank-Special Graphs Work Group (F. Barioli, W. Barrett, S. Butler, S. M. Cioaba, D. Cvetkovic, S. M. Fallat, C. Godsil, W. Haemers, L. Hogben, R. Mikkelson, S. Narayan, O. Pryporova, I. Sciriha, W. So, D. Stevanovic, H. van der Holst, K. Vander Meulen, A. Wangsness), Zero forcing sets and the minimum rank of graphs, *Linear Algebra and its Applications* 428 (2008) 1628-1648.

[3] E. Almodovar, L. DeLoss, L. Hogben, K. Hogenson, K. Myrphy, T. Peters, C. Ramrez, C. Minimum rank, Maximum nullity and zero forcing number, and spreads of these parameters for selected graph families, *Involve, A Journal of Mathematics* 3 (2010) 371-392.

[4] M. A. Arbib, Y. Give'on, Algebra automata I: Parallel programming as a prolegomena to the categorical approach, *Information and Control* 12 (1968) 331-345.

[5] F. Barioli, W. Barrett, S. Fallat, H. T. Hall, L. Hogben, B. Shader, P. van den Driessche, H. van der Holst, Parameters related to tree-width, zero forcing, and maximum nullity of a graph, *Journal of Graph Theory* 72 (2013) 146-177.

[6] D. Burgarth, D. D'Alessandro, L. Hogben, S. Severini, M. Young. Zero forcing, Linear and quantum controllability for systems evolving on networks, *IEEE Transactions in Automatic Control* 58 (2013) 2349-2354.

[7] D. Burgarth, V. Giovannetti, Full control by locally induced relaxation, *Physical Review Letters* 99 (2007) 100-501.

[8] D. Burgarth, K. Maruyama, Indirect Hamiltonian identification through a small gateway, *New Journal of Physics* 11 (2009) 103-019.

[9] C. G. Cassandras, S. Lafortune, Introduction to discrete event systems, *Springer Science & Business Media*, 2009.

[10] J. Cerny, Poznamka k homogenym, Experimentom s konecinymi automatami, *Matematicko-fyzikln fakulta* 14 (1964) 208-215.

[11] A. Dovier, C. Piazza, A. Policriti, An efficient algorithm for computing bisimulation equivalence, *Theoretical Computer Science* 311 (2004) 221-256.

[12] C. J. Edholm, L. Hogben, M. Huynh, J. Lagrange, D. D. Row, Vertex and edge spread of zero forcing number, maximum nullity, and minimum rank of a graph, *Linear Algebra and its Applications* 436 (2012) 4352-4372.

[13] H. Ehrig, G. Rozenberg, HJ Kreowski, Handbook of graph grammars and computing by graph transformation, *World Scientific*, 1999.

[14] S. Even, On information lossless automata of finite order, *IEEE Transactions on Electronic Computers* 14 (1965) 561-569.

[15] L. H. Huang, G. J. Chang, H. G. Yeh, On minimum rank and zero forcing sets of a graph, *Linear Algebra and its Applications* 432 (2010) 2961-2973.

[16] S. C. Kleene, Representation of events in nerve nets and finite automata, Automata Studies, *Princeton University Press, Princeton, N.J.* 6 (1956) 3-41.

[17] D. S. Malik, J. N. Mordeson, Fuzzy automata and languages: Theory and Applications, *Chapman Hall, CRC Boca Raton, London, New York, Washington DC*, 2002.

[18] M. Roggenbach, M. Majster-Cederbaum, Towards a unified view of bisimulation: a comparative study, *Theoretical Computer Science* 238 (2000) 81-130.

[19] D. Row, A technique for computing the zero forcing number of a graph with a cut-vertex, *Linear Algebra and its Applications* 436 (2012) 4423-4432.

[20] S. Severini, Nondiscriminatory propagation on trees, *Journal of Physics A* 41 (2008) 482-502.

[21] M. Shamsizadeh, M. M. Zahedi, Bisimulation of type 2 for BL-general fuzzy automata, *Soft Computing* 23 (2019) 9843-9852.

[22] M. Shamsizadeh, M. M. Zahedi, Kh. Abolpour, Bisimulation for BL-general fuzzy automata, *Iranian Journal of Fuzzy Systems* 13 (2016) 35-50.

[23] M. Shamsizadeh, M. M. Zahedi, Kh. Abolpour, Reduction of BL-general L-fuzzy automata, *Iranian Journal of Mathematical Sciences* 2020.

[24] P. H. Starke, Abstrakte Automaten, *VEB Deutscher Verlag der Wissenschaften, Berlin*, 1969.

[25] D. B. West, Introduction to graph theory, *Upper Saddle River: Prentice hall* (2001).

Marzieh Shamsizadeh is Assistant Professor in the mathematics department at Behbahan Khatam Alanbia University of Technology, Khouzestan Iran. Her main interests in mathematics are Graph Theory, Automata Theory, and Fuzzy Logic.

Mohammad Mehdi Zahedi is Professor at the Department of Mathematics, Graduate University of Advanced Technology, Kerman, Iran. His researches encompass Graph Theory, Hyper Graph Theory, Automata Theory, Hyperstructures, Algebra and Fuzzy Logic.

Masoumeh Golmohamadian a mathematician and her main goal is to bring novel perspectives to complex problems at the intersection of different branches of mathematics. Her main interests in mathematics are Graph Theory, Hyper Graph Theory, Automata Theory, Hyperstructures and Fuzzy Logic.

Khadijeh Abolpour is Assistant Professor in the Mathematics department at the Islamic Azad University Shiraz Branch in Iran. Her research interests include Fuzzy Systems, Algebra, and Automata Theory.